

Д. Роджерс, Дж. Адамс

МАТЕМАТИЧЕСКИЕ ОСНОВЫ МАШИННОЙ ГРАФИКИ

Перевод со второго английского издания

П. А. Монахова, Г. В. Олохтоновой,

Д. В. Волкова

под редакцией

Ю. М. Баяковского, В. А. Галактионова,

В. В. Мартынюка



Москва «Мир» 2001

MATHEMATICAL ELEMENTS FOR COMPUTER GRAPHICS

Second Edition

David F. Rogers

*Professor of Aerospace Engineering
United States Naval Academy, Annapolis, Md.*

J. Alan Adams

*Professor of Mechanical Engineering
United States Naval Academy, Annapolis, Md.*

McGraw-Hill Publishing Company

New York · St. Louis · San Francisco · Auckland · Bogotá · Caracas · Hamburg · Lisbon
London · Madrid · Mexico · Milan · Montreal · New Delhi · Oklahoma City · Paris
San Juan · São Paulo · Singapore · Sydney · Tokyo · Toronto

УДК 517.91

ББК 22.19

P60

Роджерс Д., Адамс Дж.

P60 Математические основы машинной графики: Пер. с англ. —
М.: Мир, 2001. — 604 с., ил.

ISBN 5-03-002143-4

Полностью переработанное и дополненное второе издание книги известных американских специалистов (перевод первого издания вышел в издательстве «Машиностроение» в 1980 г.). Книгу отличает глубина и фундаментальность материала, четкий и лаконичный стиль изложения, удачное сочетание строгости подхода с практической направленностью. От читателя требуется знакомство с математикой в объеме вузовской программы и знание одного из языков программирования.

Для математиков-прикладников, для всех, кто занимается и интересуется машинной графикой, автоматизацией проектирования, для студентов вузов.

ББК 22.19



Издание осуществлено при поддержке Российского фонда
фундаментальных исследований по проекту № 99-01-14007

Редакция литературы по математическим наукам

Copyright © 1990, 1976 by Mc Graw-Hill, Inc.

© перевод на русский язык,

Монахов П. А., Олохтонова Г. В.,

Волков Д. В., 2001

© оформление, «Мир», 2001

ISBN 5-03-002143-4 (русск.)

ISBN 0-074053529-9 (англ.)

ОТ РЕДАКТОРОВ ПЕРЕВОДА

В 1990 г. в Далласе на конференции SIGGRAPH мы обсуждали с проф. Д. Роджерсом планы перевода и издания на русском языке данной книги. Уже через год перевод книги был подготовлен, но в силу известных обстоятельств выпуск ее задержался на много лет. И только теперь благодаря поддержке РФФИ издание книги стало возможным. Безусловно это знаковое событие. Оно свидетельствует о том, что происходит оживление и восстановление в российской науке, технике, промышленности.

Материал, изложенный в книге, не устарел. Впервые на русском языке появляется достаточно полное функциональное описание технических средств машинной графики, систематическое изложение двумерных и трехмерных преобразований, практическое руководство по геометрическому моделированию, включающее такие объекты, как кривые, поверхности, сплайны. Всего этого так не хватает нашим студентам, аспирантам, инженерам, научным работникам.

Мы искренне благодарны Д. Роджерсу за столь плодотворное сотрудничество и надеемся, что оно будет продолжено.

Ю. М. Балковский, В. А. Галактионов

ПРЕДИСЛОВИЕ К РУССКОМУ ИЗДАНИЮ

Я с большой радостью узнал о выходе в свет в переводе на русский язык второго английского издания «Математических основ машинной графики». Эта книга составит прекрасный тандем вместе с ранее изданной в переводе на русский язык книгой «Алгоритмические основы машинной графики». Я надеюсь, что в совокупности эти две книги образуют фундаментальный материал по машинной графике и в какой-то степени они скомпенсируют существующую нехватку публикаций по этому предмету на русском языке.

Сегодня, даже в большей степени, чем в ту пору, когда была написана эта книга, машинная графика глубоко проникает в искусство, технику, рекламу, связь. Я упомянул лишь некоторые немногие области приложений, а список можно продолжить. В частности, следует отметить Интернет, который возник как средство быстрого и легкого обмена текстовой информацией между учеными, а в настоящее время он сильно зависит от технологии машинной графики.

Почти на каждом вэб-сайте для представления информации используется машинная графика. Многие спецэффекты в видео- и кинофильмах не обходятся без применения средств машинной графики.

В индустриальном мире машинная графика играет огромную роль, особенно в робототехнике, автоматизированном проектировании и производстве, в научной визуализации. В наши дни обычный человек, не задумываясь, может послать картинки своим друзьям и коллегам по электронной почте простым нажатием на кнопку «мышки». Отмечу и такой факт: объем бизнеса в машинной графике достиг 90 млрд. долларов.

За мои без малого три с половиной десятка лет деятельности в области машинной графики произошли поистине удивительные изменения. Как авиакосмический инженер я всегда восхищался прекрасными традициями замечательной российской науки и техники. Я полон ожиданий, что и в новых условиях становления и укрепления открытого общества русские ученые и инженеры внесут значительный выдающийся вклад в дальнейшее развитие машинной графики.

Дэвид Ф. Роджерс
Аннаполис, Мэриленд, США
1 декабря 2000 г.

ВВЕДЕНИЕ К ПЕРВОМУ ИЗДАНИЮ

Прошло уже более десяти лет с момента выхода в свет первого издания книги; за это время машинная графика привлекла к себе пристальное внимание множества специалистов из разных областей знания. В процентном отношении основное внимание в среде увеличивающейся плеяды профессионалов по машинной графике уделялось проблемам программирования, системного проектирования, аппаратуре и т. п. Именно это обстоятельство отмечал Сазерленд в своем предисловии к книге Принса «Interactive Graphics for Computer Aided Design», изданной в 1971 г. Такая ситуация наблюдается и сегодня. Однако я считаю, что уделяется недостаточно внимания именно прикладным проблемам, а это влечет за собой дефицит информации, которую можно было бы использовать при создании действительно работоспособных систем машинной графики. Основное принципиальное требование к разработке систем машинной графики состоит в необходимости глубокого понимания связи ее практических аспектов со спецификой конкретного приложения. Фундаментом большинства приложений машинной графики являются математические методы, особенно геометрия и способы преобразования. Тем не менее многие программисты и аналитики, работающие в области машинной графики, в своих работах обычно не рассматривают основы и не разбирают сложных математических задач. Более того, потенциал машинной графики часто остается полностью не реализованным именно вследствие неадекватного использования математического аппарата с его ограничениями и допущениями. Ценность книги профессоров Роджерса и Адамса состоит в ее практической ориентации. Четкий и лаконичный стиль изложения значительно облегчает задачу читателей по освоению материала. Книга является редким примером такой работы, которую многим практикам следует поместить на видное место в своей библиотеке в качестве незаменимого энциклопедического справочника. Она может быть также с успехом использована и в качестве основы учебного курса.

Я поздравляю авторов с созданием яркой и полезной книги «Математические основы машинной графики».

S. H. «Chas» Chasen
Lockheed Georgia Company, Inc.

ПРЕДИСЛОВИЕ

За четырнадцать лет, прошедших с момента издания первой редакции данной книги, машинная графика претерпела поистине грандиозные изменения и в настоящее время представляет собой важнейшую дисциплину для специалистов из многих областей знания, и в частности, для проектировщиков электронных устройств, инженеров, математиков, физиков и художников. В это время были в совершенстве разработаны прежние концепции машинной графики, а также предложены новые идеи. Однако основополагающие математические понятия остались неизменными. Глубокое овладение базовыми математическими концепциями до сих пор остается главным ключом к пониманию и дальнейшему развитию машинной графики. Предлагаемая книга содержит материал, необходимый для усвоения этих концепций.

Данная редакция книги не является результатом простого пересмотра первого издания, а представляет собой его полную переработку с включением множества новых иллюстраций и более подробных примеров. В книге содержатся формулировки проблем и концепций программных проектов. Алгоритмы, реализующие математические теории, приведены в книге в виде псевдокодов.

Глава 1 содержит новые данные о растровых устройствах и дисплеях, струйных и термических графопостроителях, цветных камерах. В главе приводится также перечень литературы по аппаратному обеспечению машинной графики.

В главе 2 содержится подробное описание двумерных преобразований, дополненное новыми разделами по преобразованиям сплошных тел, а также описание и геометрическая интерпретация однородных координат.

Глава 3 была значительно расширена новыми разделами и дополнена материалом по комбинированным преобразованиям, повороту вокруг произвольных осей, отражению относительно любой плоскости, косоугольным проекциям (проекциям кабине и кавалье), точкам бесконечности, фотографическим преобразованиям. В главе подробно обсуждаются проблемы создания стереоизображений, а также разбираются методы фиксированного и центрального проецирования.

Глава 4 в настоящей редакции включает развернутое описание коник и методов использования конических сечений.

Глава 5 была полностью переработана. Для достижения большей ясности и наглядности изложения в нее включено детальное рассмотрение вопросов параболического слаживания, кривых Безье и В-сплайнов, а также обсуждение общенных параболических сплайнов. В главе исследуются согласования и деления В-сплайнов, а также периодические uniformные В-сплайны. В главу включено развернутое обсуждение унифицированных и неунифицированных кривых В-сплайна (NURBS).

Глава 6 также претерпела значительные изменения. Теперь она включает изучение линейных и бикубических поверхностей, а также управляемых и динамических поверхностей. В главе нашли отражение сведения о поверхностях вращения, охвата и о квадратических поверхностях Безье и поверхностях B-сплайнов. Рассматриваются вопросы сшивки поверхностей B-сплайна, деления поверхности, а также вводятся понятия гауссовой кривизны и математически правильной поверхности. Достаточно подробно в главе разбираются нерациональные и рациональные, однородные и неоднородные поверхности B-сплайнов (NURBS).

Обширное приложение включает программы машинной графики, описание методов работы с матрицами, псевдокоды, формат файла задания поверхности B-сплайна, задачи, специфики программ и алгоритмы в виде псевдокодов.

Материал книги был тщательно подобран в расчете как на студентов, так и на аспирантов. Книга может быть использована в качестве учебного пособия для полугодового вводного курса машинной графики для студентов или аспирантов. (Двухсеместровый курс основывается на другой книге «Алгоритмические основы машинной графики», которая является естественным продолжением данной.) Именно таким образом авторы и использовали книгу «Математические основы машинной графики». Если в односеместровом курсе требуется изложение более широкого круга вопросов, то можно использовать обе книги. В этом случае предлагаются следующие темы: гл. 1 обеих книг, затем гл. 2 и 3 с отдельными разделами из гл. 4 (например от раздела 4–1 до 4–8) данной книги, далее отдельные разделы гл. 2 (например, от 2–1 до 2–5, 2–7 и от 2–15 до 2–19, 2–22, 2–23 и 2–28), гл. 3 (например, с разд. 3–1, 3–2, 3–4 по 3–6, 3–9, 3–11, 3–15 и 3–16), гл. 4 (например, разд. 4–1, часть разд. 4–2 с алгоритмом отбрасывания невидимых граней, разд. 4–3, 4–4, 4–7, 4–9, 4–11 и 4–13) и гл. 5 (разд. с 5–1 по 5–3, 5–5, 5–6 и 5–14) книги «Алгоритмические основы машинной графики». Д. Роджерс с успехом использовал данный материал для вводного и обзорных курсов. Надеемся, что наша книга пригодится также профессиональным программистам, инженерам и научным работникам. Благодаря детальному изложению, примерам из практики и многочисленным иллюстрациям, книга может быть полезна читателям разной квалификации для самостоятельного изучения. Для усвоения материала достаточно знания математики на уровне средней школы и знакомства с языком программирования высокого уровня.

Возможно, читателю будет интересно узнать, как создавалась книга. Она была набрана Нэнси А. Роджерс в системе обработки текстов ТeX. При работе применялся комплекс шрифтов Computer Modern Roman. Все макроопределения, необходимые для согласования с требованиями издательства, были написаны Дэвидом Ф. Роджерсом. Для работы использовалось два компьютера: IBM AT и Zenith 386. Рукопись была введена непосредственно с бумажной копии. После ее редактирования на экране дисплея были получены с помощью лазерного принтера разрешением 300 точек на дюйм гранки, корректуры страниц и макеты. Окончательный вариант, пригодный для включения иллюстраций, был передан на фотонабор Американского Математического Общества в формате .dvi файлов системы ТeX.

Книга не появилась бы на свет без содействия многих людей. Мы признательны студентам Центра прикладной физики Университета Джона Гопкинса,

которые просмотрели исходные варианты первых пяти глав книги. Многие высказанные студентами предложения и замечания оказались очень полезными.

Отдельно хочется поблагодарить Джона Дилла и Фреда Манчмейера, двух наиболее дорогих и близких нам коллег, прочитавших рукопись целиком с красным карандашом в руках. Многие высказанные ими предложения и замечания способствовали значительному улучшению книги. Особую признательность следует выразить нашей коллеге Линде Эдлам, которая не только прочитала всю рукопись, но и проверила *все* примеры. Огромное спасибо Стивену Д. Роджерсу, прочитавшему рукопись и проверившему примеры из первых пяти глав. Замечания Майка Джайлента об этих главах также были весьма ценными. Очень полезным был обзор работ по кривым и поверхностям Безье, а также кривым и поверхностям В-сплайна, проведенный Билом Гордоном.

Усилия Вирджинии Адамс по макетированию окончательного варианта книги также были очень ценными. Мы благодарны Барбаре Битон за ее неисчерпаемое терпение при консультировании нас по особенностям системы TeX. Специально отдадим должное Джасту Залмстре, чьи выполненные весьма кстати макроопределения для TeX'a существенно облегчили нашу задачу. Традиционно высокий профессионализм был продемонстрирован редактором издательства McGraw-Hill Джимом Брэдли. Наиболее сложные иллюстративные программы проверены Мэлом Хабером. Особую благодарность заслуживает Б.Д. Кларк, который вот уже около двадцати лет является нашим редактором в McGraw-Hill и был весьма внимателен к нашим иногда неортодоксальным идеям.

*Дэвид Ф. Роджерс, Дж. Алан Адамс,
Аннаполис, февраль 1989*

ПРЕДИСЛОВИЕ К ПЕРВОМУ ИЗДАНИЮ

Настоящее время характеризуется появлением новой быстро развивающейся области знания — «машинной графики», которая включает в себя как традиционные сферы знания, так и новые информационные технологии. Теперь компьютеры используются не только крупными фирмами и организациями, уже достигшими первых успехов в области программного и аппаратного обеспечения, но и широким кругом пользователей-непрофессионалов. Такое широкое внедрение компьютеров стало возможно благодаря недорогим графическим дисплеям, режиму разделения времени, а также успехам в области мини- и микрокомпьютерной технологии. Машинная графика является сегодня практическим, надежным, эффективным и широкодоступным инструментом.

Задача данной книги состоит в том, чтобы на основе универсального обобщения изложить математическую теорию основных методов машинной графики. Несмотря на использование в книге новых приемов представления материала, в ней нет совсем незнакомого читателю математического аппарата; в ней предпринята попытка собрать воедино всю информацию по машинной графике, разбросанную по разным техническим изданиям.

При отборе материала для книги мы уделяли особое внимание основным математическим по своей природе методам, а не методам процедурного характера. Именно по этой причине читатель обнаружит в данной книге прежде всего достаточно развернутое изложение способов поворота, преобразования, построения перспективных проекций, описание кривых и поверхностей, а не обсуждение задач отсечения или удаления невидимых линий и поверхностей. Знание математики в объеме первого курса колледжа является достаточной предпосылкой для освоения основной части книги.

После обсуждения в гл. 1 современной компьютерной технологии дается описание матричных операций над графическими элементами, заданными в однородных координатах. В последующих главах содержится описание современных способов представления точек, линий, кривых и поверхностей в цифровых компьютерах. Приводятся также соответствующие программные процедуры манипуляции и вывода графических изображений.

Вместе с математическими методами формирования аксонометрических и перспективных видов приводятся методы поворота, преобразования и масштабирования геометрических фигур. Процедуры определения кривых, заданных явно и в параметрической форме, описываются как для двумерного, так и для трехмерного пространства. Методы определения кривых предусматривают использование конических сечений, круговой интерполяции дугами, кубических сплайнов, параболического сглаживания, кривых Безье и кривых на основе В-сплайнов. В книгу включено также введение в математическое описание поверхностей.

Приложение содержит подпрограммы на языке Бейсик, реализующие алгоритмы большинства базовых модулей интерактивных пакетов машинной графики. Однако эти алгоритмы преднамеренно оканчиваются такими командами или подпрограммами реальной визуализации результатов, которые безусловно не являются стандартными для конкретного графического устройства. Несмотря на то, что в приложении А приводится некое предварительное описание графических примитивов и элементов, пользователь в общем случае сочтет необходимым ориентироваться на специфику конкретной физической конфигурации графических устройств.

Главные концепции данной книги были положены в основу вводного курса машинной графики для студентов высших учебных заведений научно-технической специализации. Курс хорошо подходит как для университетов, так и для технических школ. Книга полезна также и в качестве дополнительного материала для более углубленных учебных курсов по программированию или математике. Кроме этого, она может быть с успехом использована профессиональными программистами для индивидуальной подготовки. В конечном счете хорошо документированные программы могут быть полезны пользователям, заинтересованным в развитии графических возможностей своего компьютера.

БЛАГОДАРНОСТИ

Авторы глубоко признательны Военно-Морской Академии США за поддержку и внимание. Обеспеченная администрацией научная среда, предоставленное оборудование и, особенно, участие курсантов оказало весьма благоприятное воздействие на подготовку материала данной книги.

Ни одна книга не может быть написана без содействия многих людей, и мы хотели бы выразить здесь благодарность некоторым из них. Прежде всего Стиву Коонеу, просмотревшему всю рукопись и сделавшему множество ценных предложений, Ричарду Рейзенфельду, изучившему материал по кривым и поверхностям В-сплайна, профессору Пьеру Безье, который рецензировал материал по кривым и поверхностям Безье, а также Ивану Сазерленду, предложившему нам включить в гл. 3 обсуждение методов трехмерного преобразования. Особой благодарности заслуживают бывшие и нынешние члены группы САПР в Кэмбриджском университете. В первую очередь, это Робин Форест, Чарльз Ланг и Тони Нутбоурл, проявившие глубокое понимание предмета машинной графики. Мы благодарим также Лоя Кноппа, написавшего оригинальную программу на Фортране для кривых В-сплайна.

Авторы хотели бы выразить признательность за содействие многим сотрудникам Evans & Sutherland Computer Corporation и особенно Джиму Каллану, который является автором документации, послужившей основой для многих идей по подготовке изображений, их представлению и интерактивному взаимодействию с ними. Особо отметим Ли Биллоу, подготовившего все векторные рисунки.

Большая часть работы по художественному оформлению гл. 1 была проведена на рабочих станциях различных фирм-производителей. Отдельно выразим благодарность следующим корпорациям:

Рис. 1-3 Evans & Sutherland Computer Corporation

- Рис. 1-5 Adage Inc.
- Рис. 1-7 Adage Inc.
- Рис. 1-8 Vector General, Inc.
- Рис. 1-11 Xynetics, Inc.
- Рис. 1-12 CALCOMP, California Computers Products, Inc.
- Рис. 1-15 Gould, Inc.
- Рис. 1-16 Tektronix, Inc.
- Рис. 1-17 Evans & Sutherland Computer Corporation
- Рис. 1-18 CALCOMP, California Computers Products, Inc.

*Дэвид Ф. Роджерс,
Дж. Алан Адамс*

*Посвящается нашим женам:
Нэнси А. Роджерс и Вирджинии Ф. Адамс
и нашим семьям:
Стивену, Карен и Рэнсому Роджерс
и Линн, Дэвиду и Алану Адамс*

Глава 1

ВВЕДЕНИЕ В МАШИННУЮ ГРАФИКУ

В настоящее время машинная графика представляет собой достаточно развитую отрасль знания, однако некоторые термины и определения по-прежнему трактуются очень широко. Например, такие понятия, как САПР, интерактивная графика, машинная графика, АСТПП, часто используются либо в разных значениях, либо так, что в значительной степени теряется их точный смысл. Наиболее типичным термином такого рода является «САПР», что означает любое применение компьютера для проектирования отдельных деталей, узлов или систем. Однако такое использование компьютера необязательно предусматривает включение машинной графики, ведь процесс проектирования может осуществляться на концептуальном уровне, на уровне деталировки либо ориентироваться на интерфейс с АСТПП.

В АСТПП компьютер используется для управления процессом промышленного производства. Для АСТПП требуется язык программирования, позволяющий задавать геометрию и траекторию перемещения обрабатывающего инструмента, например язык АРТ (язык программирования станков с ЧПУ), для генерирования команд управления инструментом станка. Контроллер станка, как правило, представляет собой микро- или миникомпьютер, для которого САПР может непосредственно формировать необходимые команды управления. Одновременно можно создавать массив данных в стандартном формате, например IGES (формат передачи графических данных). Для преобразования полученных данных в формат команд конкретного станка используются специальные программы. На рис. 1-1 показан типичный обрабатывающий центр с ЧПУ и контроллером.

Компьютер в машинной графике используется для описания изображений, их хранения, манипулирования ими, их вызова и визуализации, т. е. в основном для пассивных операций. Компьютер выводит на экран предварительно подготовленное изображение, и наблюдатель не может непосредственно управлять изображением, прежде чем оно появится на экране. В каждом конкретном случае изобра-

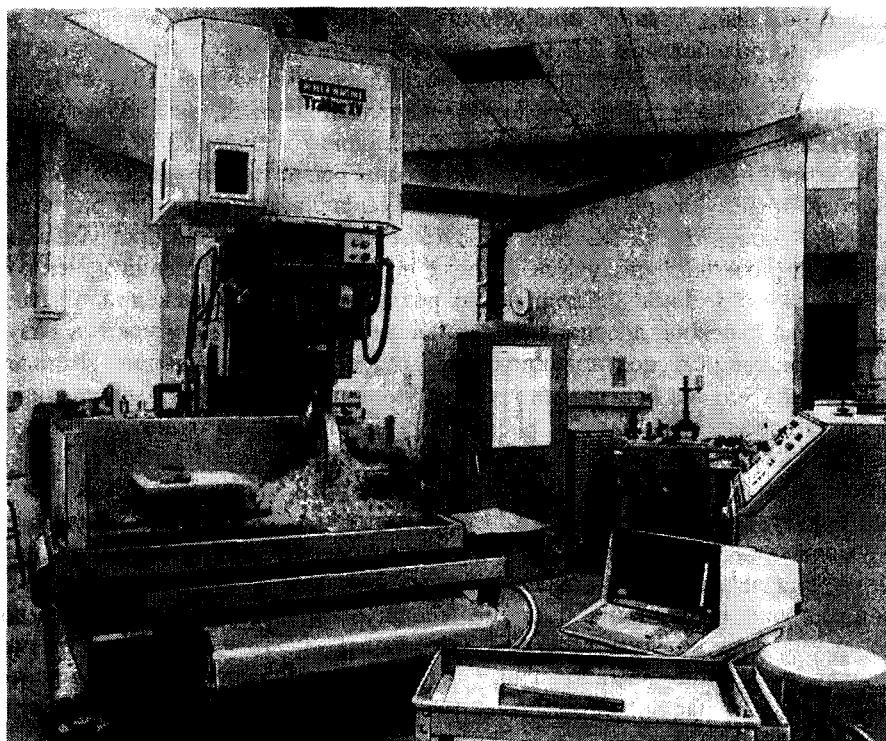


Рис. 1-1 Обрабатывающий центр с ЧПУ.

жение может быть либо таким простым, как, например, графики простых функций, либо столь сложным, как визуализация результатов моделирования процесса автоматического полета и посадки самолета или космического корабля.

В динамической интерактивной машинной графике (в дальнейшем просто интерактивной графике) компьютер используется для подготовки и визуализации графических данных. Однако интерактивная графика позволяет наблюдателю в реальном масштабе времени влиять на весь процесс представления изображения. Чтобы убедиться в серьезности ограничений, определяемых условиями работы в реальном времени, рассмотрим задачу вращения объемного изображения, состоящего из 1000 линий, со скоростью 15 град/с.

Как мы увидим в дальнейшем, картинка из 1000 линий в большинстве случаев представляется матрицей 1000×4 однородных координат конечных точек линий. Вращение, как правило, представляется путем умножения этой матрицы на матрицу преобразования размером 4×4 . Для проведения такого матричного умножения требуется выполнить 16 тыс. операций умножения, 12 тыс. операций сложения и 1 тыс. операций деления вещественных чисел. Если умножение осуществляется программно, то для того чтобы выполнить все операции вращения, может потребоваться значительное время. Обычный универсальный компьютер

с процессором вещественной арифметики, способным выполнять операции умножения с быстродействием 3.6 мкс, сложения — за 2.6 мкс и деления со скоростью 5.2 мкс, может выполнить такое матричное умножение за 0.1 с. Поскольку для того, чтобы картинка на экране начала двигаться, она должна перерисовываться не менее 30 раз в секунду, то становится очевидным, что в случае с данным примером картинка не сможет непрерывно перемещаться по экрану. Даже если предположить, что изображение обновляется только 15 раз в секунду, т. е. при повороте на каждый градус, то и в этом случае программная реализация не позволяет получить эффект непрерывного вращения и исключает возможность получения действительно реалистичных изображений. Существует множество способов достижения реального интерактивного режима визуализации изображения: использование более мощного дополнительного компьютера или более тщательное программирование, уменьшающее время выполнения матричного умножения. Однако в любом случае обязательно будет достигнут момент, когда всякое дальнейшее улучшение станет уже невозможным. Уменьшение сложности изображения также не даст нужного эффекта, так как при этом может оказаться неприемлемым качество картинки. Однако операция матричного умножения, необходимая для манипулирования изображением из нашего примера, для действительно очень сложных картинок может осуществляться на уровне микропрограмм или с помощью специализированных цифровых умножителей, применение которых всегда было наиболее эффективным и экономичным. На основе введенной терминологии в следующих разделах главы предлагается обзор машинной графики и обсуждаются различные типы современных графических устройств.

1-1 ОБЗОР МАШИННОЙ ГРАФИКИ

Машинная графика — это сложная и многосторонняя дисциплина. Поэтому для изучения ее необходимо разделить на более простые обозримые части, учитывая при этом, что конечным результатом машинной графики является изображение, которое, разумеется, можно использовать для самых различных целей: технического чертежа, деталировки в справочном руководстве, изображения архитектурной конструкции, как одного из возможных вариантов, так и эскиза проекта, рекламной иллюстрации либо отдельного кадра из мультифильма. В машинной графике фундаментальным связующим звеном является изображение, следовательно, важно разобрать следующие вопросы:

- представление изображений в машинной графике;
- подготовка изображений для вывода;
- визуализация предварительно подготовленных изображений;
- взаимодействие с изображением.

В дальнейшем под термином «изображение» мы будем понимать множество линий, точек, текстов и т. п., выводимых на графическое устройство.

1-2 ПРЕДСТАВЛЕНИЕ ИЗОБРАЖЕНИЙ

Многие алгоритмы интерпретируют изображения как набор многоугольников или ребер, причем каждый многоугольник или ребро в свою очередь может быть представлен с помощью точек (вершин). Таким образом, точки являются основными строительными блоками изображения. Не менее важны также и алгоритмы размещения таких точек. В качестве иллюстрации рассмотрим единичный квадрат в первом квадранте координатной плоскости. Этот квадрат может быть представлен своим четырьмя вершинами (см. рис. 1-2) $P_1(0, 0)$, $P_2(1, 0)$, $P_3(1, 1)$, $P_4(0, 1)$. Соответствующая алгоритмическая нотация может выглядеть следующим образом:

Последовательно соединить $P_1P_2P_3P_4P_1$.

Единичный квадрат можно также представить с помощью его четырех ребер $E_1 = P_1P_2$, $E_2 = P_2P_3$, $E_3 = P_3P_4$, $E_4 = P_4P_1$. Здесь алгоритмической нотацией служит следующая запись:

Последовательно изобразить $E_1 \ E_2 \ E_3 \ E_4$.

И наконец, для описания единичного квадрата в виде многоугольника можно использовать либо точки, либо ребра. Например, $S_1 = P_1P_2P_3P_4P_1$, или $S_1 = P_1P_4P_3P_2P_1$ или $S_1 = E_1E_2E_3E_4$. В зависимости от размерности координатного пространства основные строительные блоки (точки) можно представить либо как пары, либо как тройки чисел. Таким образом, координаты (x_1, y_1) или (x_1, y_1, z_1) определяли бы точку на плоскости или в пространстве. Две точки задавали бы отрезок или ребро, а совокупность из трех и более точек — многоугольник. Построение кривых осуществляется обычно путем их аппроксимации короткими отрезками. Символы представляются уже более сложным образом — они образуются из совокупности различных кривых или определяются матрицей точек. Однако в

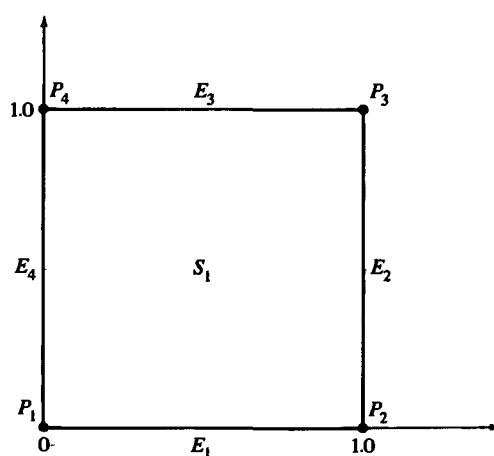


Рис. 1-2 Описание данных изображения.

основе представления символов по-прежнему лежат совокупности отрезков, точек и соответствующие им алгоритмы. Если не надо создавать специальные шрифты или проектировать графические устройства, то пользователю совсем необязательно знать все детали описания символов текста, так как большинство графических устройств имеют встроенные аппаратные или программные генераторы символов.

1-3 ПОДГОТОВКА ИЗОБРАЖЕНИЙ ДЛЯ ВЫВОДА

В конечном счете изображение определяется точками и алгоритмом их вывода. Прежде чем информация об изображении будет использована для формирования картинки, она хранится в файле, называемом базой данных. Для представления очень сложных изображений нужны сложные базы данных, для которых предполагается использование сложных алгоритмов организации доступа к ним. Данные или структуры данных в таких базах могут быть организованы в виде кольца, В-дерева, квадратичных структур и т. д. Сами базы данных могут содержать указатели, подструктуры и другую негеометрическую информацию. Организация баз данных и алгоритмы доступа к ним выходят за рамки изложения данной книги. Однако многие конкретные реализации систем с машинной графикой предназначены для работы с простыми изображениями, для которых можно предложить простые структуры данных с простым способом доступа. Наиболее элементарной структурой данных, безусловно, является линейный список. Может показаться неожиданным, но оказалось, что с помощью этой структуры представляются достаточно качественные изображения.

Точки – основные строительные блоки графической базы данных, именно поэтому особый интерес представляют основные операции преобразования точек. Существуют три типа операций, в которых точки используются как графические примитивы: во-первых, перемещение пера и курсора, след курсора (в дальнейшем просто курсор), невидимая точка; во-вторых, изображение линии между двумя точками; в третьих, вывод точки в заданном месте. В общем случае местоположение точки можно описать двумя способами: с помощью абсолютных и относительных (инкрементальных) координат. В системе относительных координат позиция точки определяется путем задания ее положения относительно предыдущей точки. Все программное обеспечение машинной графики базируется на этих основных понятиях и операциях. Более подробно основы построения программ машинной графики излагаются в разд. 1-22 и в приложении А.

Местоположения точки могут быть описаны с помощью вещественных (с плавающей точкой) или целых чисел. В случае применения целых чисел возникает множество проблем, связанных с ограниченной длиной слова компьютера. Для представления целых координат, как правило, используется полное машинное слово. Наибольшее целое число, занимающее полное машинное слово и необходимое для представления величин со знаком из положительного и отрицательного диапазонов, равно $2^{n-1} - 1$, где n – количество битов в слове. Для 16 битового слова это число равно 32 767. Для многих приложений этого бывает вполне достаточно, однако возникают серьезные проблемы, когда требуются большие числа, которые невозможно представить с помощью одного слова компьютера. Эту трудность можно преодолеть, например, воспользовавшись относительными координатами

в случае, когда их величина не превышает 60 000. При этом в абсолютной системе координат устанавливаются точки с координатами (30 000, 30 000), а затем с помощью относительных координат (30 000, 30 000) строится точка с координатами (60 000, 60 000). Однако этот способ обладает значительным недостатком: попытка сохранить относительные координаты, превышающие максимально допустимую величину, вызывает переполнение. В большинстве компьютеров результатом целочисленного переполнения является генерация чисел с противоположным знаком и неверным абсолютным значением.

От этого недостатка можно избавиться с помощью однородных координат, что вызывает некоторые усложнения, уменьшение скорости выполнения и понижение качества разрешения изображения. Однако все эти недостатки оправдываются преимуществами, которые дают возможность представления больших чисел на компьютерах с ограниченной длиной слова. Именно по этой причине, а также в связи с другими преимуществами использования однородных координат, им уделяется основное внимание в данной книге.

В системе однородных координат n -мерное пространство представляется $n + 1$ измерением, т. е. трехмерные координаты, однозначно задаваемые тройкой (x, y, z) , выражаются через четыре координаты (hx, hy, hz, h) , где h — произвольный множитель. Если все координатное пространство в 16-битовом компьютере представляется числами меньше 32 767, то h будет равно 1, а координаты будут представляться путем прямых преобразований. Однако если хотя бы одна из координат превышает число 32 767, например, $x = 60 000$, то преимущества использования однородных координат становятся очевидны. В данном случае мы установили $h = 1/2$ и координаты точки теперь можно представить в 16-битовом компьютере как $(30 000, y/2, z/2, 1/2)$. Однако в этом случае неизбежно снижение разрешающей способности, так как, например, значения $x = 60 000$ и $60 001$ теперь представляются одинаковой однородной координатой. Фактически разрешение ухудшается для всех координат даже в том случае, если только одна из них превышает максимально допустимое для целых чисел значение.

1-4 ПРЕДСТАВЛЕНИЕ ПРЕДВАРИТЕЛЬНО ПОДГОТОВЛЕННОГО ИЗОБРАЖЕНИЯ

Данные, используемые для подготовки рисунка, как правило, не совпадают с данными, необходимыми для его визуализации (дисплейным файлом). В этом файле содержится некоторая часть, вид или отдельная сцена целого изображения, представленного в общей базе данных. Выводимое изображение обычно формируется путем поворота, переноса, масштабирования и вычисления различных проекций. Эти преобразования обычно выполняются с помощью матричных операций (4×4) с данными, представленными в однородных координатах (гл. 2 и гл. 3). В случае, когда требуется последовательность преобразований, матрица каждого отдельного преобразования последовательно применяется к точкам рисунка. Однако если таких точек очень много, данный метод становится неэффективным. Гораздо удобнее другой способ, состоящий в использовании одной результирующей матрицы комбинированных преобразований, полученной путем умножения матриц каждого отдельного преобразования. Такая матричная операция называется кон-

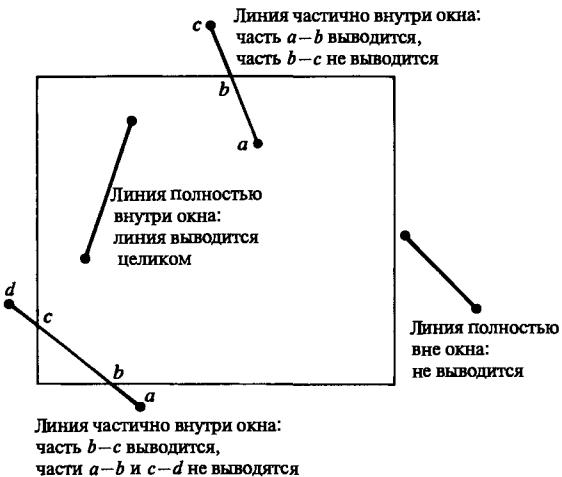


Рис. 1-3 Двумерное отсечение.

катенацией. После этого вектор координат всех точек изображения умножается на матрицу комбинированного преобразования размером 4×4 для получения результирующих точек. Данный подход позволяет значительно сократить время проведения сложных матричных операций над множеством точек.

Удаление невидимых линий или поверхностей, закраска, определение прозрачности, текстуры или световые эффекты — все эти операции могут быть выполнены еще до момента окончательной визуализации изображения. Если выводимое изображение определено не на всей базе данных, в ней необходимо выделить некоторую часть, соответствующую этому изображению. Данный процесс называется отсечением. Отсечения бывают дву- и трехмерные. В ряде случаев окно и объем отсечения могут иметь отверстия или быть неправильной формы. Стандартное отсечение по границам дву- или трехмерных областей часто реализуется аппаратно. Подробное обсуждение особенностей реализации эффектов отсечения выходит за рамки данного изложения. Более детальное описание приведено в работе [1-1].

С процессом визуализации изображений обычно связано два ключевых понятия: окно и поле зрения. Процесс вывода с использованием окон предусматривает извлечение части графических данных из базы путем их отсечения по границам окна. Программная реализация операций отсечения и вывода окна, как правило, требует значительного времени, что исключает возможность ее применения в интерактивных графических системах. С другой стороны, в сложных графических системах для этого используется специализированная аппаратура или микропрограммные средства. Задача отсечения изображения включает выделение и отбрасывание отрезков или их частей, лежащих вне окна. Отсеченные отрезки или их части не передаются на графическое устройство вывода.

На плоскости окно задается левым, правым, верхним и нижним ребрами ограничивающего прямоугольника. Координаты вершин ребер определяются в системе координат пользователя или в мировой системе координат, т. е. именно в

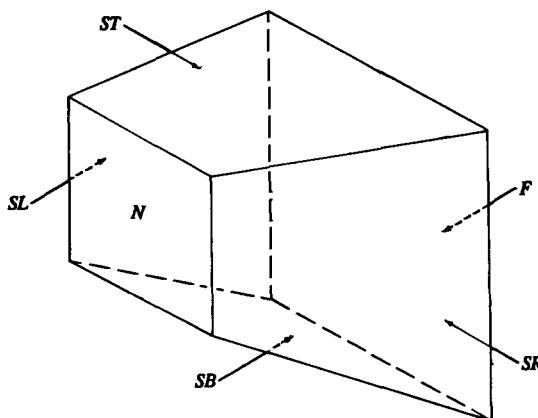


Рис. 1-4 Трехмерная пирамида видимости.

той системе, в которой хранится вся информация из графической базы данных. Как правило, координаты геометрических примитивов запоминаются в виде вещественных чисел.

Процесс отсечения значительно упрощается в случае так называемых регулярных окон, ребра которых параллельны осям координат. Нерегулярные отсекающие окна также представляют определенный интерес для многих конкретных приложений [1-1]. На рис. 1-3 показан процесс отсечения на плоскости. Линии удаляются, выводятся на экран полностью или частично в зависимости от их местоположения относительно границ окна. В трехмерном пространстве регулярное окно или отсекающий объем представляет собой прямоугольный параллелепипед (брюс) или, для перспективных видов, усеченную пирамиду (пирамиду видимости). На рис. 1-4 показана типичная пирамида видимости, ближайшая к наблюдателю грань помечена символом N , наиболее удаленная — символом F , а стороны образованы гранями SL , SR , ST и SB .

Поле зрения — это область на экране дисплея, предназначенная для размещения попавшего в окно изображения. Регулярное поле зрения на плоскости задается левым, правым, верхним и нижним ребрами ограничивающего прямоугольника. Поле зрения может быть определено в реальных физических координатах устройства, часто задаваемых в виде целых чисел. Координаты поля зрения могут быть нормализованы по некоторому произвольному диапазону, например, $0 \leq x \leq 1$, $0 \leq y \leq 1$ и задаваться в виде вещественных чисел. Содержание одного окна может быть выведено в нескольких полях зрения одного и того же устройства (рис. 1-5). Сохранение пропорций окна и поля (полей) зрения позволяет избежать искажений. Отображение на поле зрения графических данных, попавших после отсечения в окно, предусматривает выполнение операций перевода и масштабирования (приложение А).

Кроме графической информации, большинство изображений могут содержать алфавитно-цифровые и символические данные. Существуют два основных способа генерации символов: программный и аппаратный. Если при программной ге-

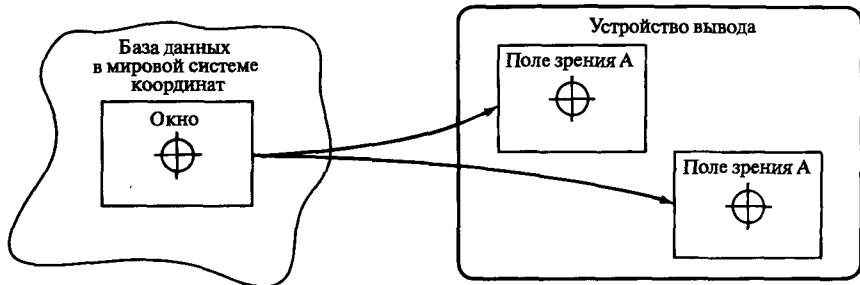


Рис. 1-5 Несколько полей зрения для одного окна.

нерации символов используются линии, то они отсекаются обычным образом. Это бывает необходимо в случае, когда символы подвергаются операции отсечения, а затем операции преобразования. Однако многие графические устройства имеют встроенные аппаратные генераторы символов, и в этом случае символы идентифицируются только своим кодом, а их изображение формируется непосредственно перед выводом на экран. Данная реализация намного эффективнее, но обладает меньшей гибкостью, чем программная генерация символов, так как не позволяет проводить отсечение и выполнять общие преобразования, ограничивающиеся в этом случае поворотом и масштабированием.

При использовании аппаратного генератора управляющая программа графического устройства должна установить размер, ориентацию и точку привязки символа или строки текста. После этого в дисплейный файл заносятся коды символов, имеющих установленные характеристики. Затем запускается генератор текстов, который интерпретирует строку текста, задавая аппаратуре всю информацию, необходимую для изображения каждого символа. После этого символы текста персылаются на устройство вывода.

1-5 ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ С ИЗОБРАЖЕНИЕМ

После того как изображение на экране построено, возникает необходимость как-то взаимодействовать с ним или модифицировать его. Для этого был разработан ряд диалоговых устройств. Среди них можно назвать планшет, световое перо, рычаг, мышь, ручки для ввода скалярных величин, функциональные переключатели или кнопки и, разумеется, обычную алфавитно-цифровую клавиатуру. Прежде чем перейти к обсуждению этих физических устройств, рассмотрим функциональные возможности диалоговых графических устройств. Обычно насчитываются четыре или пять функциональных логических классов [1-2] — [1-4]. Логическими диалоговыми устройствами являются локатор, валютор, селектор и кнопка. Из-за широкой распространенности алфавитно-цифровой клавиатуры ее часто выделяют в пятый класс, называемый клавиатурой. В действительности клавиатуру можно концептуально и функционально считать набором кнопок.

Функцией локатора является выдача координатной информации в двух или трех измерениях. Обычно выдаваемые значения координат являются нормализо-

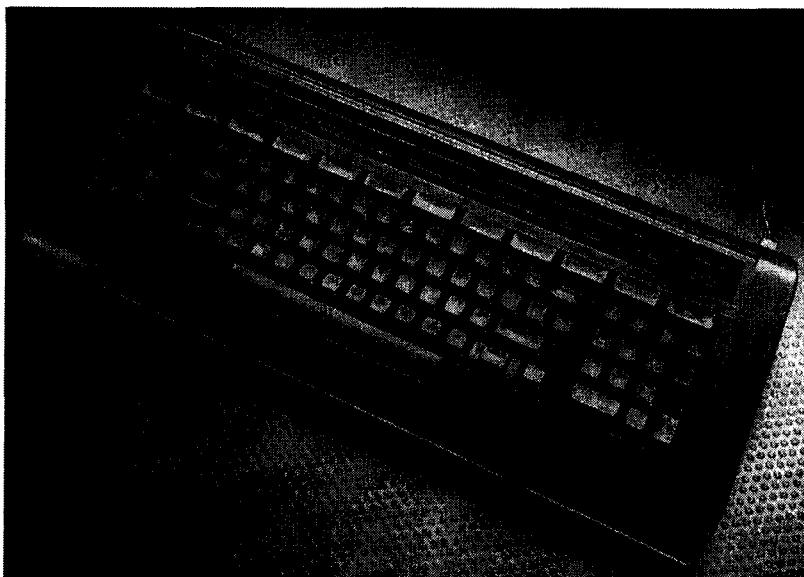


Рис. 1-6 Алфавитно-цифровая клавиатура. (С разрешения фирмы Evans & Sutherland Computer Corp.)

ванными и могут быть как относительными, так и абсолютными. Валюатор применяется для ввода одиночной величины. Обычно это вещественное число между нулем и некоторым вещественным максимумом. Кнопка используется для выбора и активирования событий или процедур, управляющих сценарием диалога. Кнопка обычно представляет двоичную («включено» или «выключено») цифровую информацию. В функцию селектора входит идентификация или выбор объектов или подкартинок в выведенном изображении. Логическая клавиатура обрабатывает текстовую информацию. На рис. 1-6 показана типичная клавиатура.

Наиболее общим устройством класса локаторов является планшет (рис. 1-7). Планшеты можно использовать либо отдельно, либо в комбинации с графическим дисплеем на ЭЛТ. В первом случае их часто называют оцифровывателями. Сам по себе планшет состоит из плоской поверхности и карандаша (похожего на обычный карандаш), который используется для указания точки на поверхности планшета. Обычно положение карандаша распознается, если он находится на небольшом расстоянии от планшета. При работе с ЭЛТ обратная связь на экране обеспечивается с помощью небольшого символа (курсора), отслеживающего перемещение карандаша по поверхности планшета. В случае когда обратная связь используется в качестве отдельного оцифровывателя, она обеспечивается с помощью цифровых отсчетов.

Планшеты выдают координатную информацию в двух или трех измерениях. На рис. 1-8 показан трехмерный планшет. Выдаваемые планшетом значения представлены в координатах устройства. Они программно преобразуются в пространство координат пользователя. Обычное разрешение и точность составляют от 0.025 до 0.0025 см.

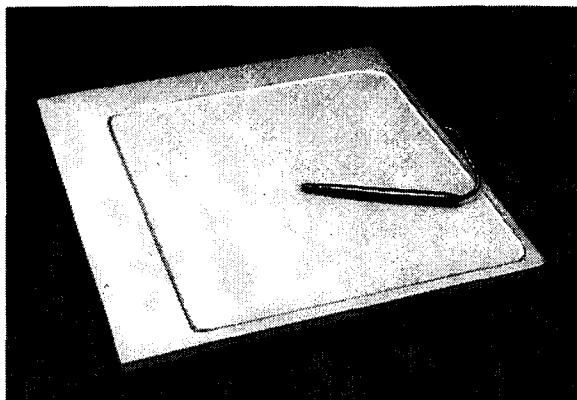


Рис. 1-7 Типичный планшет. (С разрешения фирмы Adage.)

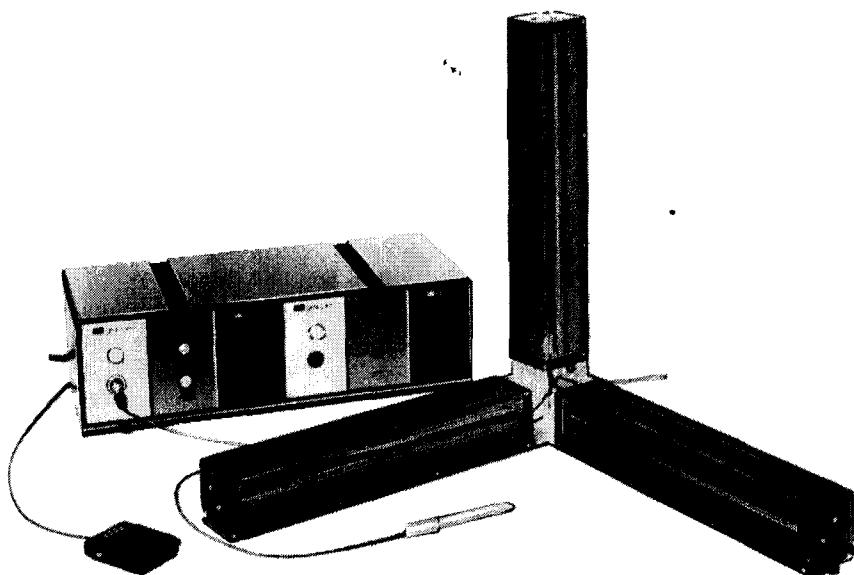


Рис. 1-8 Трехмерный звуковой планшет. (С разрешения фирмы Science Accessories Corp.)

При создании планшетов используется несколько различных принципов. В первом планшете RAND [1-5] применялась ортогональная сетка из отдельных проводов, расположенных под поверхностью. Каждый провод кодируется таким образом, чтобы карандаш, действующий как приемник информации, в каждой точке пересечения проводов получал уникальный цифровой код. Декодирование этого кода дает координаты карандаша x , y . Очевидными ограничениями на разрешение таких матрично кодируемых планшетов является плотность размеще-

ния проводов и способность приемника обрабатывать уникальный код. Точность ограничивается линейностью конкретных проводов, а также их параллельностью в двух ортогональных направлениях.

В одной достаточно интересной реализации планшета применяются звуковые волны. Карандаш используется для создания искры электрического разряда, вызывающей звуковую волну. Она движется во всех направлениях от карандаша по поверхности планшета, образуя круговой звуковой фронт. На его краях в перпендикулярных направлениях смонтированы два чувствительных ленточных микрофона. Координатные расстояния можно определить с помощью точного измерения времени, за которое волна проходит с карандаша до микрофонов. Данную методику можно распространить на три измерения (рис. 1-8).

Наиболее популярны конструкции планшета, в основе которых лежит электромагнитный принцип. При этом электрические импульсы проходят через сетку из проводов, расположенных под поверхностью планшета. Время, за которое чередующиеся импульсы, параллельные координатным осям x , y , проходят от края планшета до карандаша, определяется с помощью карандаша и соответствующих индикаторов. Эти данные легко преобразуются в координаты x , y .

Сенсорная панель похожа на планшет и относится к классу локаторов. В типичной сенсорной панели на двух смежных сторонах расположены источники света, а на двух противоположных смежных сторонах смонтированы светочувствительные элементы. Любой предмет, например палец, прерывая два ортогональных луча света, позволяет определить пару координат x , y . Из-за низкого разрешения этот прибор лучше всего использовать для грубых операций указания. С этой целью сенсорную панель часто монтируют поверх экрана ЭЛТ.

В таких локаторных устройствах, как рычаг, шар и мышь, в качестве делителя напряжения часто используются чувствительные переменные резисторы или потенциометры. Аналогичным образом реализуются и ручки для ввода скалярных величин, которые являются устройствами класса валюаторов. Точность всех этих устройств зависит от качества потенциометра и обычно колеблется от 0,1 до 10% всего диапазона измерения. Несмотря на то что разрешение потенциометра по существу бесконечно, его использование в цифровых системах требует аналого-цифрового (АЦ) преобразования. Обычно разрешение АЦ преобразователя изменяется в диапазоне от 8 до 14 бит, т. е. от 1/256 до 1/16384. Валюаторы также реализуются с помощью цифровых преобразователей углового положения, которые для каждого инкрементального угла поворота ручки выдают цифровой результат. Типичное разрешение — от 1/256 до 1/1024.

Типичным валюатором является рычаг (рис. 1-9). Подвижный рычаг обычно оборудуется двумя валюаторами — либо потенциометрами, либо преобразователями углового положения, смонтированными на основании прибора. Валюаторы выдают результаты, пропорциональные смещению ручки. В рычаг можно просто добавить третью степень свободы, используя, например, третий валюатор для измерения угла поворота. В качестве эха обратной связи рычага обычно употребляется графический курсор.

Шар во многом аналогичен рычагу (рис. 1-10). Чаще всего его можно встретить в радарных установках, например в системах управления воздушными перевозками. Сферический шар смонтирован в основании прибора, причем над по-

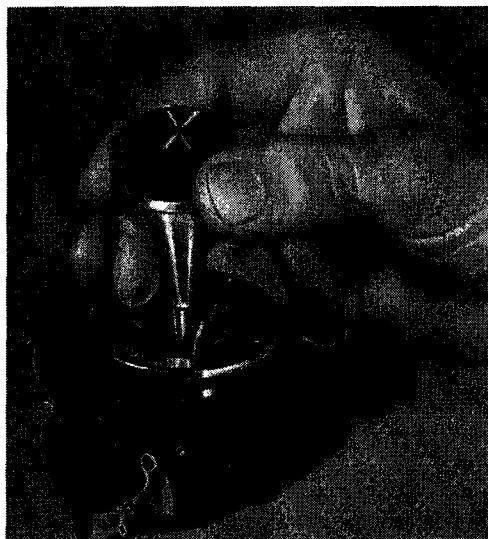


Рис. 1-9 Рычаг. (С разрешения фирмы Measurement Systems, Inc.)

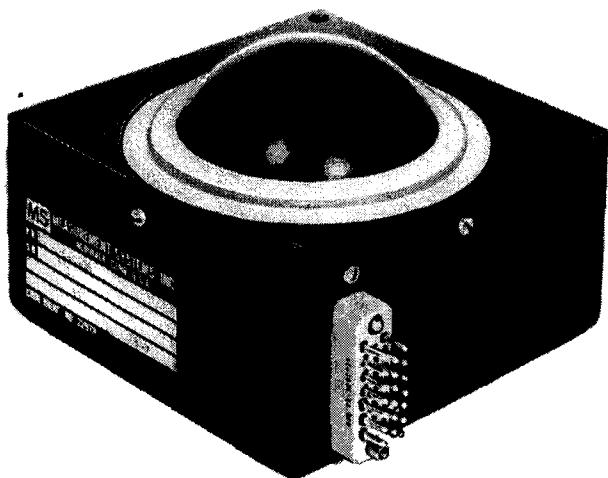


Рис. 1-10 Шар. (С разрешения фирмы Measurement Systems, Inc.)

верхностью возвышается только его часть. Шар может свободно вращаться в любом направлении. В основании смонтированы два валюатора (потенциометры или преобразователи углового перемещения), которые улавливают поворот шара и выдают результаты, пропорциональные относительным величинам углов. Кроме обычной обратной связи через курсор, пользователи получают тактильную обратную связь в виде скорости поворота или углового импульса вращения шара.

Рычаг имеет фиксированное положение с фиксированным началом координат. Мышь и шар, напротив, характеризуются только относительным положением ко-



Рис. 1-11 Мышь. (С разрешения фирмы Apple Computer, Inc.)

ординат. Обычная мышь состоит из шара, касающегося поверхности и расположенного в маленьком, легком футляре. При движении мыши по поверхности шар вращается и управляет осями двух валюаторов (потенциометров или преобразователей углового перемещения). Совокупное движение осей дает координаты x , y . Типичная мышь показана на рис. 1-11. Мышь можно поднять, переместить в любом направлении и поставить на поверхность, по-другому ориентировав. В этом случае изменяется система координат, в которой генерируются данные, т. е. меняется положение мыши, но не система координат, связанная с данными. Используемый для обратной связи курсор не движется, когда мышь не контактирует с поверхностью. Из-за проскальзывания шара возникает уменьшение точности. Совсем недавно в продаже появились устройства, работающие на оптическом или на магнитном принципе. В обоих случаях исключаются неточности, связанные с проскальзыванием.

Возможно, простейшими из валюаторов являются ручки для ввода скалярных величин. Такие ручки, показанные на рис. 1-12, являются по существу чувствительными вращающимися потенциометрами или точными цифровыми преобразователями углового положения, которые обычно объединяются в группы. Такие устройства особенно удобны при задании параметров для функций поворота, перевода или масштабирования.

Кнопки и функциональные переключатели, показанные на рис. 1-13, являются переключателями рычажного или нажимного действия. Они могут быть постоянно нажаты, постоянно отжаты либо иметь кратковременный контакт. Наиболее удобный тип функционального переключателя объединяет в себе обе эти возможности. Обычно можно программно управлять световыми индикаторами, указывающими, какие переключатели или кнопки в данный момент активны. Кнопки

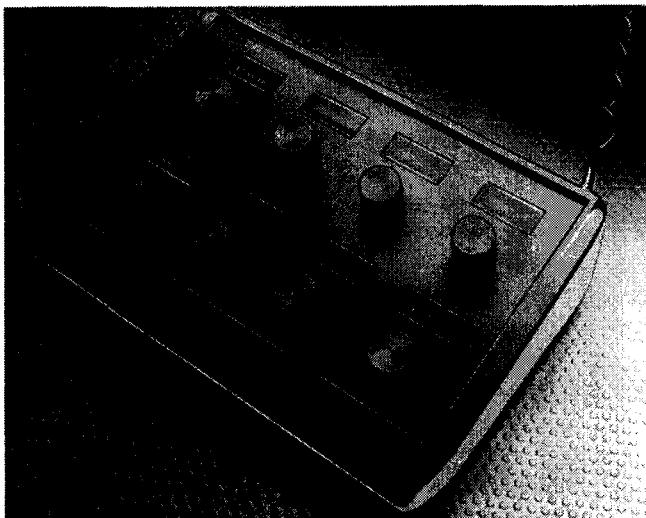


Рис. 1-12 Набор ручек для ввода скалярных значений. (С разрешения фирмы Evans & Sutherland Computer Corp.)

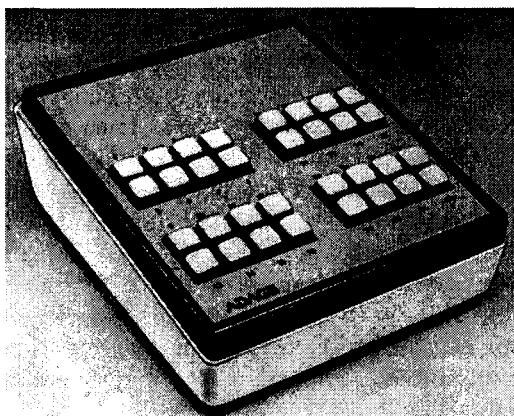


Рис. 1-13 Функциональные кнопки. (С разрешения фирмы Adage.)

и переключатели часто входят в состав других устройств. Например, на тонком конце карандаша планшета обычно располагается переключатель, срабатываемый при нажатии карандаша. На мыши также находится одна или несколько кнопок.

Единственным подлинным устройством указания является световое перо (рис. 1-14), которое содержит светочувствительный фотоэлемент и соответствующую электрическую цепь. Основная информация, передаваемая световым пером, связана с синхронизацией и зависит от того, что изображается на экране, и от порядка, в котором происходит его заполнение. Это исключает использование све-

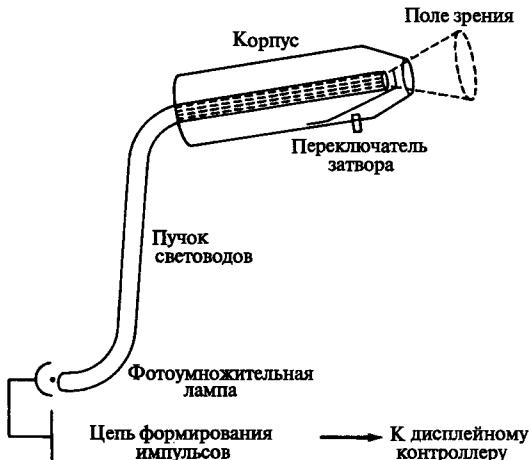


Рис. 1-14 Схема светового пера.

тowego пера с дисплеями на запоминающей ЭЛТ (гл. 1-7). Его можно использовать только при дисплеях с регенерацией — векторных или растровых.

В тот момент, когда активизированное световое перо оказывается над той областью ЭЛТ векторного дисплея (гл. 1-8), в которой происходит рисование, дисплейному контроллеру посыпается сигнал. Он позволяет определить конкретную команду в дисплейном буфере, выполняемую в данный момент времени, что дает возможность идентифицировать отрезок, объект или подкартинку на экране. Используя графический курсор, можно применять световое перо в качестве локатора для векторного устройства с регенерацией изображения.

Так как в растровом дисплее изображение генерируется в фиксированной последовательности, световое перо используется для определения горизонтальной сканирующей строки (координаты y) и позиции в этой строке (координаты x). Отслеживая программу в контроллере, можно снова определить, какой отрезок, объект или подкартинка были указаны. Этот метод несколько усложняется в случае чересстрочной развертки (гл. 1-12). Приведенное выше описание означает также, что для растрового устройства световое перо может быть использовано скорее всего как локатор, а не как устройство указания.

Несмотря на то что для всех логических устройств существуют соответствующие физические устройства, в конкретной реализации некоторые из них могут отсутствовать. Таким образом, необходимо уметь моделировать логические интерактивные устройства ввода. На рис. 1-15 приведен пример, в котором световое перо используется для моделирования функций логической кнопки путем выбора из меню световых кнопок.

Планшет — одно из самых универсальных физических устройств. Его можно использовать как оцифровыватель для получения координат x , y . В добавок к этому оно прекрасно подходит для моделирования всяких логических интерактивных функций (рис. 1-16). Из этого же рисунка видно использование планшета в качестве локатора (позиция (а) на рис. 1-16). Используя графический курсор,

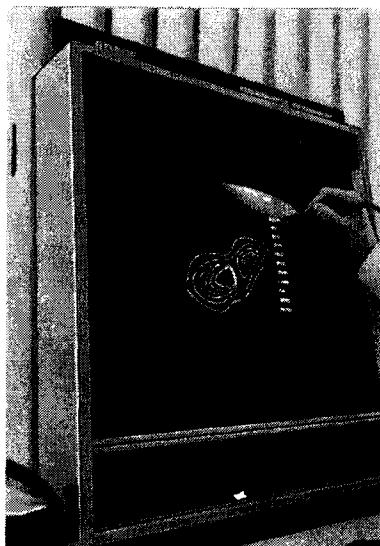


Рис. 1-15 С помощью выбора из меню световое перо используется для моделирования функции логической кнопки. (С разрешения фирмы Adage.)

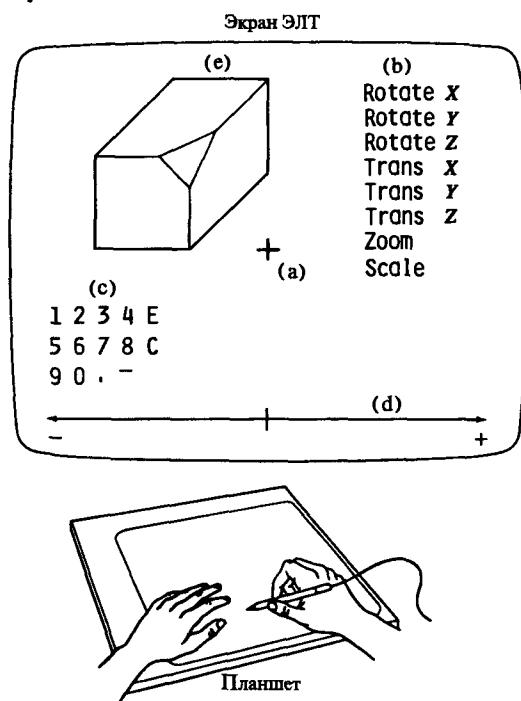


Рис. 1-16 Использование планшета для моделирования всех логических интерактивных функций: (а) локатора, (б) кнопки, (с) клавиатуры, (д) валюатора, (е) селектора.

можно реализовать функцию кнопки. Для этого с помощью карандаша планшета указываем курсором на кнопку меню (либо вблизи нее). Координаты, полученные с планшета, сравниваются с известными координатами x, y кнопок меню. Если обнаружено совпадение, то соответствующая кнопка активизируется (позиция (b) на рис. 1-16). Аналогичным образом на планшете можно реализовать клавиатуру (позиции (c) на рис. 1-16).

Отдельный валюатор обычно реализуется в сочетании с кнопкой. Конкретная функция для определения значения выбирается кнопкой, как правило, из меню. Затем валюатор можно смоделировать с помощью «числовой оси» (позиция (d) на рис. 1-16). Перемещение курсора вдоль оси генерирует координаты x, y , одна из которых интерпретируется как процентное отношение к диапазону валюатора.

Функцию указания можно реализовать с помощью локатора путем определения относительных координат x и y небольшого «окна указания». Данное окно становится затем графическим курсором, и для его позиционирования используется карандаш планшета. Координаты x, y каждого из интересующих отрезков, объектов или подкартинок сравниваются затем с текущим местоположением. Если они попадают в пределы окна, то данный элемент считается указанным. Для сложных изображений программная реализация данного алгоритма может оказаться медленной, но при аппаратной реализации заметной задержки не будет. Несмотря на то что световое перо нельзя использовать в качестве оцифровывателя, его, подобно планшету, тоже можно применять для моделирования всех логических интерактивных функций.

1-6 ТИПЫ ГРАФИЧЕСКИХ УСТРОЙСТВ

Существует много разнообразных устройств для вывода изображений, построенных с помощью машинной графики. Наиболее типичными являются электронно-лучевые трубы (ЭЛТ), плазменные дисплеи, перьевые графопостроители, матричные, электростатические и лазерные печатающие устройства, устройства для фото и киносъемки. Кроме устройств визуализации большое развитие получают устройства сохранения изображения.

Запоминающие ЭЛТ с прямым копированием изображения (рисование отрезками), векторные дисплеи (рисование отрезками) и растровые сканирующие дисплеи с регенерацией (поточечное рисование) образуют три основных класса дисплеев на ЭЛТ. Развитие электронной технологии позволило использовать в одном дисплее несколько способов изображения. Класс устройств с плоским экраном образуют газоразрядные плазменные, электролюминесцентные дисплеи, дисплеи на жидкких кристаллах и дисплеи, построенные на основе светодиодов. При обсуждении различных дисплеев мы стоим на пользовательской или концептуальной точке зрения, т. е. в основном рассматриваем функциональные возможности дисплеев, а не особенности их электронной схемы.

1-7 ГРАФИЧЕСКИЕ ДИСПЛЕИ НА ЗАПОМИНАЮЩЕЙ ТРУБКЕ

Из всех дисплеев на ЭЛТ наиболее просто устроен дисплей на запоминающей ЭЛТ с прямым копированием изображения. Запоминающую ЭЛТ, называемую

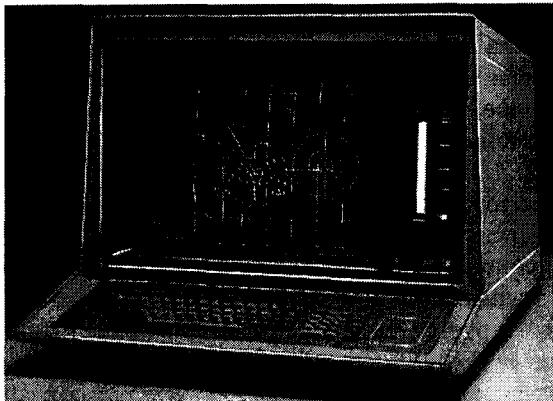


Рис. 1-17 Графический дисплей на запоминающей трубке. (С разрешения фирмы Tektronix.)

также бистабильной запоминающей трубкой, можно рассматривать как ЭЛТ, покрытую люминофором с длительным временем послесвечения. Линия или символ остаются на ней видимыми в течение длительного времени (до одного часа), прежде чем станут окончательно неразличимыми. На рис. 1-17 показан типичный дисплей такого типа. Чтобы нарисовать отрезок на дисплее, интенсивность электронного луча увеличивают до такой величины, которая вызывает запоминание следа луча на люминофоре. Для стирания изображения на всю трубку подают специальное напряжение, снимающее свечение люминофора. Экран вспыхивает и принимает исходное (темное) состояние. Весь процесс стирания занимает около $1/2$ сек. Поскольку вспыхивает вся трубка, то стираются все отрезки и символы. Таким образом, стереть отдельные линии и символы невозможно, и нельзя осуществить динамическое изображение или анимацию. Иногда для обеспечения возможности ограниченной регенерации используется промежуточное состояние (режим рисования поверх изображения); см. ниже. В этом случае интенсивность электронного луча принимает значение меньше того порогового, которое вызывает запоминание, но достаточное для свечения люминофора. Поскольку в этом режиме изображение не сохраняется, для его видимости необходима постоянная перерисовка.

Дисплей на запоминающей трубке способен изображать фактически неограниченное количество векторов, а мерцание изображения в принципе невозможно. Его разрешение обычно составляет 1024×1024 адресуемых точек (10 бит) на экране 8×8 дюймов (ЭЛТ с диагональю 11 дюймов) или 4096×4096 (12 бит) либо на экране 14×14 дюймов (ЭЛТ с диагональю 19 дюймов), либо на экране 18×18 дюймов (ЭЛТ с диагональю 25 дюймов). По вертикали обычно видно только 78% адресуемой области.

Дисплей на запоминающей трубке — это векторный дисплей, или дисплей с произвольным сканированием. Иначе говоря, отрезок (вектор) может быть нарисован непосредственно из одной адресуемой точки в любую другую. Относительно легко, быстро и недорого можно получить твердую копию экрана. Дис-

плеи на запоминающей трубке в некоторой степени легче программировать, чем векторные или растровые дисплеи с регенерацией изображения. Дисплеи на запоминающих ЭЛТ можно объединять с микрокомпьютерами в сателлитные графические системы или графические терминалы. В последнем случае алфавитно-цифровая или графическая информация передается от главного компьютера в терминал через интерфейс. Обычно используется последовательный интерфейс, по которому за единицу времени передается только один бит информации, однако может применяться и параллельный интерфейс. Уровень интерактивности дисплея с запоминающей трубкой ниже, чем у векторного дисплея вследствие небольшой скорости интерфейса и неудовлетворительных характеристик стирания.

1-8 ГРАФИЧЕСКИЕ ДИСПЛЕИ С РЕГЕНЕРАЦИЕЙ ИЗОБРАЖЕНИЯ

В отличие от дисплея на запоминающей трубке в векторном (рисующем отрезки или векторы) дисплее с регенерацией изображения используется люминофор с очень коротким временем послесвещения. Такие дисплеи часто называют дисплеями с произвольным сканированием. Из-за малого времени послесвещения изображение на ЭЛТ за секунду должно многократно перерисовываться или регенерироваться. Минимальная частота регенерации должна составлять по крайней мере 30 кадров в секунду при рекомендуемой частоте от 40 до 50 кадр/с. Частота меньше, чем 30 кадр/с, приводит к мерцанию изображения, как это бывает при замедленном прокручивании кинофильма. На такое изображение неприятно смотреть, и его трудно использовать.

Для векторного дисплея с регенерацией кроме ЭЛТ требуются еще два элемента: дисплейный буфер и дисплейный контроллер. Дисплейный буфер — это участок непрерывной памяти, содержащей всю информацию, необходимую для вывода изображения на экран. Функция дисплейного контроллера состоит в циклической обработке полученной информации со скоростью регенерации. Сложность рисунка (количество изображаемых векторов) ограничивается двумя факторами — размером дисплейного буфера и скоростью дисплейного контроллера. Еще одним ограничением является скорость обработки геометрической информации, например скорость выполнения таких операций, как преобразование, отсечение и генерация текстов.

На рис. 1-18 представлены блок-схемы двух высокопроизводительных векторных дисплеев. В обоих случаях предполагается, что такие геометрические преобразования, как поворот, перенос, масштабирование, перспективное проецирование и отсечение, реализованы аппаратно в геометрическом процессоре. В первом случае (рис. 1-18а) геометрический процессор работает медленнее, чем это необходимо при регенерации применяемых на практике изображений (4000–5000 векторов). Таким образом, геометрические данные (экранные координаты векторов), посыпаемые центральным процессором (ЦП) графическому дисплею, обрабатываются до сохранения в дисплейном буфере. Следовательно, в нем содержатся только те инструкции, которые необходимы генератору векторов и символов. Обычно векторы задаются экранными координатами, а контроллер читает информацию из буфера и передает ее генератору векторов и символов. При до-

стижении конца дисплейного буфера контроллер возвращается на его начало, и цикл повторяется снова.

При использовании первой схемы реализуется концепция двойной буферизации и независимого изменения изображения и его регенерации. Так как в этой конфигурации геометрический процессор не успевает сгенерировать сложное новое или измененное изображение за один цикл регенерации, то дисплейный буфер делится на две части. В то время, когда измененное изображение обрабатывается и записывается в одну половину буфера, дисплейный контроллер регенерирует ЭЛТ из другой половины. При завершении изменения изображения буфера меняются ролями и процесс повторяется. Таким образом, новое или измененное изображение может генерироваться в каждый второй, третий, четвертый и т. д. циклы регенерации. Использование двойной буферизации предотвращает одновременный вывод части старого и части нового измененного изображения в течение одного и более циклов регенерации.

Во второй схеме (рис. 1-18б) геометрический процессор работает быстрее, чем необходимо для регенерации достаточно сложных изображений. В этом случае исходная геометрическая база данных, переданная из ЦП, сохраняется непосредственно в дисплейном буфере, а векторы обычно задаются в мировых координатах в виде чисел с плавающей точкой. Дисплейный контроллер за один цикл регене-

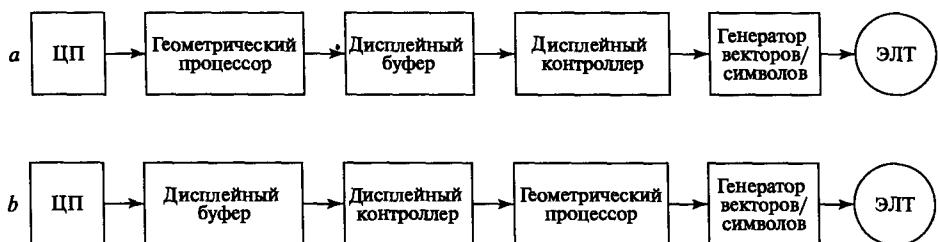


Рис. 1-18 Концептуальные блок-схемы векторных дисплеев с регенерацией изображения.

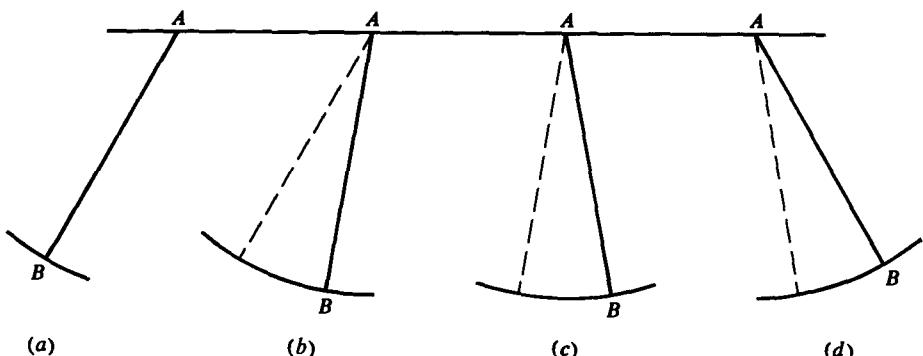


Рис. 1-19 Динамическое движение.

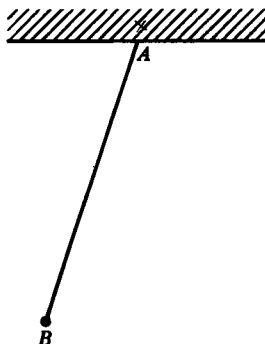


Рис. 1-20 Сегментация дисплейного буфера.

рации считывает информацию из буфера, пропускает ее через геометрический процессор и передает генератору векторов. При таком способе обработки геометрические преобразования выполняются «на лету» за один цикл регенерации.

При использовании любой конфигурации в дисплейном буфере содержатся команды рисования каждого вектора, символа и подкартинки. Следовательно, любой конкретный элемент изображения может быть изменен независимо от другого. Эта особенность вместе с малым временем послесвечения люминофора ЭЛТ позволяет выводить динамические изображения, что и показано на рис. 1.19, на котором представлена последовательность из четырех циклов. Отрезок, изображенный сплошной линией, — это отрезок, рисуемый для текущего цикла, штриховой линией изображен отрезок из предыдущего цикла. Между двумя циклами регенерации происходит изменение координат конца отрезка — точки *B*. Создается впечатление, что отрезок вращается вокруг точки *A*.

В большинстве случаев лишь часть изображения является динамически изменяемой. В действительности основная часть картинки остается статичной. Такое разделение наводит на мысль о необходимости сегментации дисплейного буфера, которая показана на рис. 1-20. Неподвижны, т. е. не изменяются от одного цикла регенерации к другому: горизонтальный отрезок, заштрихованная часть и буква *A*, используемые для показа опоры отрезка *AB*. В то же время для получения динамического изображения меняются положения конца отрезка *AB* и буквы *B*. Эти изолированные части базы данных помещены в отдельные сегменты дисплейного буфера. Для конфигурации, показанной на рис. 1-18a, геометрический процессор может пропускать статический сегмент в дисплейном буфере, так как он не изменяется, и это существенно сокращает работу процессора при изменении рисунка. В данном случае можно модифицировать изображение только в динамическом сегменте. Еще одним достоинством подобного метода является сокращение общего количества данных, передаваемых из ЦП в геометрический процессор при каждом изменении изображения.

Для конфигурации, представленной на рис. 1-18b, возможны разные типы сегментации. Напомним, что для данного случая база данных изображения сохраняется в дисплейном буфере в виде мировых координат, а обработка изображения

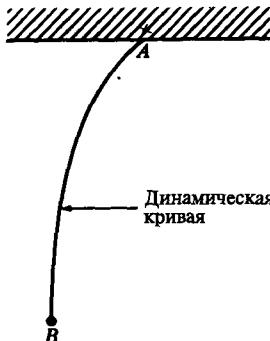


Рис. 1-21 Интеллектуальная сегментация дисплейного буфера.

происходит «на лету», за каждый цикл регенерации. Для картинки, показанной на рис. 1-20, в дисплейном буфере создаются два сегмента — статический и динамический. В любом случае обработка совершается мгновенно. Информацию в динамическом сегменте можно изменить с помощью функций, предоставляемых геометрическим процессором. Таким образом, модификация изображения может происходить локально в графическом устройстве, а для этого не требуется связь с ЦП. В конкретном случае на рис. 1-20 единственной функцией, необходимой для локального динамического изменения, является поворот вокруг точки А.

Для динамического изменения применительно к рис. 1-21 требуется связь с ЦП, т. е. необходимо некоторое интеллектуальное изменение изображения. Снова создаются два сегмента: статический, содержащий опорную линию, заштрихованную часть и букву А, и динамический, состоящий из кривой АВ и буквы В. Предположим, что форма кривой АВ должна изменяться от одного цикла регенерации к другому в зависимости от физических факторов. Это изменение выполняет прикладная программа в ЦП. Для модификации сегмента с динамически изменяющимся изображением должны быть переданы и сохранены в дисплейном буфере новые данные, например форма кривой.

Применение сегментации изображения не ограничивается только движением или мультилипикацией, хотя само понятие было введено с помощью примеров из этой области. Сегментировано может быть любое изображение, что особенно полезно для интерактивных графических программ. Понятие сегментации аналогично модульному программированию. Выбор модульных сегментов изображения, их размер и сложность зависят от конкретной прикладной области. Сложность элементов изображения изменяется от отдельных точек до полных описаний объектов. Типичный векторный дисплей с регенерацией изображения показан на рис. 1-22.

Чтобы показать, насколько важна скорость передачи или ширина полосы канала связи между ЦП и графическим устройством, рассмотрим затраты на интеллектуальное изменение кривой, описываемой 250 отрезками или точками, заданными тремя координатами. Пусть для представления чисел с плавающей точкой используются шесть значащих цифр (литер), и пусть каждая цифра задается одним байтом (8 бит). Тогда при скорости регенерации 30 кадр/с и при усло-

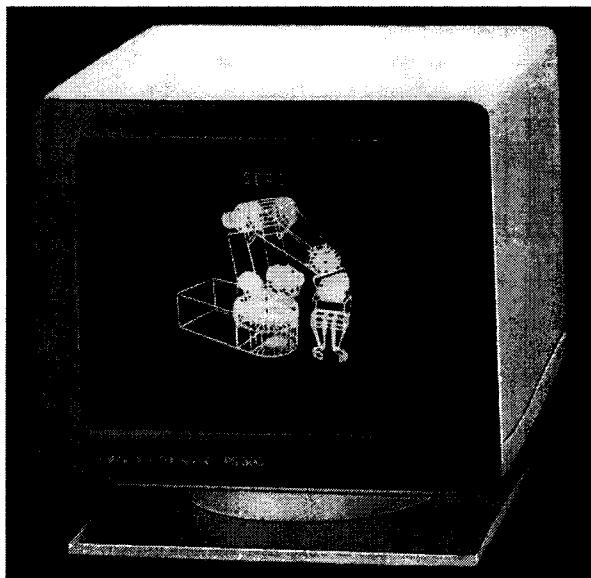


Рис. 1-22 Векторный дисплей с регенерацией изображения. (С разрешения фирмы Evans & Sutherland Computer Corp.)

вии, что модификация производится в каждом цикле, требуемая ширина полосы связи составит

$$30[(\text{кол-во точек})(\text{кол-во коорд./точку})(\text{кол-во знач. цифр/точку})(\text{кол-во бит/литера})] \text{ или} \\ 30(250)(3)(6)(8) = 1\,080\,000 \text{бит/с}$$

Таким образом, необходимая скорость передачи данных может легко превысить 1 Мбит/с. Для сложных трехмерных скульптурных поверхностей требуемая ширина полосы может возрасти в 10 раз, т. е. достигнуть 10 Мбит/с. В большинстве случаев для поддержки динамической графики в реальном масштабе времени такие скорости требуют применения между ЦП и графическим устройством параллельного интерфейса или интерфейса с прямым доступом в память.

1-9 РАСТРОВЫЕ ГРАФИЧЕСКИЕ ДИСПЛЕИ С РЕГЕНЕРАЦИЕЙ ИЗОБРАЖЕНИЯ

Как дисплеи на запоминающих ЭЛТ, так и дисплеи с произвольным сканированием являются устройствами рисования отрезков, т. е. отрезок прямой может быть нарисован непосредственно из любой адресуемой точки в любую другую. Графическое устройство на растровой ЭЛТ работает иначе. Растворное устройство можно представить в виде матрицы дискретных ячеек (точек), каждая из которых может быть подсвечена, т. е. оно является точечно-рисующим устройством. Невозможно, за исключением специальных случаев, непосредственно на-

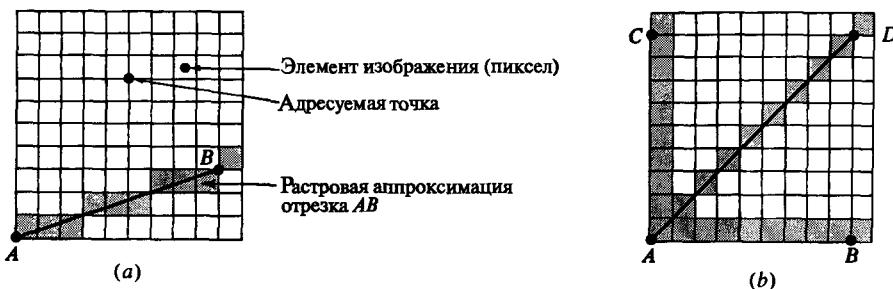


Рис. 1-23 Растворенная развертка отрезка.

рисовать отрезок прямой из одной адресуемой точки или пикселя в матрице в другую адресуемую точку или пиксель. Отрезок можно лишь аппроксимировать последовательностями точек (пикселов), близко лежащими к реальной траектории отрезка, что и проиллюстрировано на рис. 1-23а. Как видно из рис. 1-23б, отрезок прямой из точек получится только в случае горизонтальных, вертикальных или расположенных под углом 45° отрезков. Все другие отрезки будут выглядеть как последовательности ступенек. Это явление называется лестничным эффектом, или «зазубренностью». Методы, позволяющие устранить лестничный эффект обсуждаются в [1-1].

Чаще всего для графических устройств с растровой ЭЛТ используется буфер кадра. Буфер кадра представляет собой большой непрерывный участок памяти компьютера. Для каждой точки или пикселя в растре отводится как минимум один бит памяти. Эта память называется битовой плоскостью. Для квадратного раstra размером 512 × 512 требуется 2^{18} ($2^9 = 512$; $2^{18} = 512 \times 512$), или 262 144, бита памяти в одной битовой плоскости. Изображение в буфере кадра строится

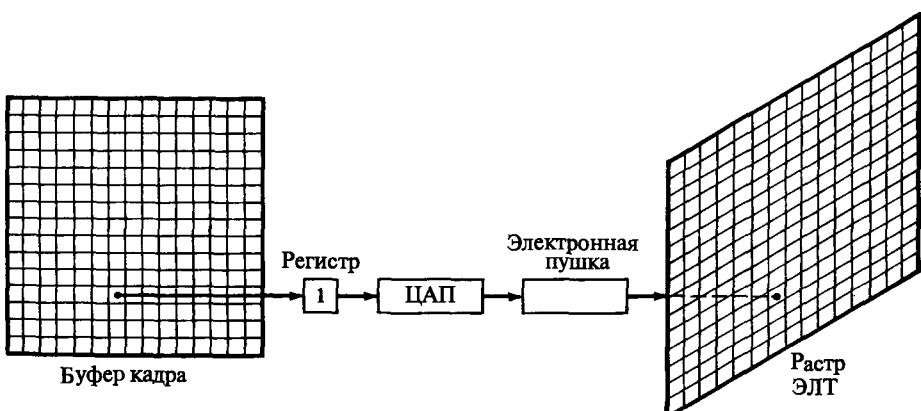


Рис. 1-24 Черно-белый буфер кадра (с одной битовой плоскостью) для растрового графического устройства.

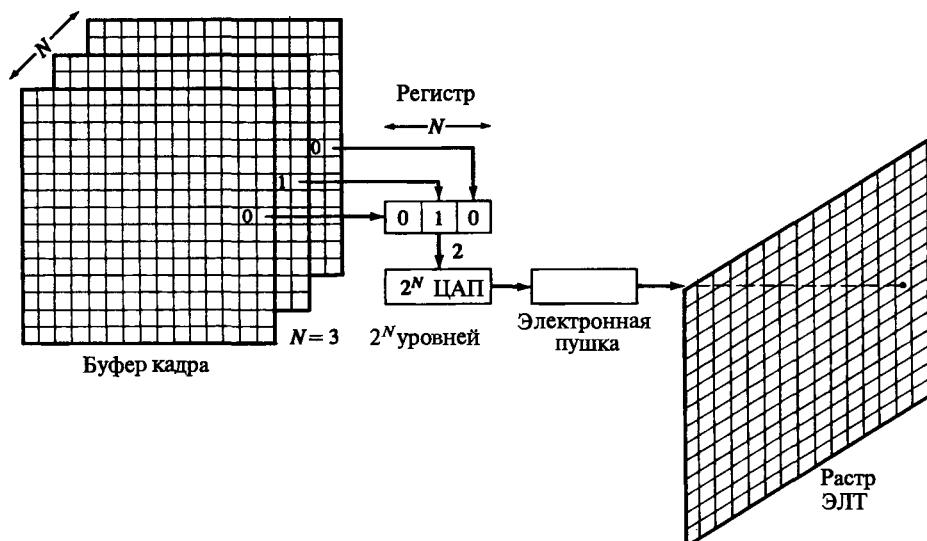


Рис. 1-25 Полутоновый черно-белый буфер кадра с N битовыми плоскостями.

побитно. Из-за того что бит памяти имеет только два состояния (двоичные 0 или 1), при одной битовой плоскости можно получить лишь черно-белое изображение. Битовая плоскость является цифровым устройством, тогда как растровая ЭЛТ – аналоговое устройство, для работы которого требуется электрическое напряжение. Поэтому при считывании информации из буфера кадра и ее выводе на графическое устройство с растровой ЭЛТ должно происходить преобразование из цифрового представления в аналоговый сигнал. Такое преобразование выполняет цифро-аналоговый преобразователь (ЦАП). Каждый пиксель буфера кадра нужно считать и преобразовать, прежде чем он будет отображен на растровой ЭЛТ. На рис. 1-24 приведена схема графического устройства с черно-белой растровой ЭЛТ, которое построено на принципе буферизации кадра в одной битовой плоскости.

Цвета или полутона серого цвета могут быть введены в буфер кадра путем использования дополнительных битовых плоскостей. На рис. 1-25 показаны схемы буфера кадра с N битовыми плоскостями для градации серого цвета. Интенсивность каждого пикселя на ЭЛТ управляется содержимым соответствующих пикселов в каждой из N битовых плоскостей. В соответствующую позицию регистра загружается бинарная величина (0 или 1) из каждой плоскости. Получившееся в результате двоичное число интерпретируется как уровень интенсивности между 0 и $2^N - 1$. С помощью ЦАП это число преобразуется в напряжение между 0 (темный экран) и $2^N - 1$ (максимальная интенсивность свечения). Таким образом можно получить 2^N уровней интенсивности. Рис. 1-25 иллюстрирует систему с тремя битовыми плоскостями для 8 (2^3) уровней интенсивности. Для каждой битовой плоскости требуется полный объем памяти при заданном разрешении растра: например, буфер кадра с тремя битовыми плоскостями для растра 512×512 занимает 7 864 32 (3 × 512 × 512) битов памяти.

Таблица 1-1 Комбинации цветов для простого 3-битового буфера кадра

	Красный	Зеленый	Синий
Черный	0	0	0
Красный	1	0	0
Зеленый	0	1	0
Синий	0	0	1
Желтый	1	1	0
Голубой	0	1	1
Пурпурный	1	0	1
Белый	1	1	1

Число доступных уровней интенсивности можно увеличить, незначительно расширив требуемую для этого память и воспользовавшись таблицей цветов, как это схематично показано на рис. 1-26. После считывания из буфера кадра битовых плоскостей получившееся число используется как индекс в таблице цветов, в которой должно содержаться 2^N элементов. Каждый ее элемент может иметь W бит, причем W может быть больше N . В этом случае можно получить 2^W значений интенсивности, но одновременно могут быть доступны лишь 2^N из них. Для получения дополнительных значений интенсивности необходимо изменить или перезагрузить таблицу цветов.

Поскольку существует три основных цвета, можно реализовать простой цветовой буфер кадра с тремя битовыми плоскостями, по одной на каждый из цветов. Отдельная битовая плоскость управляет индивидуальной электронной пушкой для каждого из трех основных цветов, используемых в устройстве. Комбинации их трех основных цветов дают восемь цветов на экране ЭЛТ, которые вместе с соответствующими им двоичными кодами приведены в табл. 1-1. Схема простого цветового растрового буфера кадра показана на рис. 1-27.

Для каждой из трех цветовых пушек могут использоваться дополнительные битовые плоскости. На рис. 1-28 схематично показан цветовой буфер кадра с 8 битовыми плоскостями на каждый цвет, т. е. буфер кадра с 24 битовыми плоскостями. Каждая группа плоскостей управляет 8-разрядным ЦАП и может генерировать 256 (2^8) оттенков, или интенсивностей, красного, зеленого или синего цвета. Их можно комбинировать для получения 16 777 216 [$(2^8)^3 = 2^{24}$] возможных цветов. Это «полноцветный» буфер кадра.

Полноцветный буфер кадра можно еще увеличить путем использования групп битовых плоскостей в качестве индексов в таблицах цветов, как это показано на рис. 1-29. При N битах на цвет и W -разрядных элементах таблиц цветов одновременно могут быть показаны $(2^3)^N$ цветовых оттенков из палитры с $(2^3)^W$ возможных цветов. Например, при буфере кадра с 24 битовыми плоскостями ($N = 8$) и тремя 10-разрядными таблицами цветов ($W = 10$) могут быть получены 16 777 216 (2^{24}) цветовых оттенков из палитры 1 073 741 824 (2^{30}) цветов, т. е. около 17 млн. оттенков из палитры немногим менее чем 1 млрд. цветов. Несмотря на то что на рис. 1-29 схематически показаны три различных таблицы цветов, при небольшом количестве физических битовых плоскостей (не более 12) более предпочтительной является ситуация, когда изменение цветов осуществляется непрерывно в таблице с $(2^3)^N$ элементами.

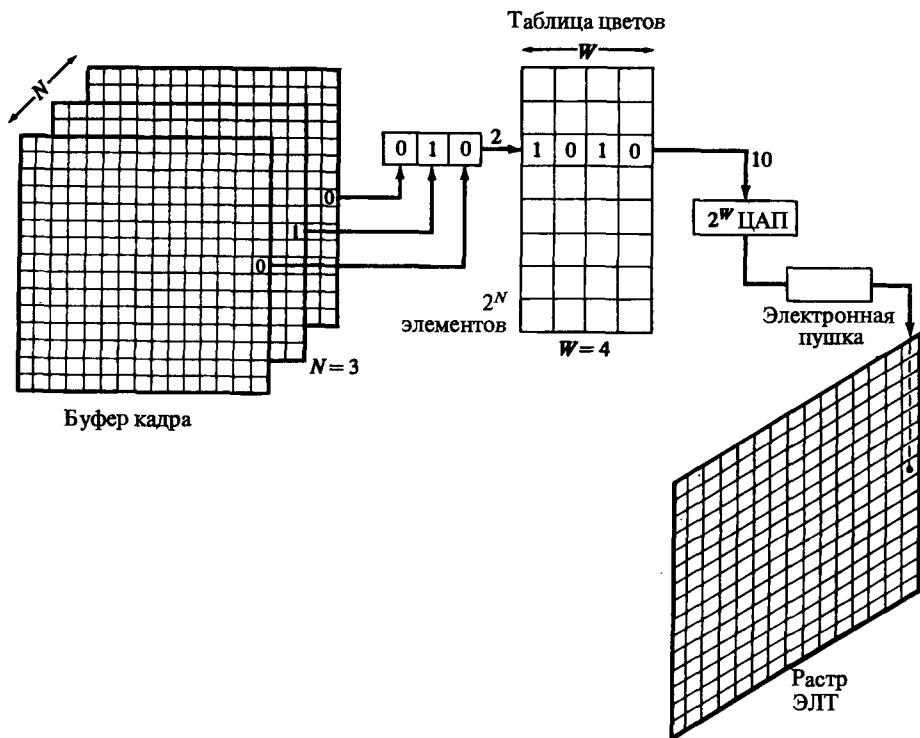


Рис. 1-26 Полутоновый черно-белый буфер кадра с N битовыми плоскостями и W -разрядной таблицей цветов.

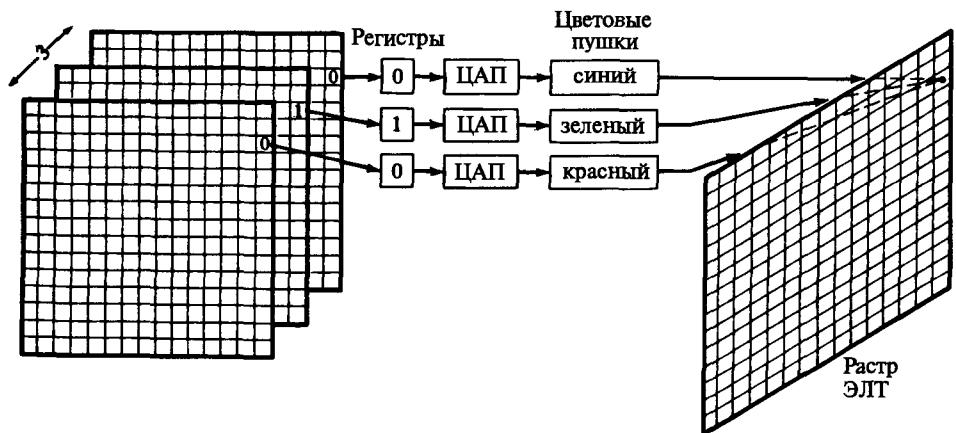


Рис. 1-27 Простой цветовой буфер кадра.

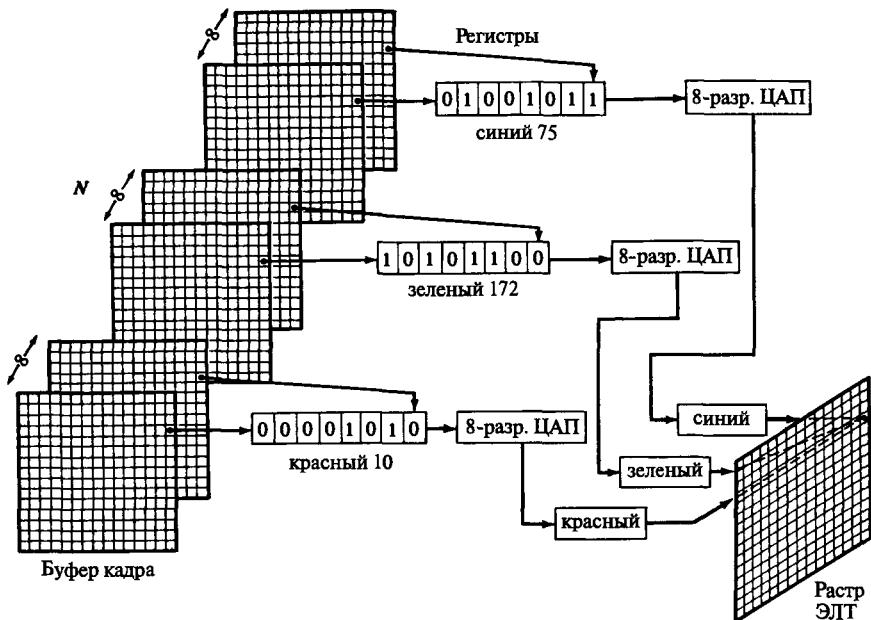


Рис. 1-28 Цветовой буфер кадра с 24 битовыми плоскостями.

Из-за большого количества пикселов в растровых графических устройствах трудно достигнуть производительности, необходимой для работы в реальном масштабе времени, а также для работы с приемлемой частотой регенерации, или смены кадра. Например, если среднее время доступа к каждому отдельному пикселу равно 200 нс ($200 \times 10^{-9} \text{ с}$), то для организации доступа ко всем пикセルам кадра размером 512×512 потребуется 0.0524 с , что эквивалентно частоте регенерации 19 кадров (картинок) в секунду, а это значительно меньше минимально необходимой частоты, равной 30 кадр/с. В буфере кадра размером 1024×1024 содержится немногим более 1 млн. бит (1 Мбит), и при среднем времени доступа 200 нс требуется 0.21 с для обращения ко всем пикセルам, что составляет 5 кадр/с. Буфер кадра размером 4096×4096 содержит 16.78 млн. бит на каждую битовую плоскость! Доступ к ним займет 0.3 с. Для достижения скорости регенерации 30 кадров в секунду при таком растре требуется средняя эффективная скорость доступа 2 нс/пиксель. Напомним, что за этот небольшой период времени свет успевает пройти два фута.

Работа в реальном времени с растровыми графическими устройствами осуществляется путем одновременного доступа к группам по 16, 32, 64 и более пикселов. В случае цветового буфера кадра каждый пиксель может содержать до 32 бит, при этом все битовые плоскости для каждого пикселя доступны одновременно. При среднем времени доступа для каждой группы пикселов 1600 нс возможна работа в реальном времени для буферов кадров размером 512×512 и 1024×1024 .

Несмотря на то что для растровых устройств труднее достигнуть производительности, необходимой для работы в реальном масштабе времени с приемлемыми скоростями регенерации, чем на векторных дисплеях с регенерацией изображе-

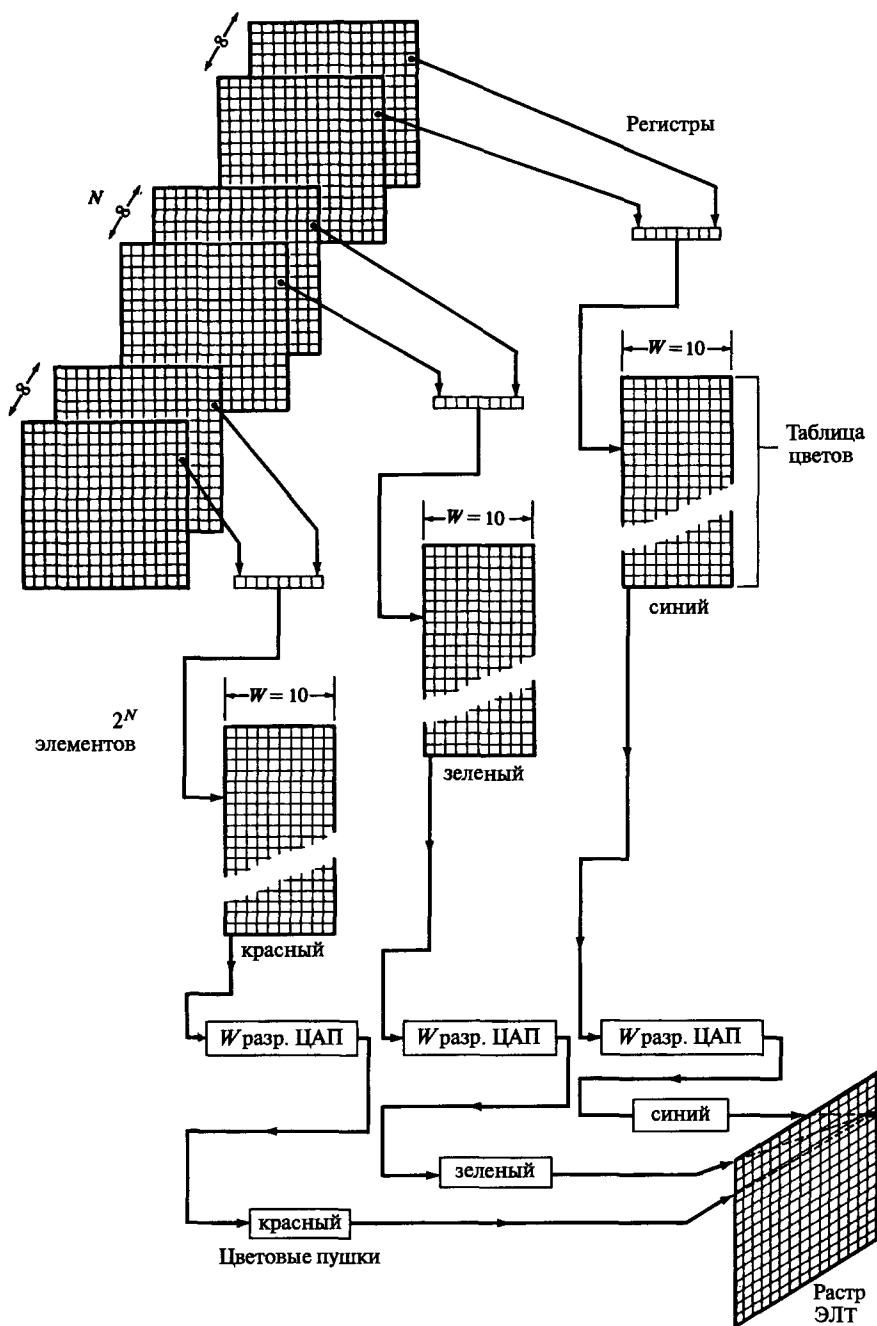


Рис. 1-29 Цветовой буфер кадра с 24 битовыми плоскостями и 10-разрядной таблицей цветов.

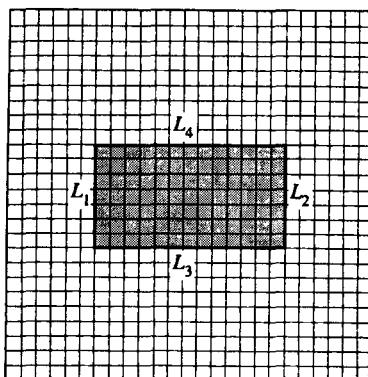


Рис. 1-30 Изображение сплошной фигуры на растровом графическом устройстве.

ния, на них легче изображать сплошные фигуры с плавными переходами цветов. Как показано на рис. 1-30, растровое представление сплошной «полигональной» фигуры концептуально достаточно просто. Здесь представление сплошной фигуры, ограниченной отрезками L1, L2, L3, L4, достигается установкой всех пикселов внутри ограничивающего многоугольника в соответствующий цвет в буфере кадра. Алгоритм «растровой развертки» сплошной области обсуждается в работе [1-1].

1-10 УСТРОЙСТВО ЭЛЕКТРОННО-ЛУЧЕВОЙ ТРУБКИ

Описанный выше буфер кадра сам по себе не является устройством вывода, он просто применяется для хранения рисунка. Наиболее часто в качестве устройства отображения, используемого с буфером кадра, выступает видеомонитор. Для того чтобы понять принципы работы растровых дисплеев и в некоторой степени векторных устройств с регенерацией, нужно иметь определенное представление об ЭЛТ и видеотехнике.

На рис. 1-31 схематично показана ЭЛТ, используемая в видеомониторах. Катод (отрицательно заряженный) нагревается до тех пор, пока возбужденные

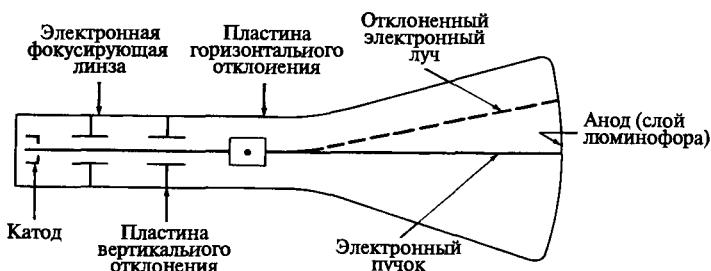


Рис. 1-31 Электронно-лучевая трубка.

электроны не создадут расширяющегося облака (электроны отталкиваются друг от друга, так как имеют одинаковый заряд). Эти электроны притягиваются к сильно заряженному положительному аноду. На внутреннюю сторону расширенного конца ЭЛТ нанесен люминофор. Если бы электронам ничто не препятствовало, то в результате их воздействия на люминофор весь экран ЭЛТ засветился бы ярким светом. Однако облако электронов с помощью электронных линз фокусируется в узкий, строго параллельный пучок, и теперь электронный луч дает одно яркое пятно в центре экрана. Луч отклоняется, или позиционируется, влево или вправо от центра и(или) выше или ниже центра с помощью усилителей горизонтального и вертикального отклонения.

Именно в данный момент проявляется отличие векторных дисплеев — как с запоминанием, так и с регенерацией — от растровых. В векторном дисплее электронный луч может быть отклонен непосредственно из любой произвольной позиции в любую другую позицию на экране ЭЛТ (аноде). Поскольку люминофор нанесен на экран сплошным слоем, то в результате получается почти идеальная прямая. В отличие от этого в растровом дисплее луч может отклоняться только в строго определенные позиции на экране, образующие своеобразную мозаику. Эта мозаика и составляет видеоизображение.

1-11 УСТРОЙСТВО ЦВЕТНОЙ РАСТРОВОЙ ЭЛТ

Цветная растровая ЭЛТ похожа на обычную черно-белую ЭЛТ, принцип действия которой был описан в предыдущем разделе. Данная трубка содержит три электронные пушки, по одной на каждый основной цвет: красный, зеленый и синий. Электронные пушки часто объединены в треугольный блок, соответствующий подобному треугольному блоку точек красного, зеленого и синего люминофоров на экране ЭЛТ (рис. 1-32). Для того чтобы электронные пушки возбуждали только соответствующие им точки люминофора (например, красная пушка возбуждала только точку красного люминофора), между пушками и поверхностью экрана помещена перфорированная металлическая решетка — так называемая теневая маска стандартной ЭЛТ. Отверстия в маске образуют такие же треуголь-

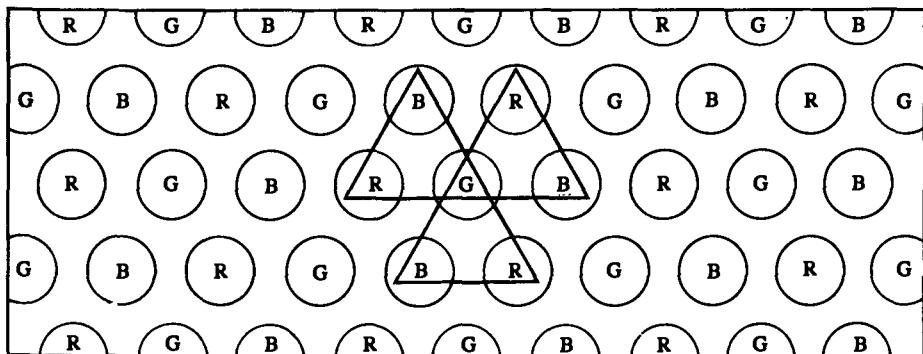


Рис. 1-32 Точечный люминофорный растр для ЭЛТ с теневой маской.

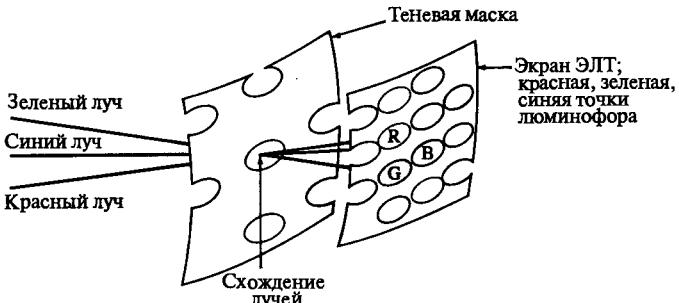


Рис. 1-33 Расположение электронной пушки и теневой маски цветной ЭЛТ.

ные блоки, как и точки люминофора. Расстояния между отверстиями называются шагом. Цветовые пушки расположены таким образом, что их лучи сходятся и пересекаются в плоскости теневой маски (рис. 1-33). После прохождения через отверстие красный луч, например, защищен или маскирован от пересечения с зеленой или синей точками люминофора. Он может пересечь лишь красную точку. Изменяя интенсивность электронного луча для каждого основного цвета, можно получить различные оттенки. Комбинация этих оттенков дает большое количество цветов для каждого пикселя. Обычно в дисплее с высоким разрешением на каждый пиксель приходится от двух до трех цветовых триад.

1-12 СИСТЕМЫ С ТЕЛЕВИЗИОННЫМ РАСТРОМ

Процесс преобразования хранящегося в буфере кадра растрового изображения в упорядоченный набор точек на телевизоре называется растровой разверткой. Сканируемый набор точек и частота воспроизведения основаны как на особенностях визуального восприятия, так и на принципах электроники. Системе визуального восприятия человека для рассмотрения элементов картинки требуется конечное время. Однако обеспечить впечатление непрерывности позволит только такой интервал, который настолько мал, что инерция зрительного восприятия перекроет мерцание. На мерцание влияет ряд факторов, включая яркость изображения и свойства конкретного люминофора, используемого для покрытия экрана ЭЛТ. Опыт показывает, что для практических целей минимальной скоростью вывода или изменения изображения является 25 кадр/с при условии, что минимальная скорость регенерации или воспроизведения в два раза больше, т. е. 50 кадр/с. Аналогичная ситуация имеет место и при демонстрации кинофильма. При этом показывается 24 кадр/с, но каждый кадр показывается дважды, и в результате получается эффективная скорость воспроизведения 48 кадр/с. Таким образом, для фильма скорость изменения равна 24 кадр/с, а скорость регенерации — 48 кадр/с. В телевидении тот же эффект достигается с помощью метода чересстрочной развертки.

Телевидение использует метод растрового сканирования. В американской стандартной видеосистеме используются в совокупности 525 горизонтальных строк с кадровым, или видовым, отношением 4:3, т. е. высота видовой области равна трем

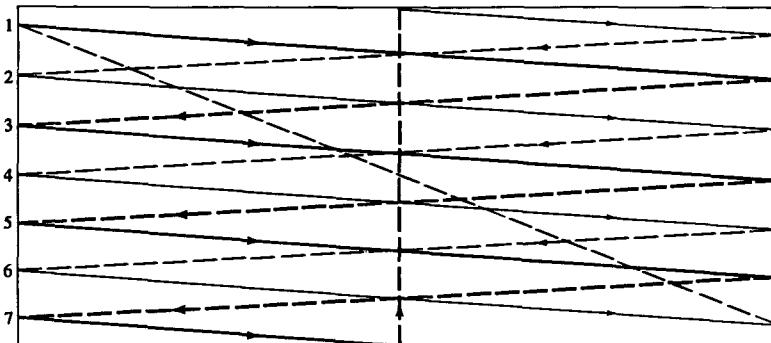


Рис. 1-34 Схема чересстрочной развертки для раstra из семи строк. Нечетное поле начинается со строки 1. Штриховой линией обозначен горизонтальный обратный ход луча. Вертикальный обратный ход луча для нечетного поля начинается внизу в центре, а для четного поля — внизу справа.

четвертям ее ширины. Скорость воспроизведения, или смены кадра, составляет 30 кадр/с. Однако каждый кадр делится на два поля, каждое из которых содержит по половине картинки. Эти два поля чередуются или перемежаются. Они попеременно показываются через каждые $1/60$ с. Одно поле содержит все строки с нечетными номерами (1, 3, 5, ...), а другое — с четными (2, 4, 6, ...). Сканирование начинается в верхнем левом углу экрана с нечетного поля. Каждая строка в поле сканируется или представляется слева направо. В то время как электронный луч движется поперек экрана слева направо, он также перемещается вертикально вниз, но с намного меньшей скоростью. Таким образом, «горизонтальная» сканирующая строка в действительности немножко наклонена. При достижении правого края экрана луч становится невидимым и быстро возвращается к левому краю. Такой горизонтальный возврат луча обычно занимает около 17% времени, отведенного для одной сканирующей строки. Затем этот процесс повторяется со следующей нечетной строкой. Так как половина от 525 равна 262.5 строки, то при завершении сканирования поля нечетных строк луч окажется в центре нижней строки экрана (рис. 1-34 и рис. 1-35). Луч после этого быстро переводится в центр верхней стороны экрана. Так осуществляется вертикальный перевод луча для нечетного поля. Время, затрачиваемое на него, эквивалентно времени, затрачиваемому на сканирование 21 строки. Затем показывается поле четных строк, после чего луч оказывается в нижнем правом углу. Перевод луча для поля четных строк возвращает его в верхний левый угол, и весь процесс повторяется снова. Таким образом, показываются два поля для каждого кадра, т. е. 60 полей в секунду. Данный метод существенно уменьшает мерцание, так как глаз воспринимает частоту воспроизведения полей.

Хотя в принятой в США стандартной видеосистеме предусмотрено 525 строк, на самом деле видимы только 483 строки, так как время, затрачиваемое на сканирование 21 строки, уходит на вертикальный перевод луча¹. В течение этого

¹ Во многих графических устройствах с растровым сканированием это время используется для обработки другой информации.



Рис. 1-35 Схема 525-строчного стандартного кадра.

времени электронный луч невидим или отключен. Время, отпущенное на каждую сканирующую строку, легко подсчитывается для частоты воспроизведения 30 кадр/с:

$$1/30 \text{ с/кадр} \times 1/525 \text{ кадр/строка} = 63,5 \text{ мкс/строка.}$$

Так как примерно 10,5 мкс расходуется на горизонтальный перевод луча, то вывод видимой части каждой строки должен быть завершен за 53 мкс. В строке при нормальном соотношении сторон видеообласти, равном 4:3, находятся 644 пикселя. Таким образом, время, отпущенное на считывание и вывод пикселя, равно

$$53 \text{ мкс/строка} \times 1/644 \text{ строка/пиксель} = 82 \text{ нс.}$$

Во многих растровых дисплеях, построенных на основе буфера кадра, применяется разрешение 512 пикселов на строку. Считывание и вывод пикселов при таком разрешении происходит примерно за 104 нс. Аналогичные результаты получаются для 625-строчной видеосистемы с частотой воспроизведения 25 кадр/с, используемой в Великобритании и большинстве стран Европы.

Метод чересстрочной развертки не является абсолютно необходимым при выводе изображения, однако изображение без чередования строк будет несовместимым со стандартным телевизионным приемником. При отсутствии чересстрочной развертки придется для устранения мерцания увеличить частоту воспроизведения до 60 кадр/с, а это, безусловно, в 2 раза сокращает время для обработки пикселя. Более высокое разрешение по числу строк и количеству пикселов в строке также уменьшает это время; например, при разрешении 1024 × 1024 на считывание и вывод пикселя отводится в 4 раза меньше времени, чем при разрешении 512 × 512, — примерно 25 нс! В этом случае потребуется очень быстрая память для буфера кадра и такой же высокопроизводительный ЦАП.

1-13 ДИСПЛЕИ С ПЛОСКИМ ЭКРАНОМ

Несмотря на то что в настоящее время электронно-лучевые трубы (ЭЛТ) в качестве устройств графического вывода находятся вне конкуренции, им присущ ряд

недостатков: большие габариты, тяжелый вес, хрупкость и ограниченные размеры экрана (обычно 30 дюймов по диагонали). На фоне этих и других ограничений ЭЛТ очень выигрышно выглядят плазменные дисплеи, принадлежащие к типу растровых устройств с регенерацией изображения.

Плазменные дисплеи по способу формирования изображения делятся на два принципиально различных класса устройств: активные (светоэмиссионные) и пассивные (основанные на модуляции света). К первым устройствам относятся ЭЛТ, дисплеи на газоразрядных панелях, электролюминесцентные и вакуумные дисплеи. Жидкие кристаллы и светодиоды представляют класс пассивных устройств.

Среди устройств с активной генерацией света для решения задач интерактивной машинной графики наиболее подходят плазменные и электролюминесцентные дисплеи, которые способны генерировать большие изображения с высокой разрешающей способностью. За исключением некоторых специальных приложений, например в авиации, где дисплеи на светоэмиссионных диодах обладают некоторыми преимуществами, дисплеи на жидких кристаллах являются наиболее удачными устройствами из класса систем с пассивной регенерацией света.

Как показано на рис. 1-36, изображение на плоской ЭЛТ получается после прохождения луча параллельно экрану и отражения под прямым углом. Использование эффекта отражения позволяет значительно уменьшить протяженность колбы ЭЛТ и, соответственно, габариты дисплея. ЭЛТ в форме плоской панели обладают теми же преимуществами, что и обычные ЭЛТ, но при значительно меньших габаритах. Большая длина «шейки» ЭЛТ ограничивает области ее применения. Использование вакуумных флюоресцентных дисплеев также в значительной степени сдерживается их габаритами.

Дисплеи на газоразрядных панелях, электролюминесцентные дисплеи и дисплеи на жидких кристаллах имеют несколько общих характеристик. Все они состоят из матрицы отдельных пикселов растра, в каждом из которых помещено некоторое устройство или вещество, которые под воздействием напряжения или тока начинают испускать свет либо модулировать падающий свет. Требуемое для этого напряжение или ток подается каждому пикселу через отдельный электрический вентиль, например тонкопленочный транзистор, диод или резистор типа

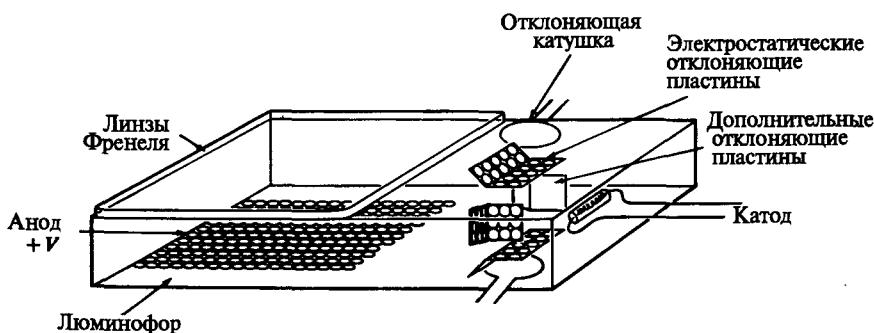


Рис. 1-36 Схема ЭЛТ.

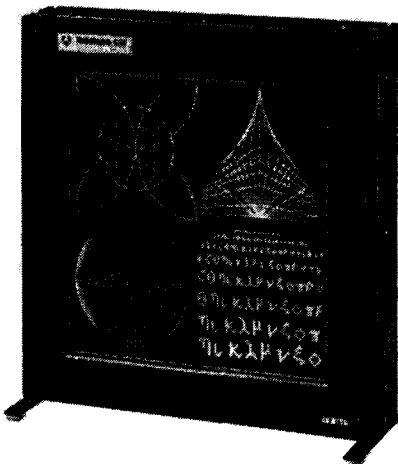


Рис. 1-37 Типичный плазменный дисплей. (С разрешения фирмы Thomson CSF Electron Tubes & Device.)

металл—изолятор—металл, размещенный под каждым пикселом. Дисплеи, основанные на данном принципе, называются матричными.

Другой способ, получивший широкое распространение, основан на использовании адресации пикселов по строкам и столбцам, что позволяет значительно уменьшить количество вентилей или драйверов генерации изображения. Данный способ предполагает наличие у материала, из которого изготовлен дисплей, некоторого порога переключения. Эмиссия или модуляция света в таком элементе возможна только при превышении такого порога переключения. На соответствующую строку матрицы пикселов подается напряжение или ток, недостаточные для активации отдельного пикселя, а на соответствующий столбец подается другое напряжение или ток. Каждое в отдельности воздействие по строке или по столбцу ниже порогового значения, а в сумме они превышают его. По этой причине пиксели, не лежащие на пересечении нужной строки и столбца, оказываются в невозбужденном состоянии и только один пиксель получает напряжение или ток выше порогового значения становится активным.

При использовании такой адресации весьма желательно наличие для пикселов бистабильной памяти, в которой однажды активизированный пиксель сохраняет свое состояние, пока не произойдет его явное переключение. Бистабильная память исключает необходимость в постоянной регенерации пикселов, и, как следствие этого, не требуется внешней памяти для регенерации. Кроме того, значительно упрощается устройство дисплейного контроллера.

Принцип работы плазменного или газоразрядного дисплея предельно прост. Дисплей состоит из матрицы ячеек (растра), окруженных стеклянной оболочкой. Каждая ячейка заполнена газом (обычно неоном или смесью аргона с неоном) под давлением (ниже атмосферного). При действии достаточно высокого напряжения происходит диссоциация газа, т. е. атомы лишаются своих электронов. Такой газ в состоянии диссоциации называется плазмой, отсюда и возникло название — плаз-

менные дисплеи. При рекомбинации электронов происходит освобождение энергии в форме фотонов и газ начинает светиться с характерным оранжево-красным оттенком. На рис. 1-37 показан типичный плазменный дисплей.

Активация в плазменных дисплеях может быть типа AC (переменный ток) или DC (постоянный ток) либо представлять комбинацию AC/DC. На рис. 1-38 *a*, *b* и *c* приведены схемы плазменных дисплеев с активацией типа AC, DC и гибридной типа AC/DC соответственно. Дисплеи с типом DC являются более простыми, чем AC, и состоят из полой диэлектрической панели, содержащей полости с газом, которые помещены между панелями с матрицей проводников. Электронное поле прилагается непосредственно к газу. Дисплеи с типом DC обладают достаточно продолжительным циклом регенерации.

В дисплеях с типом AC слой диэлектрика размещается между проводником и газом. Таким образом, единственный путь соединения между газом и проводником — это емкость (конденсатор), и поэтому для получения диссоциированного газа требуется использование напряжения. Плазменные дисплеи типа AC имеют бистабильную память, что исключает необходимость в длительной регенерации. Эффект бистабильной памяти достигается при использовании низкого AC постоянного напряжения. Имеющейся емкости оказывается достаточно для поддержания напряжения активации токопроводящих пикселов, но недостаточно для активации непроводящих.

В гибридных дисплеях с типом активации AC/DC (рис. 1-38*c*) используется DC-напряжение для «первичного» газа, а для облегчения активации применяется AC-напряжение. Принципиальное преимущество гибридных AC/DC-плазменных дисплеев — это снижение сложности управляющей схемы. На практике находят применение как дисплеи больших размеров (3×3 метра с разрешением 4096×4096), так и системы высокого разрешения (100 пикселов на дюйм). На основе использования данных принципов можно также создавать цветные и полутоновые монохромные дисплеи.

В электролюминесцентном дисплее фосфоресцирующий материал излучает свет под действием электрического поля от AC или DC. В качестве фосфоресцирующего материала обычно используется сульфид цинка с добавками марганца, поэтому эти дисплеи излучают желтый свет. Для адресации пикселов используется такой же способ, как и в плазменных дисплеях. При подаче напряжения, превышающего пороговое значение, возбуждаются электроны марганцевой составляющей люминофора. Когда возбужденный атом возвращается на более низкую энергетическую орбиту, он испускает фотон, вызывающий характерный желтый цвет. Можно также получить хороший спектр серого цвета, так как уровень люминесценции управляет напряжением и частотой приложенного электрического поля. При использовании других типов добавок получаются различные цвета свечения покрытия, следовательно, из совокупности нескольких фосфоресцирующих слоев может быть сформирован цветной дисплей.

Люминофор представляет собой либо макроскопический субстрат, напыляемый в виде толстого слоя, либо тонкую пленку частиц толщиной в одну молекулу. Дисплеи с типом возбуждения тонкопленочного покрытия AC и DC получили наибольшее распространение в системах машинной графики. Принципиальная схема электролюминесцентного дисплея приведена на рис. 1-39, а на рис. 1-40

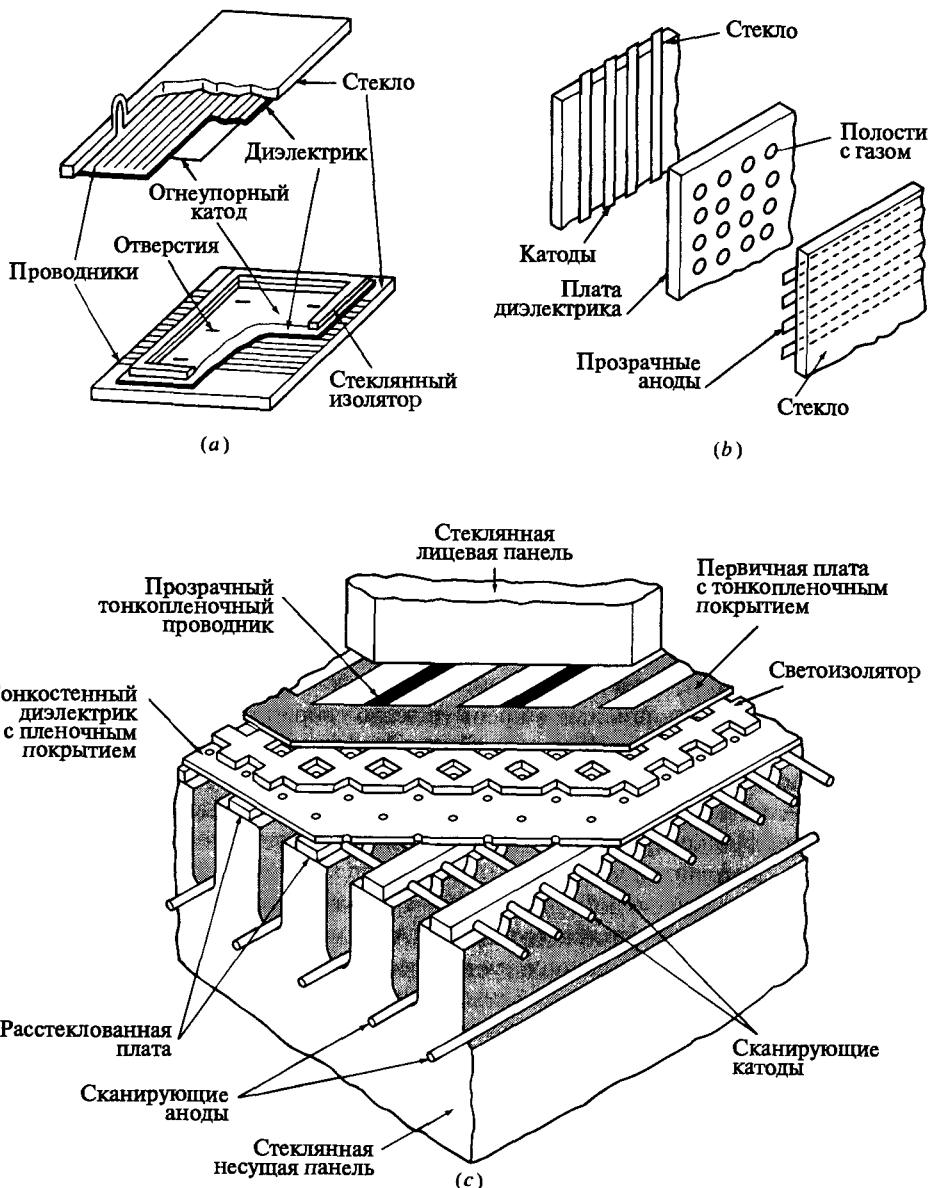


Рис. 1-38 Принципиальная схема плазменного газоразрядного дисплея: а) АС тип; б) DC тип; в) AC/DC тип.

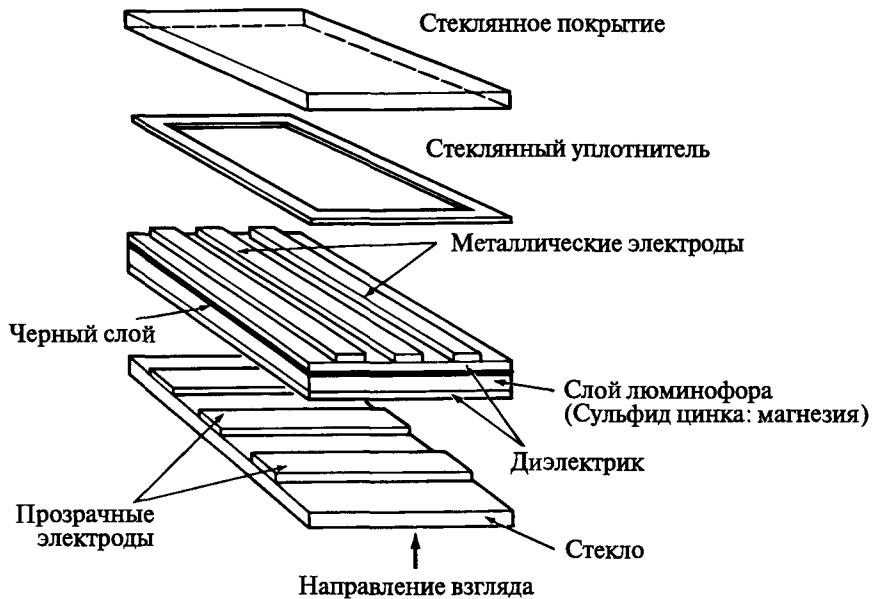


Рис. 1-39 Принципиальная схема электролюминесцентного дисплея типа АС.

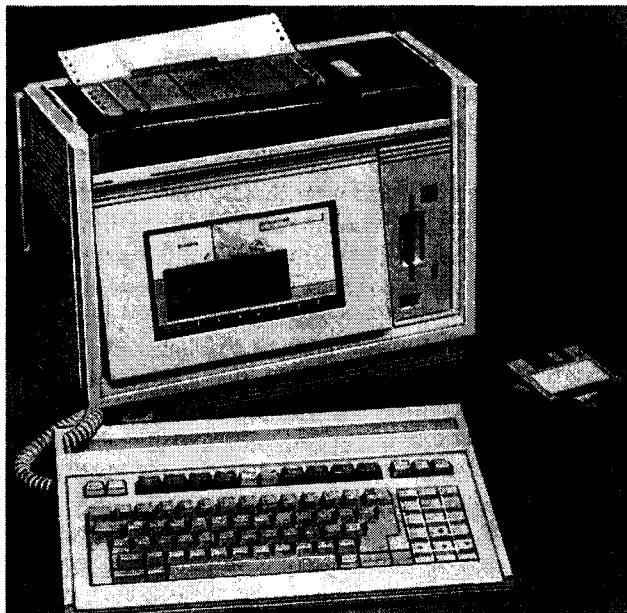


Рис. 1-40 Электролюминесцентный дисплей. (С разрешения фирмы Hewlett Packard.)

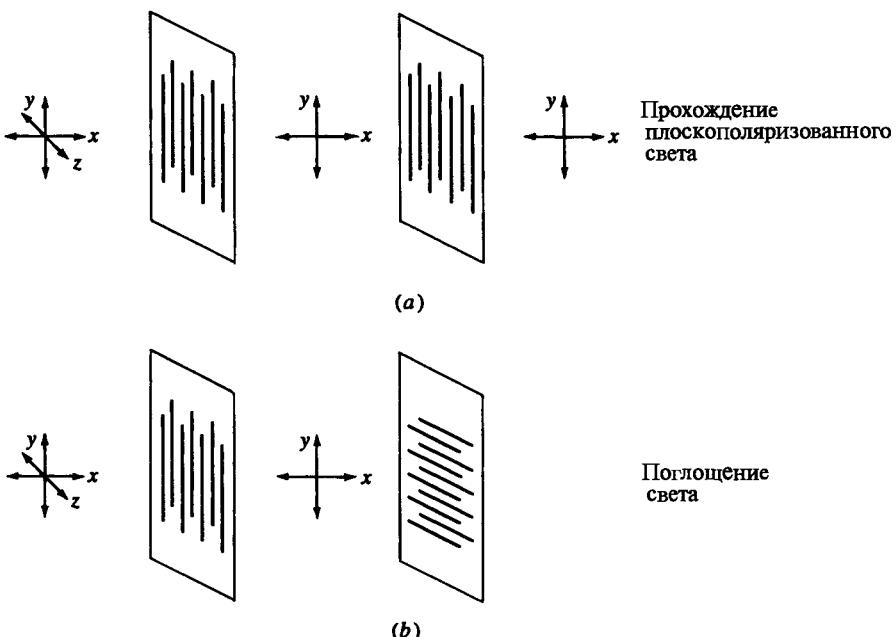


Рис. 1-41 Поляризация света: а) прохождение плоскополяризованного света; б) поглощение света.

изображен типичный дисплей такого типа. Из рисунка видно, что дисплей с разрешением 100 пикселов на дюйм имеет вполне компактные габариты.

Если плазменные и электролюминесцентные дисплеи представляют собой примеры устройств, использующих активную панель, то дисплеи на жидкких кристаллах являются образцом пассивной технологии, основанной либо на пропускании, либо на отражении падающего света. Для изменения характеристики падающего света используются свойства поляризации, которыми обладают некоторые органические вещества.

Рис. 1-41 наглядно иллюстрирует основные принципы поляризации света. Некогерентный свет (рис. 1-41а) проходит через первый (на рисунке левый) поляризатор и поляризуется по плоскости xy . Оси поляризации второго поляризатора также ориентированы в плоскости xy , поэтому свет беспрепятственно проходит дальше. На рис. 1-41б плоскость второго поляризатора повернута под прямым углом относительно первого, и поэтому свет, поляризованный в плоскости первого поляризатора, поглощается при прохождении через второй.

Органические вещества, способные сохранять стабильность в мезофазе при температуре между жидким и твердым состоянием, называются жидкими кристаллами. Различают три типа мезофазы: смектическую, нематическую и холестерическую.

В нематической фазе оси молекул жидкого кристалла располагаются параллельно, причем на их направление выравнивания влияют температура, поверхностное натяжение, давление и, что особенно важно для дисплеев, электрические

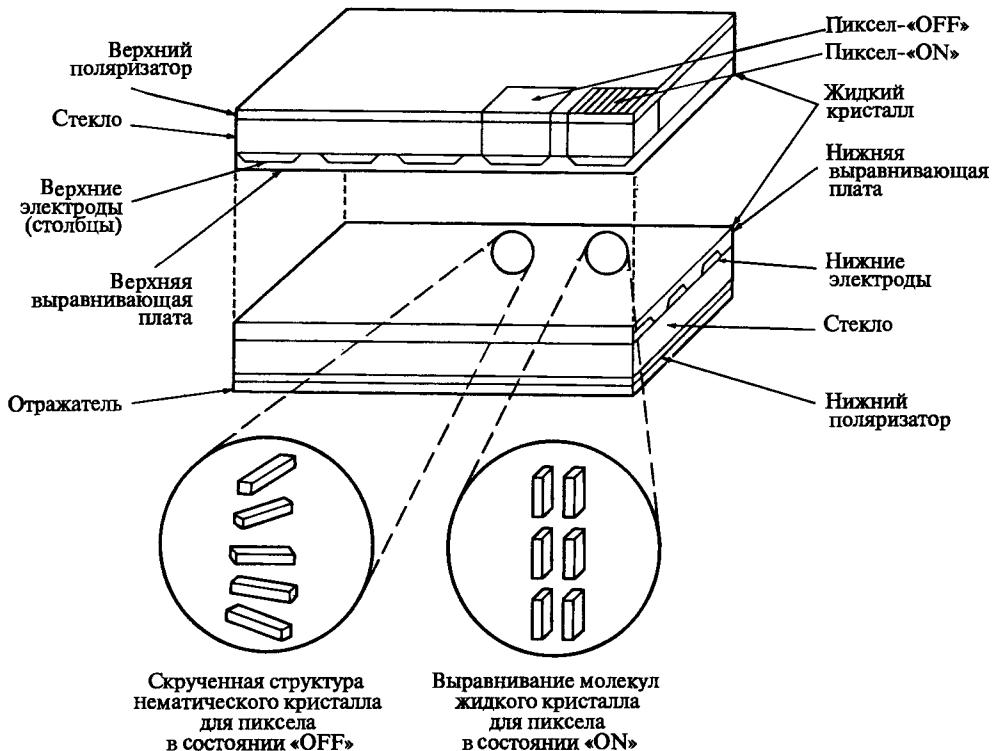


Рис. 1-42 Принципиальная схема дисплея на скрученных нематических жидкокристаллах.

и магнитные поля. Оптические характеристики жидкого кристалла также достаточно чувствительны к этим воздействиям.

Ключевым моментом одного из возможных способов построения дисплеев на жидкокристаллах является создание «бутерброда» из нематических кристаллов со взаимоперпендикулярными осями. На рис. 1-42 приведено изображение принципиальной схемы отражающего нематического жидкокристаллического дисплея. Нижняя и верхняя пластина бутерброда из жидкокристаллов имеют канавки, причем канавки на верхней плате направлены под прямым углом к канавкам нижней платы. Жидкие кристаллы прилегают к поверхности платы с соблюдением направлений канавок (рис. 1-42).

Дисплей имеет два плоских поляризатора, каждый из которых расположен на обеих сторонах бутерброда и повернут под прямым углом к другому. При затемненном пикселе или в состоянии поворота падающий на поверхность свет поляризуется первым поляризатором, затем проходит через жидкокристаллический бутерброд, где он поворачивается на прямой угол, далее идет сквозь второй поляризатор и затем отражается обратно, а пиксел становится подсвеченным.

Включение пикселя осуществляется путем подачи электрического поля на жидкий кристалл, что вызывает его переориентацию. Свет, падающий на па-

нель дисплея, поляризуется теперь первым поляризатором, проходит через жидкий кристалл без каких-либо изменений и оказывается полностью поглощенным вторым поляризатором. Пиксель становится темным. Жидкокристаллические дисплеи на повернутых нематических кристалах требуют для своей работы постоянной регенерации.

Бистабильные дисплеи на жидкких кристаллах могут быть построены на базе смектических кристаллов, две устойчивые ориентации молекул которых обладают различными оптическими свойствами, например уровнем отражения. Это позволяет достигать оптических контрастов за счет приложения импульсного напряжения, вызывающего изменение состояния молекул. Для дисплеев на кристаллах данного типа не требуется регенерации.

Использование цветных фильтров или фосфоров позволяет строить цветные дисплеи на основе повернутой нематической технологии или с использованием окрашивающих добавок. В дисплеях на кристаллах с добавками в основную жидкокристаллическую подложку вносятся двухцветные посторонние молекулы, спектральные характеристики которых используются для формирования различных цветов. Под действием электрического поля происходит изменение ориентации основных и привнесенных молекул, что дает возможность свету беспрепятственно проходить сквозь кристалл. На рис. 1-43 схематически показан типичный жидкий кристалл типа гость–хозяин. Отметим, что в этом случае требуется только один поляризатор. На основе данного кристалла можно построить также дисплеи, отражающие свет.

Дисплеи на жидкких кристаллах имеют адресацию типа строка–столбец либо содержат активные матричные элементы, например транзисторы или диоды в каждом пикселе. Обычная разрешающая способность данных дисплеев составляет 100 пикселов на дюйм.

1-14 ЭЛЕКТРОСТАТИЧЕСКИЕ ГРАФОПОСТРОИТЕЛИ

Электростатические графопостроители принадлежат к типу устройств с растровым сканированием (см. разд. 1-9, 1-12). Основной принцип их функционирования – это нанесение частиц краски на участки бумаги, обладающие определенным электрическим зарядом. На рис. 1-44 представлена обобщенная схема устройств данного типа. Специальное покрытие носителя изображения,держивающее электростатический заряд, помещается около записывающей головки, которая содержит от 1 до 3 рядов пишущих узлов или перьев. Обычно на одном дюйме располагается от 72 до 400 таких перьев. Чем больше рядов пишущих узлов, тем выше плотность печати. При прохождении носителя под рядами перьев каждое из них наносит отрицательно заряженную точку. Далее носитель проходит через тоновый аппликатор, где положительно заряженные частицы краски притягиваются к точкам изображения, имеющим отрицательный заряд, в результате на носителе пропускает видимое изображение. Затем частицы графитовой краски просушиваются и фиксируются на носителе. В качестве материала носителя изображения для графопостроителя может быть использована непрозрачная или полупрозрачная пергаментная бумага и прозрачная или матовая полиэфирная пленка.

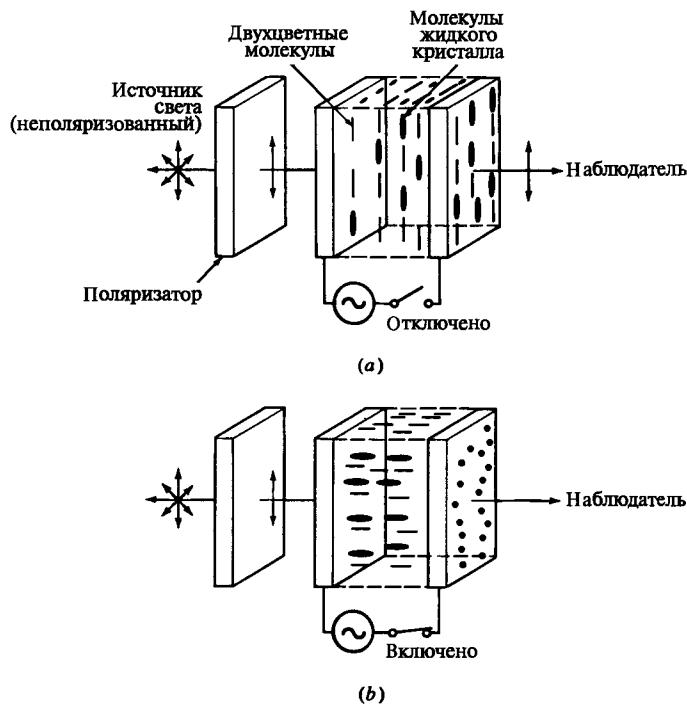


Рис. 1-43 Жидкие кристаллы типа гость–хозяин (закрашены). Молекулы ориентированы для блокировки (а) и для свободного прохождения (б).

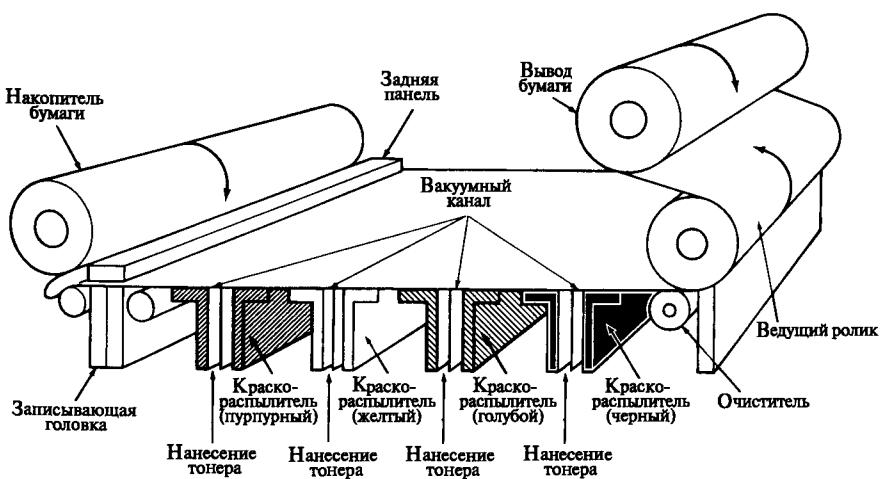


Рис. 1-44 Принципиальная схема цветного электростатического графопостроителя.

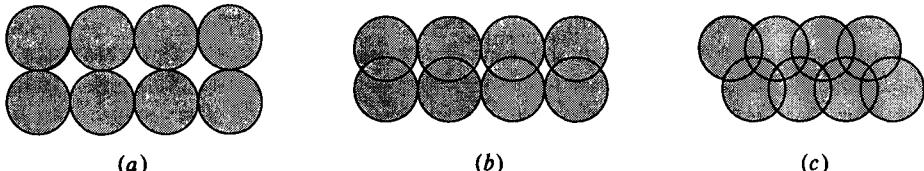


Рис. 1-45 Образцы размещения точек: а) без совмещения, б) совмещение по оси y , в) совмещение по осям x и y .

Типичная ширина электростатических графопостроителей достигает 72 дюймов, а длина изображения ограничивается только длиной рулона материала носителя (обычно это 500 футов). Разрешающая способность данных устройств зависит от количеством точек на дюйм (dpi). Точность определяется несколькими различными способами. Предполагается, что направление движения материала может быть изменено на противоположное, точность в этом случае определяется величиной ошибки, т. е. величиной расхождения в размещении двух точек, нанесенных одним и тем же пером при двух последовательных проходах.

Электростатические графопостроители являются растровыми устройствами вывода, поэтому для них имеет место лестничный эффект [1-1]. В устройствах с высоким разрешением данный эффект устраняется за счет совмещения отдельных точек. На рис. 1-45 показан типичный образец размещения точек. Обычный уровень совмещенности точек составляет от 30% до 50%. В связи с тем что электростатический графопостроитель является устройством с растровым сканированием, элементы изображения, например линии и многоугольники, должны иметь растровую развертку, а само изображение организуется по растровому принципу: сверху вниз, слева направо. Изображение может быть подготовлено на главном компьютере, однако необходимо соблюдать ряд требований организаций вычислений. Кроме этого, для получения высокой скорости обработки картинки необходимо использовать быстрый интерфейс для передачи развернутого по растрю изображения. Например, для графопостроителя шириной 36 дюймов, имеющего разрешение 400 точек на дюйм, требуется передавать данные от главного компьютера со скоростью 360 Кб/с для получения скорости рисования, равной 1 дюйм/с. С целью повышения быстродействия многие электростатические графопостроители работают со специальным контроллером [1-6]. На рис. 1-46 изображен типичный электростатический графопостроитель.

Существуют также цветные графопостроители, в которых используются три основных субтрактивных цвета [1-1] — голубой (cyan), пурпурный (magenta) и желтый (yellow) — и четвертый, черный, которые наносятся на материал за один или четыре прохода. В случае одного прохода, отдельные ряды перьев используются для нанесения каждого из цветов. При нескольких проходах каждый цвет наносится в течение одного прохода. Обычный графопостроитель поддерживает 7 цветов. Цветовая закраска сплошных многоугольников осуществляется путем псевдотонирования или заполнения узором [1-1]. Количество возможных цветов зависит от размеров используемого узора (например, 2×2 , 2×3 или 4×4) и от разрешения графопостроителя. Обычно при высоком разрешении графопостроитель способен выводить несколько тысяч цветов. Стандартная

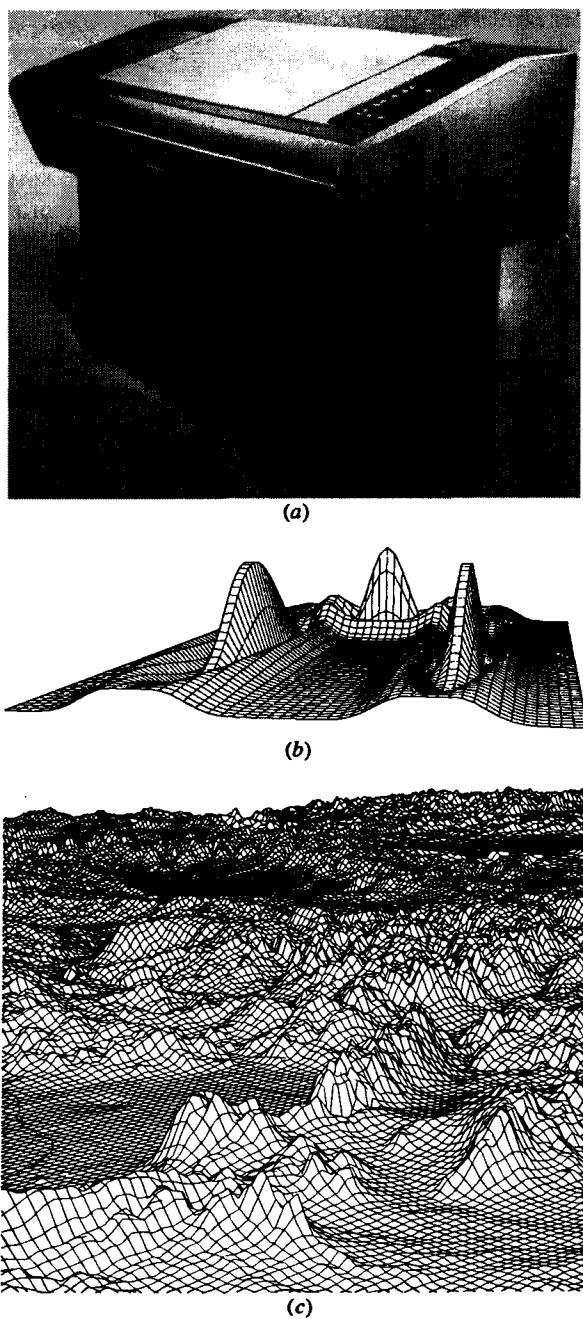


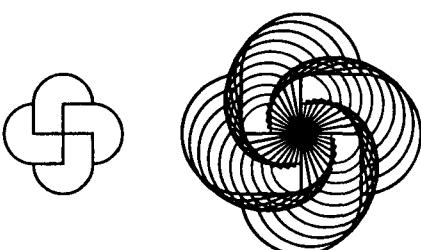
Рис. 1-46 а) Электростатический графопостроитель. (С разрешения фирмы Versatec Corp.) б) Низкое разрешение 100 dpi; в) Высокое разрешение 400 dpi.

ширина электростатических графопостроителей — 42 дюйма, а разрешение — 400 точек на дюйм.

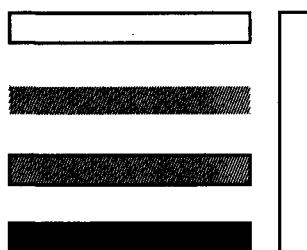
1-15 ЛАЗЕРНЫЕ ПЕЧАТАЮЩИЕ УСТРОЙСТВА

Несмотря на то что лазерные печатающие устройства (принтеры) первоначально создавались для использования в типографском деле, они нашли достаточно широкое применение и в машинной графике при решении задач вывода графических изображений. Возможности этих устройств оказались особенно полезными в различного рода издательских системах, где возникает необходимость в компоновке на одной странице текстов и изображений [1-7]. Такие системы позволяют комбинировать выведенные различными шрифтами тексты, художественные иллюстрации и полутонаовые изображения, как, например, представлено на рис. 1-47.

Лазерный принтер принадлежит к семейству растровых устройств. Принципы его действия, схематично представленные на рис. 1-48, во многом подобны принципам, применяемым при ксерокопировании. Печатающий механизм имеет покрытый фоточувствительным веществом барабан, по поверхности которого может перемещаться луч лазера (газоразрядного или полученного от полупроводникового диода). При вращении барабана его покрытие получает электрический заряд, который сохраняется до момента падения на него когерентного лазерного луча. В результате прохождения луча на барабане появляются зоны (точки), не имеющие заряда, которые в совокупности образуют негативное изображение будущей картинки. Позиционирование луча на барабане осуществляется с помощью вращающегося многогранного отражателя. Черный порошок на пластиковой основе (тонер) притягивается к заряженным точкам на барабане, изображение с которого переносится затем на лист положительно заряженной бумаги. На завершающей стадии формирования изображения происходит фиксация тонера путем его вплавления под действием температуры и давления.



Картинка наверху слева формируется командами `gline()` и `arc()`. Затем путем масштабирования и поворота можно получить различные изображения



Определяя переменную `box`, можно обеспечить вывод прямоугольника различными способами. Используя команды `stroke`, `setgray`, `fill`, можно выводить один и тот же прямоугольник в разных видах и положениях

Рис. 1-47 Лазерный принтер позволяет формировать страницу путем комбинации текстов, иллюстраций и полутонаовых изображений. (С разрешения фирмы QMS, Inc.)

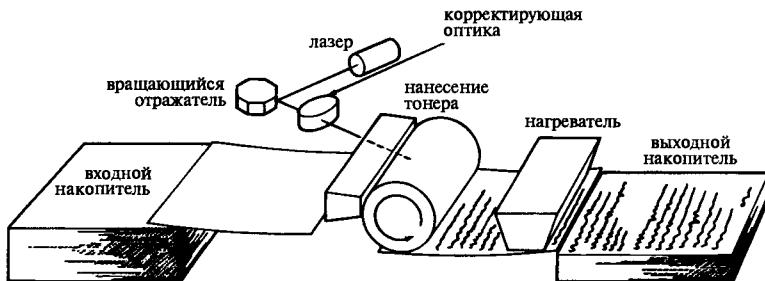


Рис. 1-48 Схема устройства лазерного принтера.

Обычно лазерный принтер имеет разрешение от 240×240 до 600×600 т/дюйм. Наиболее типичное разрешение — это 300×300 т/дюйм. Лабораторные эксперименты показывают, что практически высший предел возможностей данной технологии — 800×1000 т/дюйм. При более высоком разрешении происходит спекание соседних точек тонера. Несмотря на то что лазерные принтеры используются для типографского набора, их обычное разрешение значительно меньше, чем разрешение, полученное при хорошем фотонаборе (1200 — 5000 т/дюйм). Качество современных принтеров ограничено размерами отдельной точки, однако лабораторные эксперименты по изменению размеров точки дают обнадеживающие результаты. Изменение размеров точки позволяет увеличить гибкость при формировании полутонаовых изображений [1-1]. Хотя размеры лазерных принтеров определяют формат используемой для них бумаги, в настоящее время выпускаются устройства шириной 36 дюймов с разрешением 400 т/дюйм, которые позволяют получать изображения форматов D (22×44) и E (34×44) — формат А0 по ЕСКД.

Лазерная печать является достаточно продолжительным процессом. Вывод страницы происходит с постоянной скоростью, так как для получения высокого разрешения требуется и высокая скорость обработки. Например, при разрешении 300×300 для печати со скоростью 8 страниц в минуту при размере $8,5 \times 11$ дюймов, с печатным полем $7,5 \times 10,5$ обеспечение непрерывной бесперебойной работы потребует скорости 1 Мбит/с. Для поддержания такой же высокой скорости, как и в электростатических графопостроителях, используется специальный контроллер, общая схема которого приведена на рис. 1-49. Главный компьютер выдает описание страницы на языке высокого уровня [1-7]. Видеоконтроллер/процессор отвечает за генерацию шрифтов, линий или полутонаовых изображений и за передачу всего необходимого потока битовой информации (точек) на печатающее устройство.

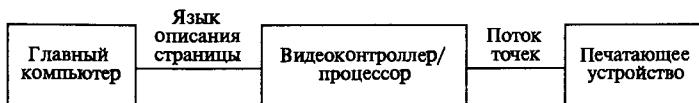


Рис. 1-49 Конфигурация вычислительной системы лазерного принтера.

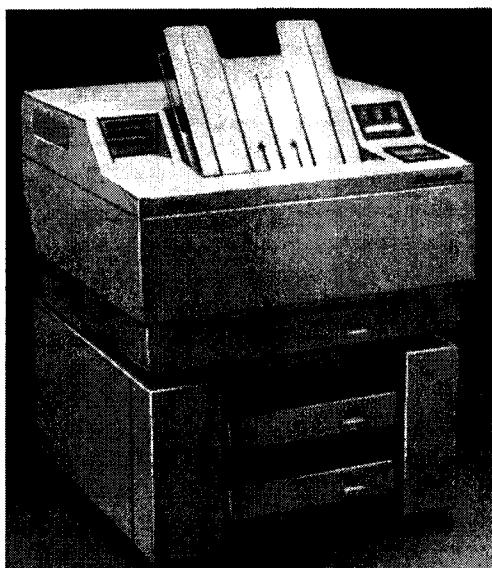


Рис. 1-50 Лазерный принтер. (С разрешения фирмы Toshiba America, Inc.)

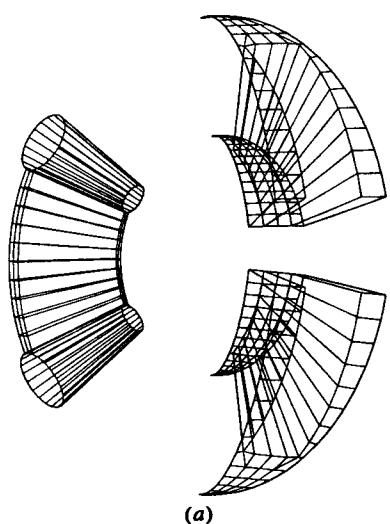


Рис. 1-51 Примеры изображений, выводимых лазерным принтером: а) векторная графика; б) полутоновые поля.

На рис. 1-50 показан типичный лазерный принтер, а на рис. 1-51 приведены примеры обычных векторных и полутонаовых изображений.

1-16 МАТРИЧНЫЕ ГРАФОПОСТРОИТЕЛИ

Принтеры использовались еще на заре становления машинной графики, когда применялись обычные строковые устройства, принадлежащие к типу устройств с растровым сканированием. С помощью таких принтеров можно получать изображения, графики и даже полутонаевые картинки. Вполне естественно, что при разрешении 1/6 дюйма по вертикали и 0.1 по горизонтали изображения были довольно грубыми.

В машинной графике применяется множество принтеров, основанных на использовании различных методов и принципов печати: ленточные или построчные устройства, барабанные принтеры, цепные печатающие устройства и лепестковые принтеры. Однако наибольшую популярность и распространение получили именно матричные устройства.

Печатающий механизм в матричных принтерах представляет собой матрицу из тонких перьев или иголок, закрепленную на печатающей головке, которая перемещается вдоль страницы. Основу механизма составляет электромагнит для одной или нескольких стальных пружин, смонтированных на иглах или молоточках. В состоянии покоя молоточки удерживаются постоянным магнитом и не со-прикасаются с красящей лентой. Для нейтрализации действия этого постоянного магнита используется электромагнит. При его включении стальные пружины подают молоточки вперед, вызывая соприкосновение иголки с лентой. В результате на бумагу, которая расположена между лентой и опорой, наносится точка. После этого электромагнит отключается, молоточек возвращается в первоначальное положение. В конструкциях со свободными молоточками их возврат и освобождение происходит в состоянии возбуждения электромагнита.

Существует множество различных конфигураций расположения иголок (рис. 1-52). При невысокой плотности размещения иголок размером от 0.012 до 0.013 дюйма получается изображение точки размером 0.014–0.015 дюймов, что позволяет иметь разрешение от 60 до 80 т/дюйм. При большей плотности размещения иголок размером 0.008 дюймов образуется точка размером 0.009–0.010 дюйма, а разрешение составляет 180 т/дюйм. Таким образом, видно, что при высокой плотности точек получается отпечаток с перекрытием. В зависимо-



Рис. 1-52 Образцы размещения иголок в матричном печатающем устройстве.

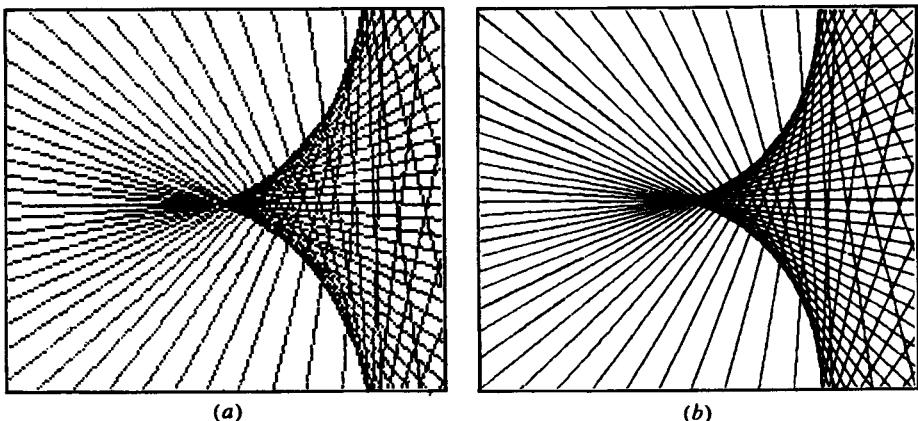


Рис. 1-53 Сравнение качества вывода матричных печатающих устройств: а) 9 иголок — разрешение 70 точек на дюйм; б) 24 иголки — разрешение 180 точек.

сти от плотности точек перекрытие может составить от 20 до 50%. Перекрытие точек уменьшает уровень смещения или ступенчатый эффект [1-1], что способствует улучшению качества изображения. На рис. 1-53 представлена картинка, полученная на матричных принтерах с разрешением 70 и 180 т/дюйм и иллюстрирующая данный эффект.

Цветное изображение на матричных печатающих устройствах получается с помощью многоцветных красящих лент. Так как бумага обладает средней отражающей способностью, то используется четырехцветная лента, на которой имеются три основных цвета: голубой, пурпурный, желтый (CMY), а также черный цвет. Использование методов псевдотонирования и заполнения узорами [1-1, 1-8] позволяет получать оттенки разного цвета.

1-17 УСТРОЙСТВО СТРУЙНОЙ ПЕЧАТИ

Устройства струйной печати являются растровыми устройствами и используются главным образом в недорогих системах формирования цветных изображений. Основной принцип струйной печати состоит в распылении микроскопических капель краски на носитель изображения. Существуют два типа устройств струйной печати: с непрерывным распылением и с распылением по запросу.

Устройства первого типа формируют поток частиц путем распыления краски через форсунку. Такой поток прерывается с помощью ультразвуковых волн. Капельки краски, которые должны быть нанесены на бумагу, получают электрический заряд. Специальные отражательные пластины управляют направлением движения частиц. Все незаряженные частицы отклоняются таким образом, что они попадают в желоб и возвращаются затем в резервуар. Обычно в качестве носителя изображения используется бумага или прозрачная пленка. Типичная система струйной печати показана на рис. 1-54а. В случае применения технологии распыления по запросу, как следует из ее названия, выброс краски происходит, только если требуется закрасить определенную точку. Краска поступает из ре-

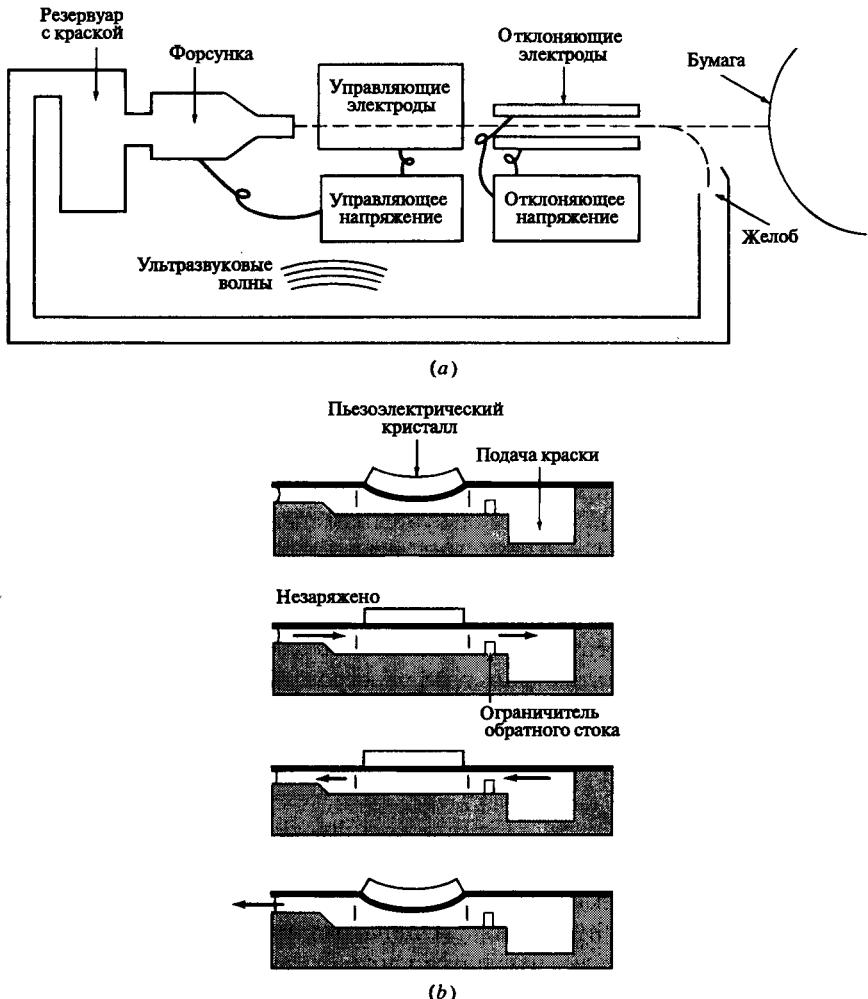


Рис. 1-54 Принципиальная схема технологии струйной печати: а) непрерывное распыление; б) распыление по запросу.

зерьуара и попадает под давлением в форсунку. Далее с помощью пьезоэлектрического кристалла, на который подается напряжение, происходит впрыскивание краски через голову форсунки. Кристалл расширяется под действием напряжения, уменьшая объем камеры с краской, что и вызывает ее выброс из форсунки. При отключении напряжения кристалл сжимается, увеличивая объем резервуара, и это приводит к всасыванию краски обратно в форсунку. На рис. 1-54б представлена иллюстрация данного процесса.

Разрешающая способность устройств струйной печати определяется размерами капли краски, а следовательно, и форсунки. При использовании максимально узкой форсунки, возникает ряд проблем с обеспечением свободного прохожде-

ния краски, предотвращением загрязнения форсунки, а также с удалением пузырьков воздуха. Обычное разрешение устройств данного типа составляет 120–400 т/дюйм. Лабораторные эксперименты показывают, что возможно разрешение в 1000 т/дюйм.

Цветные устройства струйной печати имеют, как правило, четыре форсунки, три для основных цветов — голубого, пурпурного, желтого — и одну для черного цвета. Основное преимущество струйной технологии состоит в способности смешивать цвета. Так как жидкие частицы краски наносятся за один проход носителя, они успевают перемешаться еще до высыхания. Это позволяет получать глубину и резкость цвета, не достижимую при использовании других технологий. Применение методов псевдотонирования и узоров, как уже отмечалось, позволяет получать несколько тысяч цветов [1-1, 1-8].

1-18 УСТРОЙСТВО ТЕРМОПЕЧАТИ

Различают два основных принципа термопечати: с прямым и косвенным воздействием тепла. В первом случае используется специальная термоочувствительная бумага, которая изменяет свой цвет под действием тепла. Изображение на такой бумаге формируется печатающим узлом, который воздействует на бумагу в процессе движения. Устройство пишущего узла матричного принтера и печатающие головки в виде полного ряда пишущих узлов аналогичны устройству узлов, используемых в электростатических графопостроителях. Устройства термопечати позволяют получать разрешение в 40 т/дюйм. Главные и наиболее серьезные ограничения технологии с прямой передачей тепла связаны с необходимостью применения специальной бумаги, которая имеет свойство выцветать с течением времени.

В технологии с косвенным действием тепла используется тонкая пленка или бумажная лента, покрытая чернилами на восковой основе. Тепловыделяющие элементы, обычно располагаемые позади ленты, расплавляют восковое покрытие, и пигменты чернил попадают на бумагу. Однаково часто употребляются матричные печатающие узлы, перемещающиеся вдоль бумаги, и печатающие головки в виде полного ряда пишущих узлов. На рис. 1-55 представлена схема технологии

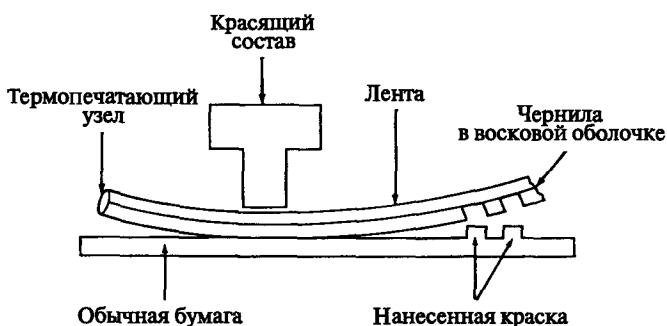


Рис. 1-55 Технология с косвенным тепловым воздействием.

с косвенным воздействием тепла. В качестве основы для нанесения изображения применяется обычная бумага. Цветные картинки получаются путем использования многоцветных лент. Традиционные семь цветов возникают в результате комбинации трех основных цветов (CMY) и черного цвета. Для формирования расширенной цветовой палитры применяются шаблоны и штриховка [1-1, 1-8]. Так как чернила почти не впитываются в бумагу, то можно получать блестящие цвета. Обычная разрешающая способность для устройств данного типа составляет от 100 до 400 т/дюйм. Чернила, употребляемые при технологии с косвенным воздействием, отличаются долговечностью.

1-19 ПЕРЬЕВЫЕ ГРАФОПОСТРОИТЕЛИ.

Перьевые графопостроители (прежде всего цифровые) делятся на три вида: планшетные, барабанные и графопостроители с нажимным роликом. На рис. 1-56 даны принципиальные схемы каждого из этих устройств.

В планшетном графопостроителе с подвижным кронштейном (рис. 1-56a) носитель изображения закрепляется на ложе графопостроителя. Перемещение пишущего узла на плоскости осуществляется при движении кронштейна по ширине ложа для оси x и при движении собственно узла по кронштейну для оси y . На

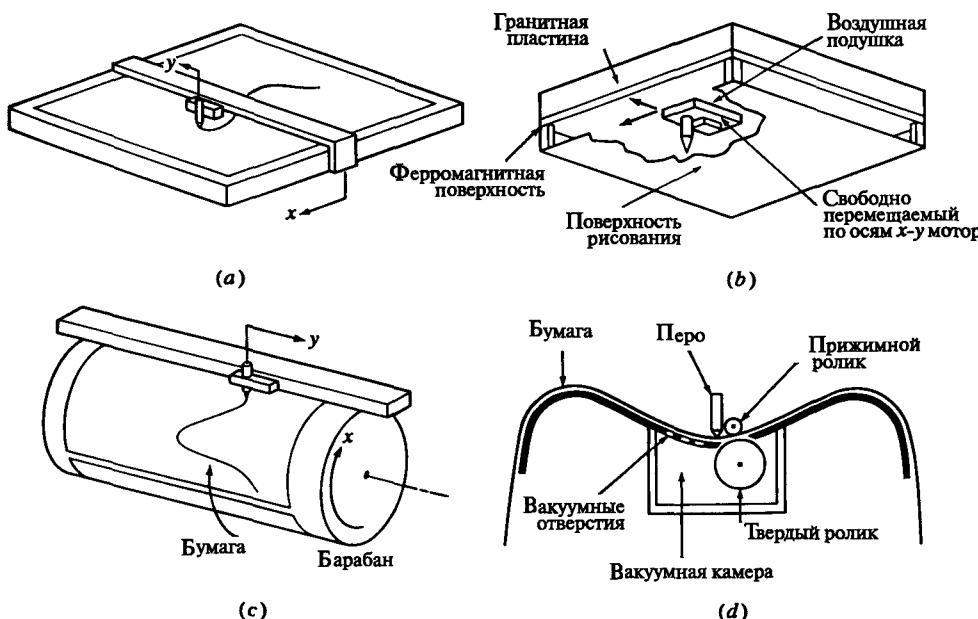


Рис. 1-56 Схемы действия перьевых графопостроителей разных типов: а) планшетный с подвижным кронштейном; б) планшетный с подвижным пишущим узлом; в) барабанный; г) с нажимным роликом.



Рис. 1-57 Планшетный графопостроитель с подвижным кронштейном. (С разрешения фирмы Xynetics, Inc.)



Рис. 1-58 Планшетный графопостроитель с подвижным пишущим узлом.

рис. 1-57 приведен типичный планшетный графопостроитель с подвижным кронштейном.

В планшетном графопостроителе с подвижным пишущим узлом (рис. 1-56б) применяется устройство рисования, которое, находясь в подвешенном состоянии, перемещается под действием магнитных сил. Данный принцип обеспечивает почти полное отсутствие трения. Управление движением пишущего узла по двум направлениям осуществляется на основе электромагнитного воздействия по принципу сервомотора. Пример графопостроителя данного типа представлен на рис. 1-58.

Крепление носителя изображения на ложе крупноформатных планшетных графопостроителей осуществляется с помощью вакуума. Данные устройства имеют разрешение не менее 0.004 дюйма и скорость 3600 дюйм/мин. Размеры планшетных графопостроителей колеблются от 8.5 × 11 дюймов до нескольких десятков футов в ширину и длину. В качестве носителя изображения может ис-

пользоваться множество материалов, таких, как бумага, пергамент, фотопленка, копировальная поверхность, лист металла и ткани. Этим разнообразным типам носителей соответствует множество устройств рисования, среди которых шариковая ручка, фломастеры и обычные чернильные перья, световые и лазерные лучи, инструменты для гравировки, копировальные резцы и резцы по ткани. Такое разнообразие типов носителей и пишущих узлов предполагает широкий спектр различных сфер применения планшетных графопостроителей. Кроме их традиционного использования для рисования, крупноформатные графопостроители применяются для создания шаблонов печатных плат и интегральных схем. Носителем изображения в этом случае является фотопленка, а пишущим инструментом луч света. Различные модификации планшетных графопостроителей используются также и для раскроек мужской и женской одежды и даже парусов. В качестве носителя здесь выступает соответствующая ткань, а пишущим узлом является нож либо лазерный луч.

Более дешевые и небольшие планшетные графопостроители не обладают такой гибкостью применения и имеют более низкую скорость и разрешающую способность. Обычно в таких устройствах используется принцип электростатического управления носителем, в качестве которого, как правило, применяется бумага, пергамент или прозрачная пленка.

Барабанные графопостроители (рис. 1-56с) являются более сложными в механическом плане устройствами, чем планшетные. Пишущий узел в них перемещается вдоль неподвижного кронштейна в одном направлении. Сам носитель вращается вперед и назад под кронштейном, обеспечивая таким образом отрисовку по другому направлению.

Носитель закрепляется на барабане при помощи вакуума либо прижимных роликов. Тип материала носителя определяется его свойствами легко вращаться. Обычно это бумага или пергамент определенной ширины. Длина носителя ограничена только длиной ленты в рулоне. Несмотря на то что разрешающая способность графопостроителей данного типа примерно равна разрешению планшетных устройств (0.0005 дюйма), скорость рисования в них, как правило, значительно ниже.

Графопостроители с прижимным роликом представляют собой смесь планшетных и барабанных устройств. Носитель изображения размещается в них между прижимным роликом и цилиндрической поверхностью рисования (рис. 1-56д). При вращении прижимного ролика носитель движется вперед и назад под неподвижным кронштейном, по которому, в свою очередь, перемещается пишущий узел. В этих графопостроителях применяется накопитель для отдельных листов либо для рулона, что и определяет формат бумаги, пергамента или прозрачной пленки. Обычная разрешающая способность данных устройств составляет 0.001 дюйм, а скорость равна 180 дюйм/мин. На рис. 1-59 показан графопостроитель с прижимным роликом.

Большинство перьевых графопостроителей работают в инкрементальном режиме и приводятся в движение шаговым мотором. Пишущий узел в них перемещается над поверхностью рисования по восьми возможным направлениям (рис. 1-60) за некоторое число небольших шагов. Результирующая линия, проведенная в направлении, отличном от указанных на рис. 1-60, будет иметь ступенчатый ха-

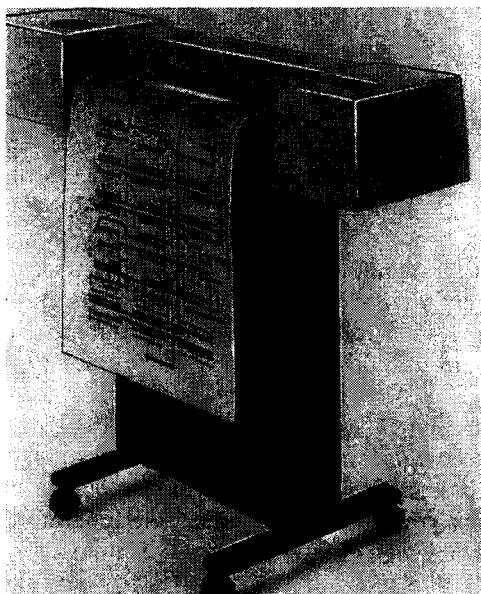


Рис. 1-59 Графопостроитель с прижимным роликом. (С разрешения фирмы Hewlett-Packard. Co.)

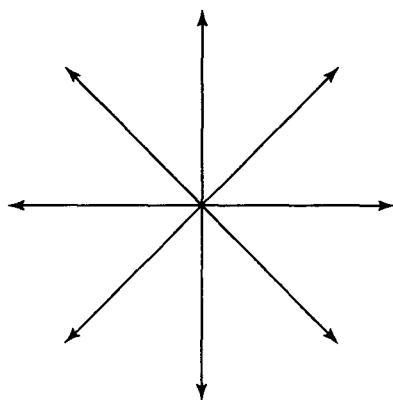


Рис. 1-60 Направления движения в инкрементальных графопостроителях.

рактер. Появление такой «лестницы», или зазубренной линии, является следствием ключевого явления, получившего название лестничный эффект (алиазинг), т. е. дефект изображения линии вследствие дискретности раstra [1-1]. Из всех алгоритмов построения линии, имеющих наилучшие показатели оптимальности выбора из восьми возможных направлений, алгоритм Брезенхема [1-1] является наиболее удачным. Данный алгоритм, часто ассоциируемый с растровыми устройствами, первоначально был разработан именно для цифровых инкрементальных графопостроителей.

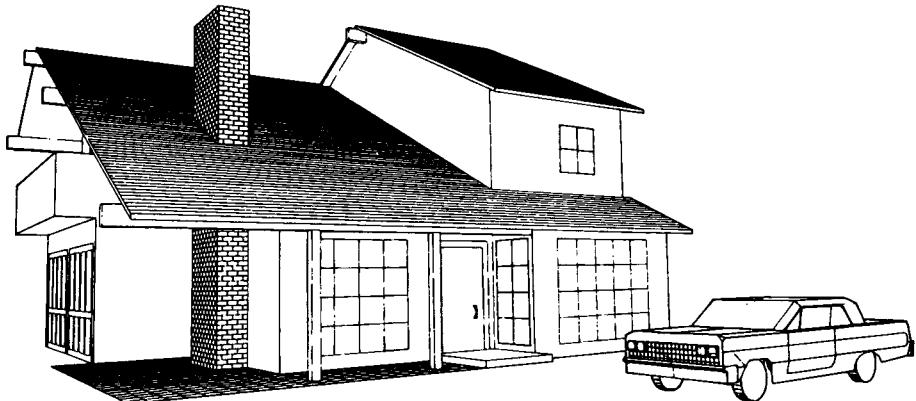


Рис. 1-61 Пример типичного изображения, получаемого на цифровом инкрементальном графопостроителе.

Корректность направления движения достигается путем использования сервомоторов, которые управляют пишущим узлом графопостроителя. В этом случае обратная связь в цепи постоянно корректирует одновременно оба мотора для получения гладкой линии.

Скорость является важной, но не единственной существенной характеристикой графопостроителя. В некоторых случаях ускорение оказывает гораздо большее влияние на общее быстродействие. Дело в том, что чертеж состоит из коротких сегментов линий и поэтому графопостроитель должен уметь стартовать и останавливаться между сегментами. Большое ускорение позволяет графопостроителю быстрее достигать нужной скорости. Возможная в настоящее время величина ускорения превышает 4 g. Многие контроллеры графопостроителей комплектуются также специальными программами, позволяющими заранее избегать лишних остановок, например, в случае, когда для представления кривой используется последовательность соединенных друг с другом коротких отрезков.

В отличие от многих графических устройств, перьевые графопостроители позволяют получать достаточно высококачественные изображения, такие как на рис. 1-61.

1-20 ЦВЕТНЫЕ ВИДЕОКАМЕРЫ

С появлением высококачественных цветных растровых дисплеев в сочетании с цветными буферами кадров (§ 1-9), способных выводить свыше 16 млн. цветов, возникла необходимость в способе представления полученных результатов. В настоящее время существует единственная такая технология, основанная на использовании цветной фотопленки. Цветная видеокамера — это наиболее часто используемое в машинной графике устройство.

Совсем простая принципиальная схема цветной камеры приведена на рис. 1-62. В методе, называемом записью последовательных полей, используется небольшая, диаметром 5 дюймов, монохроматическая ЭЛТ с идеально плоским

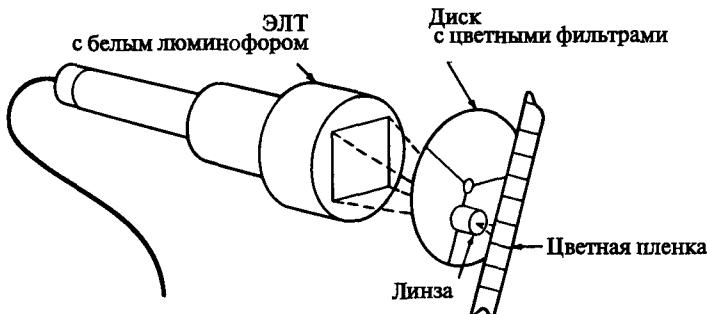


Рис. 1-62 Принципиальная схема цветной камеры.

экраном. На оптической оси от камеры к ЭЛТ сначала помещается красный фотфильтр, пропускающий только красные лучи от ЭЛТ, которые и попадают на пленку. После этого экран ЭЛТ гаснет, красный фильтр заменяется на зеленый, открывающий путь только зеленым лучам. Пленка повторно экспонируется, оставаясь в прежнем положении. В заключение процесс повторяется для синей компоненты изображения. Такая трижды экспонированная пленка содержит за-конченное изображения рисунка.

Успех всего процесса определяется качеством ЭЛТ [1-1], для уменьшения в ней искажений применяется плоский, а не изогнутый экран, устанавливаемый на большинство мониторов. Стекло ЭЛТ также должно быть высокого качества, без пузырьков, царапин и других дефектов. Для того чтобы обеспечить примерно одинаковую реакцию на все три сигнала (красный, зеленый, синий), применяется широкополосный люминофор, слой которого должен быть без пятен и узоров, иметь прекрасную зернистость и однородную текстуру. Размеры точки электронного луча должны быть минимальными размерами для достижения повышенной разрешающей способности, а электромагнитная фокусирующая система должна быть отличного качества.

При условии аккуратной эксплуатации разрешение такого устройства составляет 3000×5000 т/дюйм при диаметре монитора 5 дюймов. Отметим, что разрешение и адресуемость — это не идентичные понятия. Разрешением измеряется количество отдельных различимых точек, в то же время адресуемость указывает число позиций, в которые может быть помещен электронный луч. Из-за ограниченности возможностей человеческого глаза картинка может иметь адресуемость выше порога, за которым глаз уже не различает отдельных точек изображения.

1-21 АКТИВНЫЕ И ПАССИВНЫЕ ГРАФИЧЕСКИЕ УСТРОЙСТВА

Разберем различия между пассивными и активными графическими устройствами. Пассивные устройства просто формируют под управлением компьютера изображения, т. е. они позволяют обеспечивать графический интерфейс между компьютером и пользователем. Матричные принтеры, электростатические и перьевые графопостроители, устройства записи на пленку, дисплеи с запоминающей трубкой, а также мониторы с растровой ЭЛТ — все это примеры пассивных гра-

фических устройств. Некоторые образцы этих устройств вместе с типичными картинками, которые они способны формировать, приведены на рис. 1-17, 1-22, 1-37, 1-40, 1-46, 1-50, 1-51, 1-53, 1-57, 1-58, 1-59 и 1-61.

Активные графические устройства позволяют пользователю взаимодействовать с компьютером при помощи графики. Обычно это означает, что пользователь получает координатную информацию в некотором преобразованном виде, т. е. в виде, отличном от явного вывода соответствующих чисел. Так как картинки, кривые или поверхности можно рассматривать в виде матрицы координат, то пользователь получает достоверную иллюстративную информацию. Обычно активные устройства обладают возможностью позиционирования курсора и считывания его нового положения. Типичные активные графические устройства включают алфавитно-цифровые клавиатуры (рис. 1-6), функциональные кнопки (рис. 1-13), координатные колеса, наборы ручек для ввода скалярных значений (рис. 1-12), оцифровывающий или аналоговый планшеты (рис. 1-7, 1-8), световые перья (рис. 1-15), управляющие рычаги (рис. 1-9), координатный шар (рис. 1-10) или мышь (рис. 1-11). Несмотря на то что эти устройства могут использоваться самостоятельно, обычно они требуют применения каких-либо пассивных устройств. Как правило это ЭЛТ.

1-22 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МАШИННОЙ ГРАФИКИ

Весь комплекс программного обеспечения машинной графики не только реализует идеи, затронутые в данной книге и в работе [1-1], но и воплощает алгоритмы организации структур данных, проектирования и управления базами данных, идеологию и требования к человеко-машинному интерфейсу, охватывает языки программирования и операционные системы. Исследование всех этих вопросов выходит за рамки данной книги, однако они затрагиваются при обсуждении вопросов проектирования систем машинной графики.

В настоящее время разработаны и продолжают создаваться многочисленные графические стандарты, которые в общем случае можно объединить в три группы. Первая группа — это графический прикладной интерфейс, преобразующий идеи в форму, понятную компьютеру. Для этих целей применяются стандарты представления данных: графическая корневая система (GKS), иерархический интерактивный графический стандарт программиста (PHIGS, PHIGS+) (приложение А).

Вторая группа относится к хранению и передаче данных между графическими системами и между САПР и системами АСТПП. В данной области имеется стандарт IGES (формат передачи графических данных).

В третью группу входит стандарт для определения интерфейса между программными системами машинной графики или наборами данных и системой передачи, а также стандарт определения требований к аппаратно зависимым драйверам графических устройств. В данной области существуют стандарты: машинно-графический метафайл (CGM) и виртуальный интерфейс устройств (VDI).

Все эти стандарты имеют очень важное значение и представляют собой попытку обеспечения совместимости программ, пользователей и данных. В приложении А эти стандарты рассматриваются более развернуто, но не очень детально.

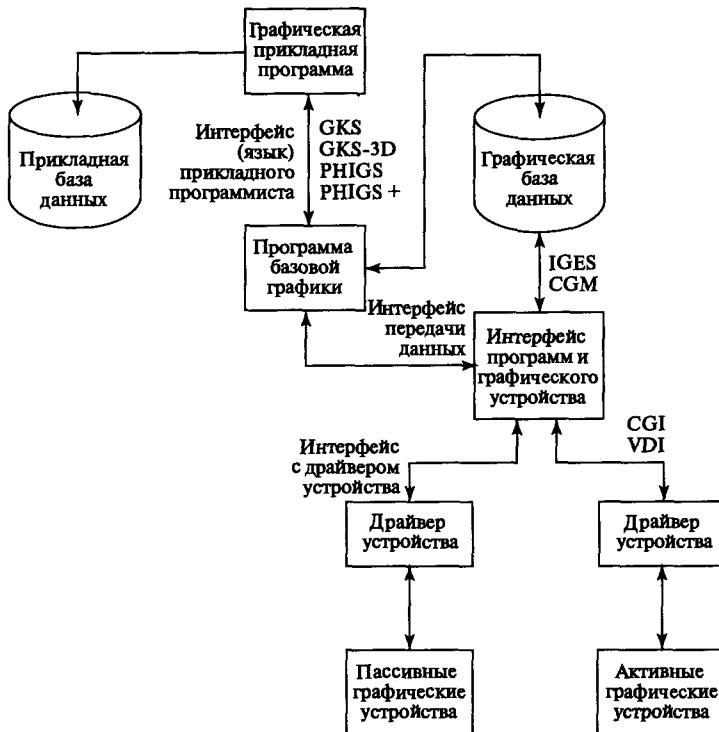


Рис. 1-63 Структура программного обеспечения машинной графики.

На рис. 1-63 показаны основные уровни организации программного обеспечения машинной графики и место каждого из стандартов. Однако задача этой книги и работы [1-1] состоит не в обсуждении стандартов, а в исследовании основополагающих концепций, математических и процедурных элементов машинной графики. Поэтому программам машинной графики уделяется минимальное внимание на уровне основных понятий. Программные системы машинной графики, реализующие все эти концепции, представлены в приложениях А и Г. На основе всех изложенных и разбираемых впоследствии концепций из этой книги и из работы [1-1], а также концепций организации структур данных и проектирования баз данных можно разработать и реализовать функционально полную программную систему машинной графики. Эти концепции помогут также разобраться в любой из существующих систем машинной графики.

1-23 ЛИТЕРАТУРА

- 1-1 Rogers D. F. *Procedural Elements for Computer Graphics*, McGraw-Hill, Book Co., New York, 1985. (Имеется русский перевод: Роджерс Д. Алгоритмические основы машинной графики. — М.: Мир, 1989.)
- 1-2 Foley J. D., Wallace V. L. *The Art of Natural Man-Machine Conversation*, Proc. IEEE, Vol. 62, pp. 462–471, 1974.

- 1-3 Wallace V. L. The Semantics of Graphic Input Devices, *Comp. Graph.*, Vol. 10, pp. 61–65, 1976.
- 1-4 Olson M. System Design Considerations for Graphics Input Devices, *Computer*, pp. 9–18, Nov. 1978.
- 1-5 Davis M. R., Ellis T. O. The RAND Tablet: A Man-Machine Graphical Communication Device, *AFIPS Conf. Proc.*, Vol. 26, Part I, 1964 FJCC, pp. 325–332, 1964.
- 1-6 Ben-Dor A., Jones B. New Graphics Controller for Electrostatic Plotting, *IEEE Comp. Graph. & Appl.*, Vol. 6, No. 1, pp. 16–25, 1986.
- 1-7 Adobe Systems Inc., *Postscript Language Reference Manual*, Addison-Wesley Publishing Co., Reading, MA, 1985.
- 1-8 Kubo S. Continuous Color Presentation Using a Low-Cost Ink Jet Printer, *Proc. Comp. Graph. Tokyo, 84*, 24–27. Apr. 1984, Tokyo, Japan, T3–6, pp. 1–10, 1984.
- 1-9 Constantine J. M., Jr., Electron Optical Technique for an Ultra High Resolution Color Film Recorder, *Journal of Photooptical Instrument Engineers*, 1984.

Глава 2

ДВУМЕРНЫЕ ПРЕОБРАЗОВАНИЯ

2-1 ВВЕДЕНИЕ

Изучение математического аппарата, лежащего в основе машинной графики, мы начнем с рассмотрения способов вывода и преобразования точек и линий. Эти способы наряду с соответствующими алгоритмами рисования используются для изображения объектов или визуализации графической информации. Возможность проводить преобразования точек и линий является фундаментом машинной графики. Нарисованный объект может быть представлен в нужном масштабе, повернут, перемещен, преобразован или модифицирован в соответствии с требованиями решаемой задачи. Все эти манипуляции с изображением можно выполнить, используя математический аппарат, изложенный в данной и последующих главах.

2-2 ИЗОБРАЖЕНИЕ ТОЧЕК

Точка представляется на плоскости двумя своими координатами, которые определяются как элементы матрицы размером 1×2 $[x \ y]$. В трехмерном пространстве используется матрица размером 1×3 $[x \ y \ z]$.

Иначе говоря, точка может задаваться в виде вектор-столбца $\begin{bmatrix} x \\ y \end{bmatrix}$ в двумерном пространстве или в виде

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

в трехмерном. Строку $[x \ y]$ или столбец $\begin{bmatrix} x \\ y \end{bmatrix}$ часто называют координатным вектором. В этой книге для формирования такого вектора используется матрица-строка, т. е. множество точек, каждая из которых определяет координатный вектор в некоторой системе измерения. Данное множество хранится в компьютере в виде матрицы или массива чисел. Положением точек можно управлять путем манипулирования соответствующей матрицей. Линии, соединяющие точки, формируют отрезки, кривые и картинки.

2-3 ПРЕОБРАЗОВАНИЯ И МАТРИЦЫ

В качестве элементов матрицы могут фигурировать различные величины: числа, сетки или коэффициенты системы уравнений. Правила в матричной алгебре определяют допустимые операции над элементами (приложение В). Многие физические задачи удобно выражаются в матричном представлении. Для моделей физических систем задача обычно ставится следующим образом: даны матрицы $[A]$ и $[B]$, найти результирующую матрицу $[T]$, такую, что $[A][T] = [B]$. В этом случае решением является матрица $[T] = [A]^{-1}[B]$, где $[A]^{-1}$ — матрица, обратная к квадратной матрице $[A]$.

В то же время матрицу $[T]$ можно интерпретировать как геометрический оператор. В этом случае для выполнения геометрического преобразования точек, представленных векторами положений в матрице $[A]$, используется умножение матриц. Предположим, что матрицы $[A]$ и $[T]$ известны. Требуется определить элементы матрицы $[B]$. Представление $[T]$ как геометрического оператора является основой математических преобразований, используемых в машинной графике.

2-4 ПРЕОБРАЗОВАНИЕ ТОЧЕК

Рассмотрим результаты умножения матрицы $[x \ y]$, содержащей координаты точки P , на матрицу общего преобразования размером 2×2 :

$$[X][T] = [x \ y] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [(ax + cy)(bx + dy)] = [x^* \ y^*]. \quad (2-1)$$

Данная запись означает, что исходные координаты точки x и y преобразуются в x^* и y^* , где $x^* = ax + cy$, $y^* = bx + dy$ (приложение В). Представляют интерес значения x^* , y^* — координаты результирующей, преобразованной точки P . Рассмотрим некоторые специальные случаи.

При $a = d = 1$ и $c = b = 0$ преобразование сводится к единичной матрице

$$[X][T] = [x \ y] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [x \ y] = [x^* \ y^*], \quad (2-2)$$

и координаты точки P останутся неизменными. Как и следовало ожидать, в линейной алгебре умножение на единичную матрицу эквивалентно умножению на 1 в обычной алгебре.

В случае $d = 1$, $b = c = 0$

$$[X][T] = [x \ y] \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix} = [ax \ y] = [x^* \ y^*], \quad (2-3)$$

где $x^* = ax$ — результат масштабирования координаты X . Эффект такого преобразования показан на рис. 2-1а. Рассмотрим теперь еще случай $b = c = 0$, т. е.

$$[X][T] = [x \ y] \begin{bmatrix} a & 0 \\ 0 & d \end{bmatrix} = [ax \ dy] = [x^* \ y^*]. \quad (2-4)$$

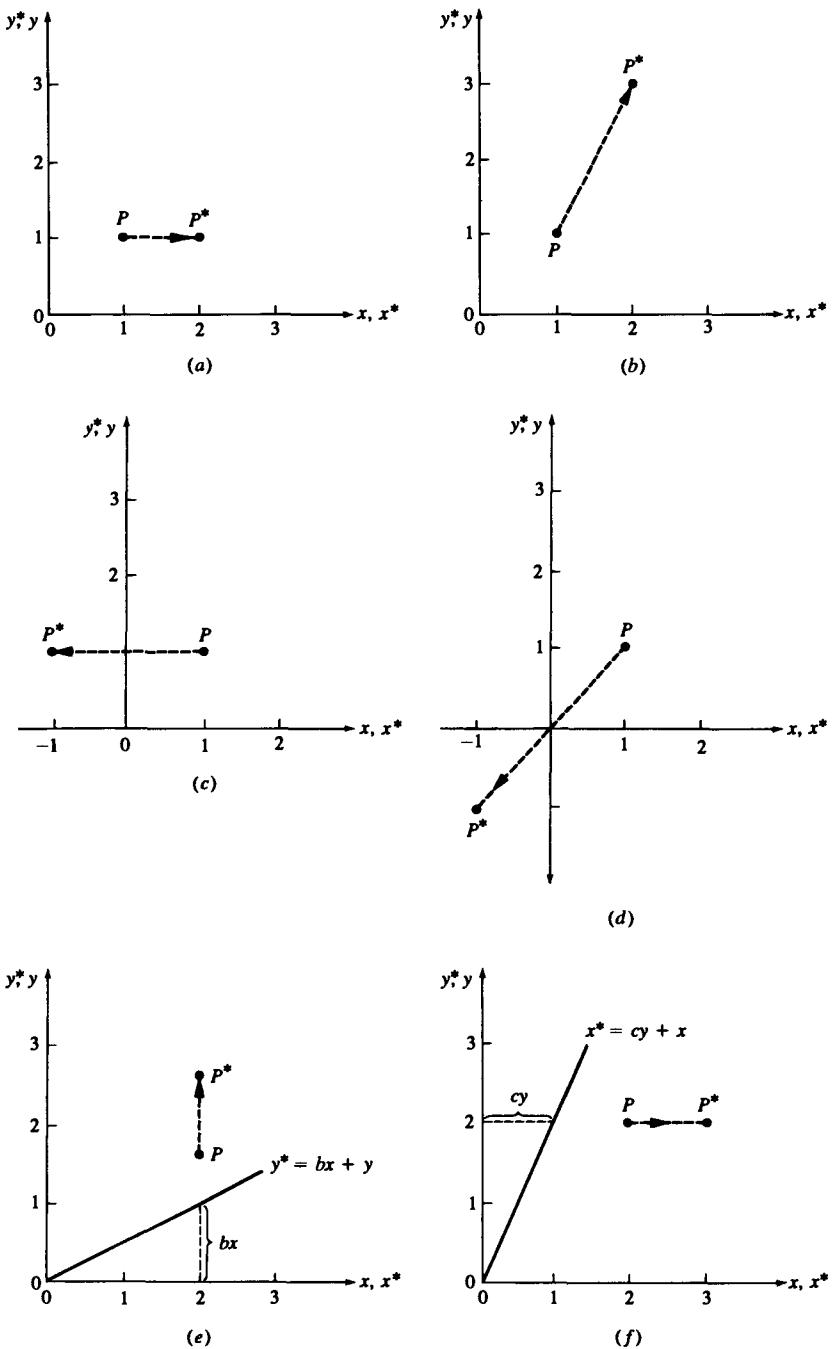


Рис. 2-1 Преобразование точек.

Данное преобразование вызывает изменение обеих координат x и y вектора P (рис. 2-1b). Если $a \neq d$, то координаты масштабируются различным образом. При $a = d > 1$ происходит растяжение вектора P или масштабирование координат. Если $0 < a = d < 1$, то имеет место сжатие.

Если значение a или d отрицательное, то вектор отражается относительно координатных осей или относительно плоскости. Чтобы убедиться в этом, возьмем $b = c = 0$, $d = 1$ и $a = 1$, тогда

$$[X][T] = [x \ y] \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = [-x \ y] = [x^* \ y^*], \quad (2-5)$$

и в результате получаем симметричное отражение относительно оси y (рис. 2-1c). Если $b = c = 0$, $a = 1$, $d = -1$, то выполняется симметричное отражение относительно оси x . Если $b = c = 0$, $a = d < 0$, то происходит отражение относительно начала координат, это показано на рис. 2-1d, где $a = -1$, $d = 1$. Заметим, что обе операции отражения и масштабирование зависят только от диагональных членов матрицы преобразования.

Рассмотрим теперь случай с недиагональными членами. Возьмем сначала значения $a = d = 1$, $c = 0$, тогда

$$[X][T] = [x \ y] \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} = [x \ (bx + y)] = [x^* \ y^*]. \quad (2-6)$$

Заметим, что координата x точки P осталась неизменной, тогда как координата y^* линейно зависит от исходных координат. Данное преобразование называется сдвигом (рис. 2-1e). Аналогично, в случае, когда $a = d = 1$, $b = 0$, преобразование приведет к сдвигу пропорционально координате y (рис. 2-1f). Таким образом, видно, что недиагональные члены матрицы преобразования создают эффект сдвига координат вектора точки P .

Прежде чем закончить с преобразованием точек, разберем действие общего преобразования, заданного выражением (2-1), когда начальный вектор лежит в точке начала координат, т. е.

$$[x \ y] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [(ax + cy) \ (bx + dy)] = [x^* \ y^*]$$

или, в случае начала координат,

$$\begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} = [x^* \ y^*].$$

Видно, что начало координат инвариантно относительно преобразования общего вида. Это ограничение устраняется при использовании однородных координат.

2-5 ПРЕОБРАЗОВАНИЕ ПРЯМЫХ ЛИНИЙ

Прямую линию можно определить с помощью двух векторов, задающих координаты ее конечных точек. Расположение и направление линии, соединяющей две

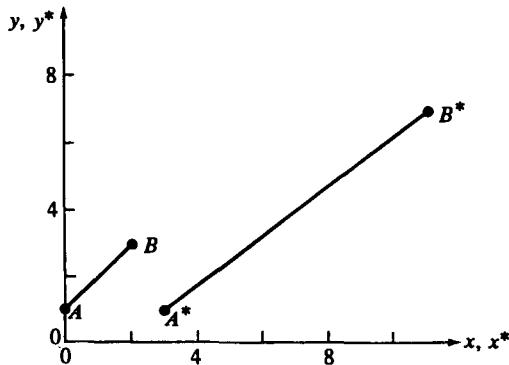


Рис. 2-2 Преобразование отрезков.

эти точки, может изменяться в зависимости от положений векторов. Реальный вид изображения линии зависит от типа используемого дисплея. В этом разделе мы рассмотрим только математические операции над конечными точками линии.

На рис. 2-2 изображена прямая линия, проходящая между двумя точками A и B . Положение векторов точек A и B задается следующим образом: $[A] = [0 \ 1]$ и $[B] = [2 \ 3]$.

Рассмотрим матрицу преобразования

$$T = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}, \quad (2-7)$$

которая, как следует из предыдущего обсуждения, приводит к сдвигу изображения. Преобразование векторов A и B с помощью матрицы $[T]$ дает новое положение векторов A^* и B^*

$$[A][T] = [0 \ 1] \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} = [3 \ 1] = [A^*] \quad (2-8)$$

и

$$[B][T] = [2 \ 3] \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} = [11 \ 7] = [B^*]. \quad (2-9)$$

Таким образом, результирующие координаты для точки A^* — это $x^* = 3$ и $y^* = 1$. Аналогично, B^* — новая точка с координатами $x^* = 11$, $y^* = 7$. В более компактном виде отрезок AB может быть представлен матрицей размером (2×2) :

$$L = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}.$$

Умножим эту матрицу на $[T]$:

$$[L][T] = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 11 & 7 \end{bmatrix} = [L^*], \quad (2-10)$$

где компоненты $[L^*]$ представляют собой преобразование координаты векторов $[A^*]$ и $[B^*]$. Результат преобразования A в A^* и B в B^* показан на рис. 2-2, где x и y — это исходные оси координат, а x^* и y^* — преобразованные оси. Из рисунка видно, что преобразование сдвига $[T]$ увеличивает длину отрезка и изменяет его направление.

2-6 ПРЕОБРАЗОВАНИЕ СРЕДНЕЙ ТОЧКИ

На рис. 2-2 видно, что (2×2) -матрица преобразует прямую линию $y = x + 1$, проходящую между точками A и B в другую прямую $y = \frac{3}{4}x - \frac{5}{4}$, которая проходит между точками A^* и B^* . Фактически с помощью (2×2) -матрицы осуществляется преобразование любой прямой в другую прямую. Все точки преобразованной линии непосредственно соответствуют всем точкам исходной линии. Это достаточно очевидно для конечных точек линии. Рассмотрим теперь преобразование средней точки прямой линии AB . Допустим, что

$$[A] = [x_1 \ y_1], \quad [B] = [x_2 \ y_2] \quad \text{и} \quad [T] = \begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

Преобразуем одновременно две крайние точки:

$$\begin{aligned} \begin{bmatrix} A \\ B \end{bmatrix} [T] &= \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \\ &= \begin{bmatrix} ax_1 + cy_1 & bx_1 + dy_1 \\ ax_2 + cy_2 & bx_2 + dy_2 \end{bmatrix} = \begin{bmatrix} A^* \\ B^* \end{bmatrix}, \end{aligned} \quad (2-11)$$

Итак, конечные точки преобразованной линии A^*B^* имеют следующие координаты

$$\begin{aligned} [A^*] &= [ax_1 + cy_1 \quad bx_1 + dy_1] = [x_1^* \quad y_1^*], \\ [B^*] &= [ax_2 + cy_2 \quad bx_2 + dy_2] = [x_2^* \quad y_2^*]. \end{aligned} \quad (2-12)$$

Средняя точка отрезка A^*B^* выражается через преобразованные конечные точки

$$\begin{aligned} [x_m^* \ y_m^*] &= \left[\frac{x_1^* + x_2^*}{2} \quad \frac{y_1^* + y_2^*}{2} \right] = \\ &= \left[\frac{(ax_1 + cy_1) + (ax_2 + cy_2)}{2} \quad \frac{(bx_1 + dy_1) + (bx_2 + dy_2)}{2} \right] = \\ &= \left[a\frac{(x_1 + x_2)}{2} + c\frac{(y_1 + y_2)}{2} \quad b\frac{(x_1 + x_2)}{2} + d\frac{(y_1 + y_2)}{2} \right]. \end{aligned} \quad (2-13)$$

Возвращаясь к исходной линии AB , можно определить среднюю точку следующим образом:

$$[x_m \ y_m] = [(x_1 + x_2)/2 \quad (y_1 + y_2)/2]. \quad (2-14)$$

Применив матрицу преобразования $[T]$ к средней точке линии AB , получаем:

$$\begin{aligned} [x_m \ y_m] [T] &= \left[\frac{x_1 + x_2}{2} \quad \frac{y_1 + y_2}{2} \right] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \\ &= \left[a \frac{(x_1 + x_2)}{2} + c \frac{(y_1 + y_2)}{2} \quad b \frac{(x_1 + x_2)}{2} + d \frac{(y_1 + y_2)}{2} \right]. \end{aligned} \quad (2-15)$$

Из сравнения выражений (2-13) и (2-15) видно, что они одинаковы, и поэтому средняя точка линии AB преобразуется в среднюю точку линии A^*B^* . Такой метод можно применить и к любым другим отрезкам разделенной линии. Таким образом, при преобразовании путем умножения на матрицу гарантируется соответствие всех точек линии AB и A^*B^* .

Пример 2-1 Средняя точка прямой

Рассмотрим отрезок AB из рис. 2-2. Положение векторов конечных точек такое: $[A] = [0 \ 1]$, $[B] = [2 \ 3]$. Преобразование $[T] = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix}$ осуществляет перемещение вектора на линию A^*B^* :

$$\begin{bmatrix} A \\ B \end{bmatrix} [T] = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 11 & 7 \end{bmatrix} = \begin{bmatrix} A^* \\ B^* \end{bmatrix}.$$

Средняя точка A^*B^* будет иметь координаты

$$[x_m^* \ y_m^*] = \left[\frac{3+11}{2} \quad \frac{1+7}{2} \right] = [7 \ 4].$$

Координаты средней точки линии AB равны

$$[x_m \ y_m] = \left[\frac{0+2}{2} \quad \frac{1+3}{2} \right] = [1 \ 2].$$

Преобразуем среднюю точку и получим

$$[x_m \ y_m] [T] = [1 \ 2] \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} = [7 \ 4] = [x_m^* \ y_m^*],$$

что полностью эквивалентно предыдущему результату.

Применением этих результатов в машинной графике любая прямая может быть преобразована в любую другую прямую путем простого преобразования ее конечных точек и восстановления линии между ними.

2-7 ПРЕОБРАЗОВАНИЕ ПАРАЛЛЕЛЬНЫХ ЛИНИЙ

Результатом преобразования двух параллельных линий с помощью (2×2) -матрицы снова будут две параллельные линии. Это можно увидеть, рассмотрев линию между точками $[A] = [x_1 \ y_1]$, $[B] = [x_2 \ y_2]$ и параллельную ей линию,

проходящую между точками E и F . Покажем, что для этих линий любое преобразование сохраняет параллельность. Так как AB , EF и A^*B^* и E^*F^* параллельны, то угол наклона линий AB и EF определяется следующим образом:

$$m = \frac{y_2 - y_1}{x_2 - x_1}. \quad (2-16)$$

Преобразуем конечные точки AB , воспользовавшись матрицей общего преобразования размером (2×2) :

$$\begin{aligned} \begin{bmatrix} A \\ B \end{bmatrix} [T] &= \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \\ &= \begin{bmatrix} ax_1 + cy_1 & bx_1 + dy_1 \\ ax_2 + cy_2 & bx_2 + dy_2 \end{bmatrix} = \\ &= \begin{bmatrix} x_1^* & y_1^* \\ x_2^* & y_2^* \end{bmatrix} = \begin{bmatrix} A^* \\ B^* \end{bmatrix}. \end{aligned} \quad (2-17)$$

Наклон прямой A^*B^* определяется следующим образом:

$$m^* = \frac{(bx_2 + dy_2) - (bx_1 + dy_1)}{(ax_2 + cy_2) - (ax_1 + cy_1)} = \frac{b(x_2 - x_1) + d(y_2 - y_1)}{a(x_2 - x_1) + c(y_2 - y_1)}$$

или

$$m^* = \frac{b + d \frac{(y_2 - y_1)}{(x_2 - x_1)}}{a + c \frac{(y_2 - y_1)}{(x_2 - x_1)}} = \frac{b + dm}{a + cm}. \quad (2-18)$$

Так как наклон m^* не зависит от x_1 , x_2 , y_1 , y_2 а m , a , b , c и d одинаковы для EF и AB , то m^* одинаково для E^*F^* и A^*B^* . Таким образом, параллельные линии сохраняют параллельность и после преобразования. Это означает, что при преобразовании (2×2) параллелограмм преобразуется в другой параллелограмм. Эти тривиальные выводы демонстрируют большие возможности использования матрицы преобразования для создания графических эффектов.

2-8 ПРЕОБРАЗОВАНИЕ ПЕРЕСЕКАЮЩИХСЯ ПРЯМЫХ

Результатом преобразования с помощью (2×2) -матрицы пары пересекающихся прямых линий также будет пара пересекающихся линий. Проиллюстрируем этот факт на примере двух прямых, изображенных на рис. 2-3 штриховой линией и заданных уравнениями

$$y = m_1 x + b_1,$$

$$y = m_2 x + b_2.$$

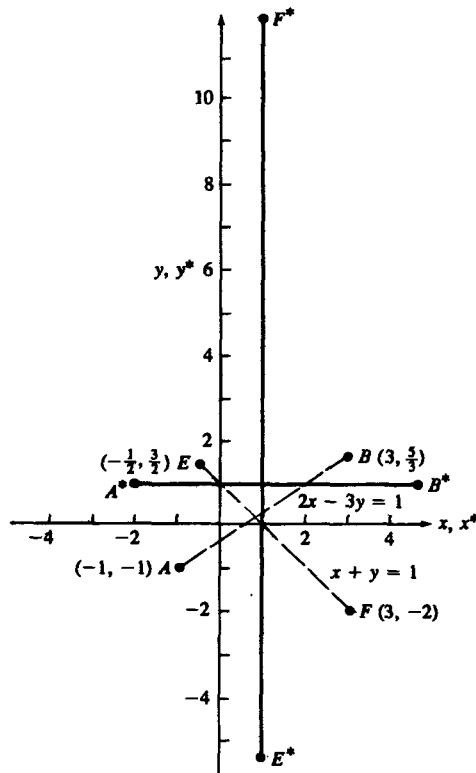


Рис. 2-3 Преобразование пересекающихся прямых.

В матричном представлении эти уравнения будут иметь вид:

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} -m_1 & -m_2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} b_1 & b_2 \end{bmatrix}$$

или

$$[X][M] = [B]. \quad (2-19)$$

Если существует решение этой системы уравнений, то линии пересекаются, в противном случае они параллельны. Решение можно найти путем инверсии матрицы. В частности,

$$[X_i] = [x_i \quad y_i] = [B][M]^{-1}. \quad (2-20)$$

Матрица, обратная $[M]$, имеет следующий вид:

$$[M]^{-1} = \begin{bmatrix} 1 & m_2 \\ \frac{m_2 - m_1}{m_2 - m_1} & \frac{m_2 - m_1}{m_2 - m_1} \\ \frac{-1}{m_2 - m_1} & \frac{-m_1}{m_2 - m_1} \end{bmatrix}, \quad (2-21)$$

так как $[M][M]^{-1} = [I]$, где $[I]$ — единичная матрица. Поэтому координаты точки пересечения двух линий можно найти следующим образом:

$$\begin{aligned}[X_i] &= [x_i \ y_i] = [b_1 \ b_2] \begin{bmatrix} \frac{1}{m_2 - m_1} & \frac{m_2}{m_2 - m_1} \\ \frac{m_2 - m_1}{m_2 - m_1} & \frac{m_2 - m_1}{m_2 - m_1} \end{bmatrix}, \\ [X_i] &= [x_i \ y_i] = \left[\frac{b_1 - b_2}{m_2 - m_1} \quad \frac{b_1 m_2 - b_2 m_1}{m_2 - m_1} \right],\end{aligned}\quad (2-22)$$

Если обе линии преобразовать с помощью (2×2) -матрицы общего преобразования вида

$$[T] = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

то их уравнения будут иметь вид

$$\begin{aligned}y^* &= m_1^* x^* + b_1^*, \\y^* &= m_2^* x^* + b_2^*.\end{aligned}$$

Соответственно можно показать, что

$$m_i^* = \frac{b + dm_i}{a + cm_i} \quad (2-23)$$

и

$$b_i^* = b_i(d - cm_i^*) = b_i \frac{ad - bc}{a + cm_i}, \quad \text{где } i = 1, 2. \quad (2-24)$$

Точка пересечения линий после преобразования отыскивается таким же образом, как и в случае исходных линий:

$$\begin{aligned}[X_i^*] &= [x_i^* \ y_i^*] = \\ &= \left[\frac{b_1^* - b_2^*}{m_2^* - m_1^*} \quad \frac{b_1^* m_2^* - b_2^* m_1^*}{m_2^* - m_1^*} \right].\end{aligned}$$

Воспользовавшись выражениями (2-23) и (2-24), получим

$$\begin{aligned}[X_i^*] &= [x_i^* \ y_i^*] = \\ &= \left[\frac{a(b_1 - b_2) + c(b_1 m_2 - b_2 m_1)}{m_2 - m_1} \quad \frac{b(b_1 - b_2) + d(b_1 m_2 - b_2 m_1)}{m_2 - m_1} \right].\end{aligned}\quad (2-25)$$

Возвращаясь теперь к точке пересечения $[x_i \ y_i]$ исходных линий и применяя уже полученную матрицу преобразования, имеем

$$\begin{aligned}[x_i^* \ y_i^*] &= [x_i \ y_i] [T] = \\ &= \left[\frac{b_1 - b_2}{m_2 - m_1} \quad \frac{b_1 m_2 - b_2 m_1}{m_2 - m_1} \right] \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \\ &= \left[\frac{a(b_1 - b_2) + c(b_1 m_2 - b_2 m_1)}{m_2 - m_1} \quad \frac{b(b_1 - b_2) + d(b_1 m_2 - b_2 m_1)}{m_2 - m_1} \right].\end{aligned}\quad (2-26)$$

Сравнение уравнений (2-25) и (2-26) показывает, что они одинаковы. Итак, точка пересечения преобразуется точно в другую точку пересечения.

Пример 2-2 Пересекающиеся прямые

Рассмотрим две штриховые линии AB и EF на рис. 2-3, конечные точки которых имеют координаты

$$[A] = [-1 \quad -1], \quad [B] = [3 \quad 5/3]$$

и

$$[E] = [-1/2 \quad 3/2], \quad [F] = [3 \quad -2].$$

Уравнение прямой AB имеет вид $-(2/3)x + y = -1/3$, а прямая EF задается уравнением $x + y = 1$. В матричном виде пучок прямых представляется в виде

$$[x \quad y] \begin{bmatrix} -2/3 & 1 \\ 1 & 1 \end{bmatrix} = [-1/3 \quad 1].$$

Используя матрицу обратного преобразования (2-21), получим точку пересечения этих прямых

$$[x_i \quad y_i] = [-1/3 \quad 1] \begin{bmatrix} -3/5 & -3/5 \\ 3/5 & 2/5 \end{bmatrix} = [4/5 \quad 1/5].$$

Теперь преобразуем эти линии с помощью матрицы

$$[T] = \begin{bmatrix} 1 & 2 \\ 1 & -3 \end{bmatrix}.$$

Результирующие прямые A^*B^* и E^*F^* показаны на рис. 2-3. В матричном виде уравнения преобразованных линий имеют вид

$$[x^* \quad y^*] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [1 \quad 1]$$

с точкой пересечения $[x_i^* \quad y_i^*] = [1 \quad 1]$.

Преобразуя точку пересечения исходных линий, получим

$$[x_i^* \quad y_i^*] = [x_i \quad y_i] [T] = [4/5 \quad 1/5] \begin{bmatrix} 1 & 2 \\ 1 & -3 \end{bmatrix} = [1 \quad 1],$$

что тождественно точке пересечения преобразованных линий.

Из рис. 2-3 и примера 2-2 видно, что исходные штриховые прямые AB и EF не перпендикулярны друг другу. Однако преобразованные прямые A^*B^* и E^*F^* , показанные сплошной линией, являются перпендикулярными. Таким образом, преобразование $[T]$ переводит две пересекающиеся неперпендикулярные прямые в две пересекающиеся перпендикулярные. Смысл обратного преобразования $[T]^{-1}$

состоит в переводе двух пересекающихся перпендикулярных прямых в две пересекающиеся, но не перпендикулярные, что может привести к неприятным геометрическим последствиям. Значительный интерес представляет вопрос, при каком условии перпендикулярные прямые преобразуются в перпендикулярные. Мы вернемся к этому вопросу в разд. 2-14, где разберем его подробнее.

Дополнительное изучение рис. 2-3 и примера 2-2 показывает, что преобразование $[T]$ включает в себя поворот, отражение и масштабирование. Рассмотрим каждое из этих преобразований отдельно.

2-9 ПОВОРОТ

Рассмотрим треугольник ABC (рис. 2-4) и с помощью следующего преобразования повернем его на 90° против часовой стрелки относительно начала координат

$$[T] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Если использовать матрицу размером (3×2) , состоящую из координат x и y вершин треугольника, то можно записать

$$\begin{bmatrix} 3 & -1 \\ 4 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ -1 & 4 \\ -1 & 2 \end{bmatrix},$$

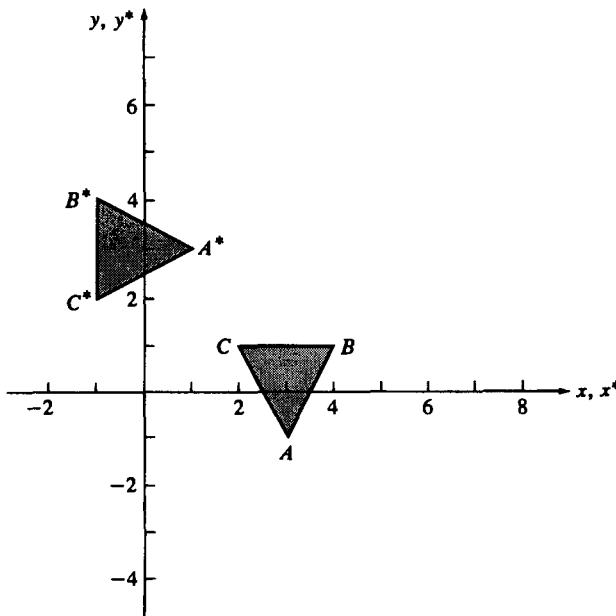


Рис. 2-4 Поворот.

что является координатами результирующего треугольника $A^*B^*C^*$. Поворот на 180° относительно начала координат достигается путем следующего преобразования

$$[T] = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix},$$

а на 270° относительно начала координат — преобразованием

$$[T] = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Разумеется, что матрица тождественного преобразования

$$[T] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

соответствует повороту вокруг начала координат на 0° или 360° . Обратим внимание, что в этих примерах не встречаются ни масштабирование, ни отражение.

В этих примерах осуществляется преобразование в специальных случаях поворота вокруг начала координат на углы 0° , 90° , 180° и 270° . Как осуществить поворот вокруг точки начала координат на произвольный угол θ ? Для ответа на этот вопрос рассмотрим вектор положения от начала координат до точки P (рис. 2-5). Обозначим r — длину вектора, а ϕ — угол между вектором и осью x . Вектор положения поворачивается вокруг начала координат на угол θ и попадает в точку P^* . Записав векторы положений для P и P^* , получаем:

$$P = [x \ y] = [r \cos \phi \ r \sin \phi]$$

и

$$P^* = [x^* \ y^*] = [r \cos(\phi + \theta) \ r \sin(\phi + \theta)].$$

Используя формулу для \cos суммы углов, перепишем выражение для P^* следующим образом

$$P^* = [x^* \ y^*] = [r(\cos \phi \cos \theta - \sin \phi \sin \theta) \ r(\cos \phi \sin \theta + \sin \phi \cos \theta)].$$

Используя определения x и y , можно переписать P^* как

$$P^* = [x^* \ y^*] = [x \cos \theta - y \sin \theta \ x \sin \theta + y \cos \theta].$$

Таким образом, преобразованная точка имеет координаты

$$x^* = x \cos \theta - y \sin \theta, \tag{2-27a}$$

$$y^* = x \sin \theta + y \cos \theta. \tag{2-27b}$$

или в матричном виде

$$\begin{aligned} [X^*] &= [X][T] = [x^* \ y^*] = \\ &= [x \ y] \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}. \end{aligned} \tag{2-28}$$

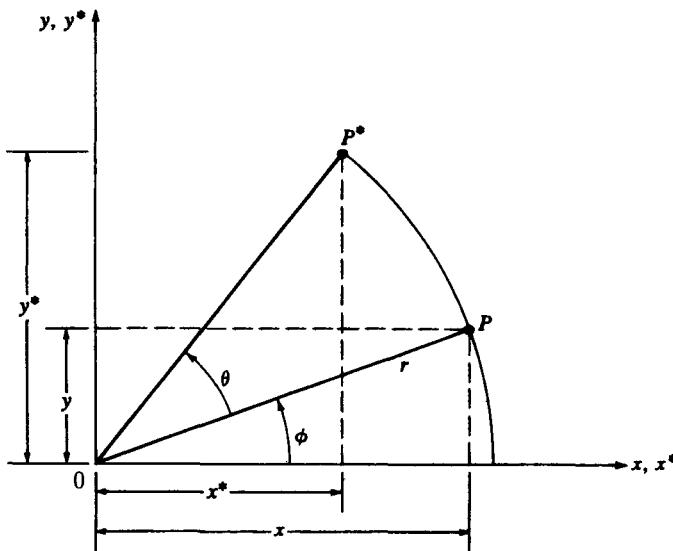


Рис. 2-5 Поворот координатного вектора.

Итак, преобразование поворота вокруг точки начала координат на произвольный угол θ задается матрицей

$$[T] = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}. \quad (2-29)$$

Повороты являются положительными, если они осуществляются против часовой стрелки относительно точки вращения (рис. 2-5).

Определитель общей матрицы поворота имеет следующий вид:

$$\det[T] = \cos^2 \theta + \sin^2 \theta = 1. \quad (2-30)$$

В общем случае преобразования по матрице с детерминантом, равным 1, приводят к полному повороту.

Предположим теперь, что требуется возвратить точку P^* обратно в P , т. е. выполнить обратное преобразование. Очевидно, что требуемый угол поворота равен $-\theta$. Из формулы (2-29) возьмем матрицу для выполнения необходимого преобразования

$$[T]^{-1} = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (2-31)$$

так как $\cos(-\theta) = \cos \theta$ и $\sin(-\theta) = -\sin \theta$. Выражение $[T]^{-1}$ является формальной записью обратной матрицы $[T]$. Можно показать, что матрица $[T]^{-1}$ является

обратной к $[T]$, если вспомнить, что результат умножения матрицы на обратную дает единичную матрицу. В нашем случае:

$$\begin{aligned}[T][T]^{-1} &= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \\ &= \begin{bmatrix} \cos^2 \theta + \sin^2 \theta & -\cos \theta \sin \theta + \cos \theta \sin \theta \\ -\cos \theta \sin \theta + \cos \theta \sin \theta & \cos^2 \theta + \sin^2 \theta \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = [I],\end{aligned}$$

где $[I]$ — единичная матрица.

Анализ выражений (2-29) и (2-31) приводит к другому интересному и полезному результату. Вспомним, что транспонирование матрицы определяется заменой ее строк столбцами. Обозначим транспонированную матрицу $[T]$ как $[T]^T$. Сравнивая ее с $[T]^{-1}$, видим, что

$$[T]^T = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = [T]^{-1}. \quad (2-32)$$

Обратная матрица вращения является транспонированной. Поскольку формально определитель обратной матрицы вычисляется гораздо сложнее, чем определитель транспонированной, то выражение (2-32) является достаточно важным и полезным результатом. В общем случае обратной для любой матрицы преобразования полного поворота, т. е. матрицы с определителем, равным +1, является ее транспонированная матрица (такие матрицы называют ортогональными).

2-10 ОТРАЖЕНИЕ

В то время как полный поворот на плоскости xy обычно осуществляется в двумерном пространстве относительно нормали к плоскости, отражение представляет собой тот же поворот на угол 180° в трехмерном пространстве и обратно на плоскость относительно оси, лежащей в плоскости xy . На рис. 2-6 приведены примеры двух отражений на плоскости треугольника DEF . Отражение относительно прямой $y = 0$ (ось x) получено с использованием матрицы

$$[T] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (2-33)$$

В этом случае новые вершины треугольника $D^*E^*F^*$ будут определяться преобразованием

$$\begin{bmatrix} 8 & 1 \\ 7 & 3 \\ 6 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 8 & -1 \\ 7 & -3 \\ 6 & -2 \end{bmatrix}.$$

Подобным образом отражение относительно оси $y = 0$ при $x = 0$ будет иметь вид

$$[T] = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (2-34)$$

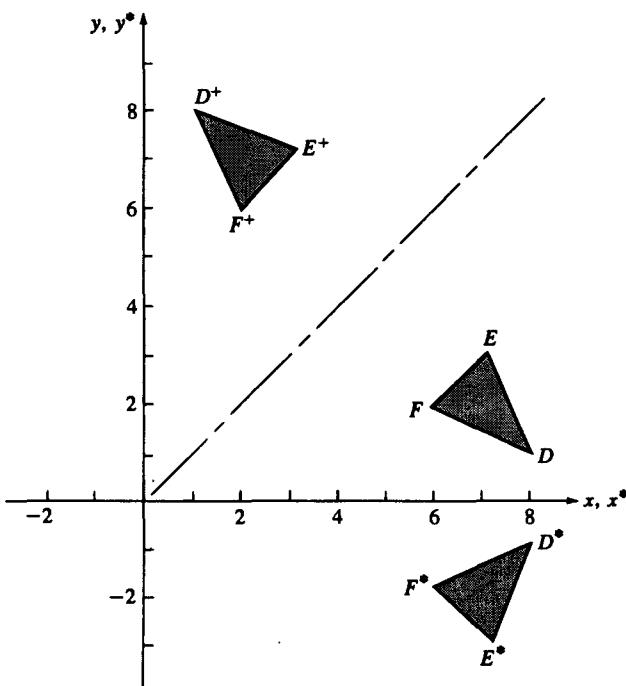


Рис. 2-6 Отражение.

Отражение относительно прямой $y = x$ осуществляется с помощью матрицы

$$[T] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (2-35)$$

Выполнив преобразования, получим координаты вершин треугольника $D^*E^*F^*$

$$\begin{bmatrix} 8 & 1 \\ 7 & 3 \\ 6 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 8 \\ 3 & 7 \\ 2 & 6 \end{bmatrix}.$$

Аналогичным образом отражение относительно оси x будет иметь вид

$$[T] = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}. \quad (2-36)$$

У каждой из этих матриц определитель равен -1 . В общем случае, если определитель матрицы преобразования равен -1 , то преобразование дает полное отражение.

Если оба полных отражения осуществляются последовательно относительно прямых, проходящих через начало координат, то результатом будет полный поворот относительно начала координат. Это можно увидеть, обратившись к следующему примеру.

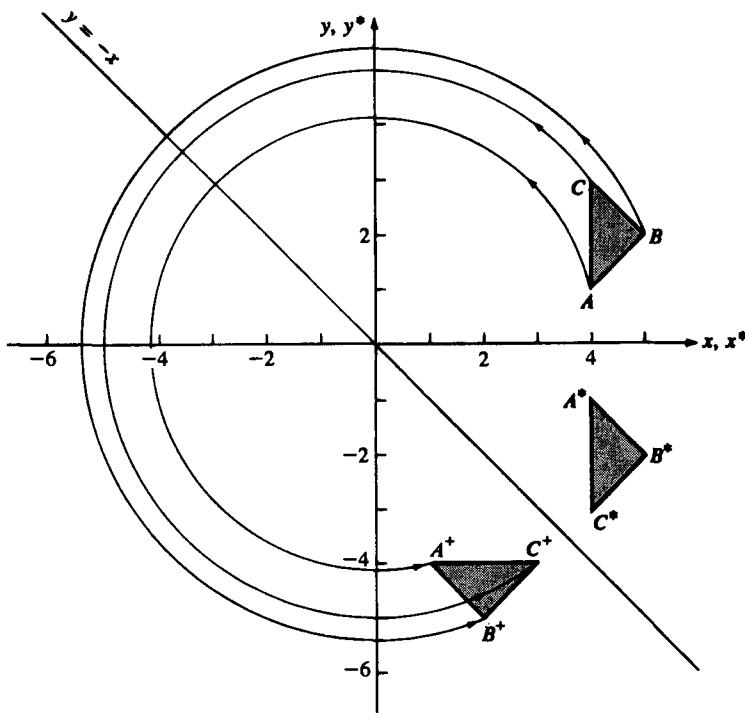


Рис. 2-7 Выполнение отражения путем поворота.

Пример 2-3 Отражение и вращение

Рассмотрим треугольник ABC , показанный на рис. 2-7. Первоначально отобразим его относительно оси x (уравнение 2-33), а затем относительно прямой $y = -x$ (см. выражение (2-36)). Результатом первого отображения будет

$$[X^*] = [X][T_1] = \begin{bmatrix} 4 & 1 \\ 5 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 4 & -1 \\ 5 & -2 \\ 4 & -3 \end{bmatrix}.$$

Результатом второго будет

$$[X^+] = [X^*][T_2] = \begin{bmatrix} 4 & -1 \\ 5 & -2 \\ 4 & -3 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -4 \\ 2 & -5 \\ 3 & -4 \end{bmatrix}.$$

Повернем треугольник относительно начала координат на угол $\theta = 270^\circ$ (см. (2-29)) и получим аналогичный результат

$$[X^+] = [X][T_3] = \begin{bmatrix} 4 & 1 \\ 5 & 2 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & -4 \\ 2 & -5 \\ 3 & -4 \end{bmatrix}.$$

Отметим, что матрицы отражения из (2-33) и (2-36) ортогональны, т. е. транспонированная матрица одновременно является обратной. Например,

$$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}^T = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}^{-1}.$$

2-11 МАСШТАБИРОВАНИЕ

Из наших рассуждений относительно преобразования точек следует, что величина масштабирования определяется значением элементов исходной диагональной матрицы. Если матрица

$$[T] = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

используется в качестве оператора воздействия на вершины треугольника, то имеет место «двукратное» расширение или равномерное масштабирование относительно точки начала координат. Если значения элементов не равны, то треугольник искажается, что проиллюстрировано на рис. 2-8. Треугольник ABC , преобразованный с помощью матрицы

$$[T] = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

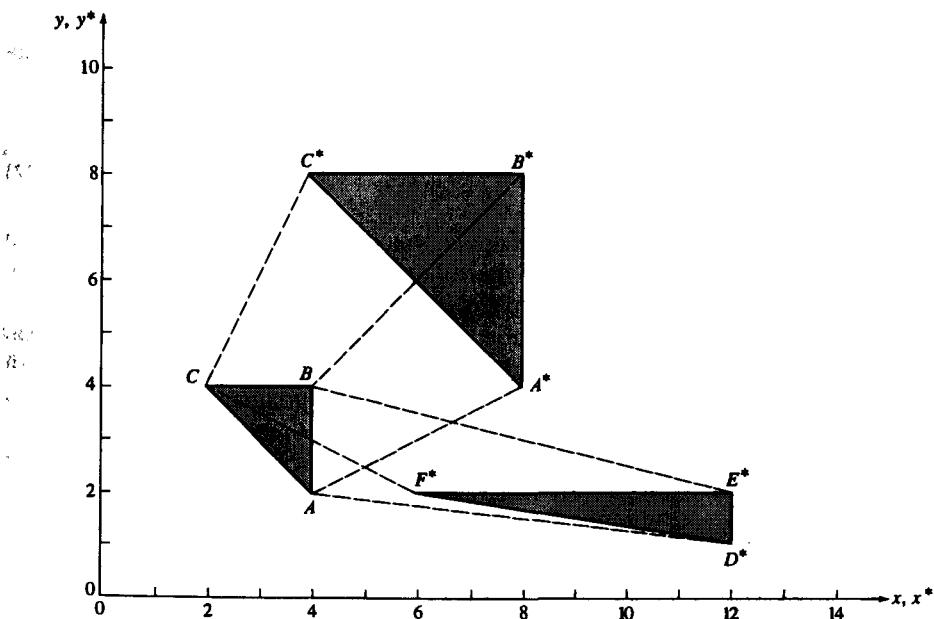


Рис. 2-8 Пропорциональное и непропорциональное масштабирование (искажение).

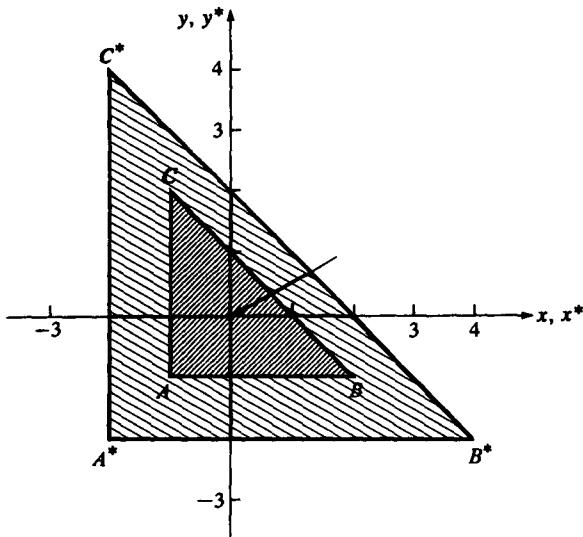


Рис. 2-9 Пропорциональное масштабирование без явного перемещения.

переходит в пропорционально увеличенный треугольник $A^*B^*C^*$. Тот же треугольник, но преобразованный с помощью матрицы

$$\begin{bmatrix} 1/2 & 0 \\ 0 & 3 \end{bmatrix}$$

переходит в треугольник $D^*E^*F^*$, имеющий искажение, вызванное разными коэффициентами масштабирования.

В общем случае при матрице

$$[T] = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad (2-37)$$

в которой $a = d, b = c = 0$, вьшолняется пропорциональное масштабирование; если $a \neq d, b = c = 0$, то масштабирование будет проведено непропорционально. В первом случае для $a = d > 1$ происходит расширение, т. е. увеличение изображения. Если $a = d < 1$, то происходит равномерное сжатие, т. е. фигура уменьшается. Непропорциональное расширение и сжатие возникают в зависимости от значений a и d , которые могут быть меньше либо больше, чем 1, независимо друг от друга.

Из рис. 2-8 видно также, что на первый взгляд преобразование треугольника является перемещением. Это объясняется тем, что относительно начала координат масштабируются координатные векторы, а не точки.

Для того чтобы лучше понять этот факт, рассмотрим преобразования ABC в $D^*E^*F^*$ более внимательно. В частности,

$$[X^*] = [X][T] = \begin{bmatrix} 4 & 2 \\ 4 & 4 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 1/2 & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 6 \\ 2 & 12 \\ 1 & 12 \end{bmatrix}.$$

Заметим, что каждая из компонент x координатных векторов треугольника DEF умножалась на масштабный коэффициент 3, а компоненты y — на 2.

Для того чтобы получить чистое масштабирование без эффекта перемещения, центр фигуры надо поместить в начало координат. Это видно из рис. 2-9, на котором треугольник ABC увеличивается в два раза при масштабировании относительно его центра с координатами, равными $1/3$ основания и $1/3$ высоты. Конкретная матрица преобразования имеет вид

$$[X^*] = [X][T] = \begin{bmatrix} -1 & -1 \\ 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -2 \\ 4 & -2 \\ -2 & 4 \end{bmatrix}.$$

2-12 КОМБИНИРОВАННЫЕ ПРЕОБРАЗОВАНИЯ

Возможности матричного метода достаточно ясно описаны в предыдущих разделах книги. С помощью матричных операций над координатными векторами, определяющими вершины фигур, можно управлять формой и положением поверхности. Однако для получения желаемой ориентации может потребоваться более одного преобразования. Так как операция умножения матриц не коммутативна, то важен порядок выполнения преобразования.

Для иллюстрации эффекта некоммутативности операции умножения матриц рассмотрим преобразования поворота и отражения координатного вектора $[x \ y]$. Если вслед за поворотом на 90° (посредством $[T_1]$) производится отражение относительно прямой $y = -x$ (посредством $[T_2]$), то эти два последовательных преобразования дают

$$[X'] = [X][T_1] = [x \ y] \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = [-y \ x]$$

и затем

$$[X^*] = [X'][T_2] = [-y \ -x] \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = [-x \ y].$$

С другой стороны, если отражение следует за поворотом, то получатся следующие результаты:

$$[X'] = [X][T_2] = [x \ y] \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = [-y \ -x]$$

и

$$[X^*] = [X'][T_1] = [-y \ -x] \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = [x \ -y].$$

Оба результата различны, что подтверждает важность порядка применения матричных преобразований.

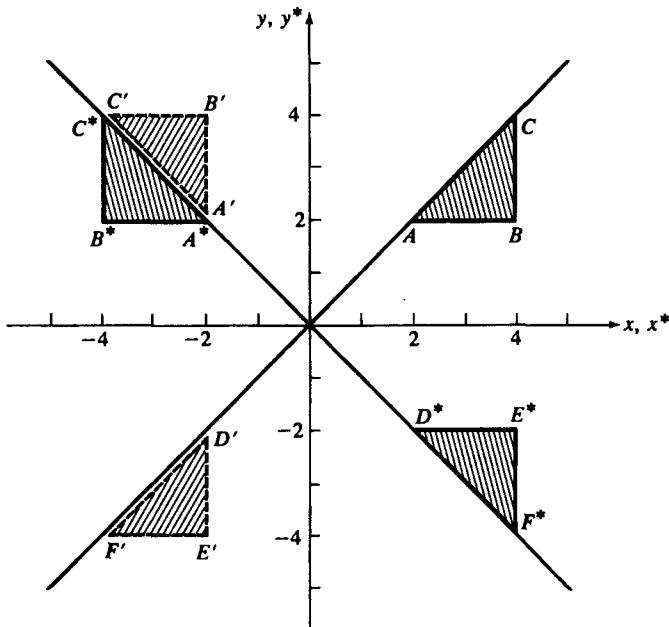


Рис. 2-10 Комбинированное преобразование на плоскости.

Другое принципиальное обстоятельство иллюстрируется этими результатами и приведенным ниже примером. Ранее отдельные матричные преобразования применялись последовательно к результатам предыдущих преобразований, например

$$\begin{bmatrix} x & y \end{bmatrix} [T_1] \rightarrow \begin{bmatrix} x' & y' \end{bmatrix}$$

и

$$\begin{bmatrix} x' & y' \end{bmatrix} [T_2] \rightarrow \begin{bmatrix} x^* & y^* \end{bmatrix}.$$

В приведенном ниже примере отдельные преобразования предварительно комбинируются или конкатенируются, а затем полученная матрица применяется к исходному вектору, т. е. $[T_1][T_2] \rightarrow [T_3]$ и $\begin{bmatrix} x & y \end{bmatrix} [T_3] \rightarrow \begin{bmatrix} x^* & y^* \end{bmatrix}$.

Пример 2-4 Комбинированные преобразования на плоскости

Рассмотрим треугольник ABC на рис. 2-10. Выполним над ним два преобразования: поворот на $+90^\circ$ вокруг точки начала координат

$$[T_1] = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

и отражение относительно линии $y = -x$

$$[T_2] = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}.$$

Результатом воздействия комбинированного преобразования $[T_3] = [T_1][T_2]$ на треугольник ABC является

$$[X^*] = [X][T_1][T_2] = [X][T_3]$$

или

$$\begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 2 \\ -4 & 2 \\ -4 & 4 \end{bmatrix}.$$

Получившийся треугольник $A^*B^*C^*$ является конечным результатом данного преобразования, а треугольник $A'B'C'$ — промежуточным результатом (рис. 2-10).

Проведем преобразование в обратном порядке

$$[X^*] = [X][T_2][T_1] = [X][T_4]$$

или

$$\begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ 4 & -2 \\ 4 & -4 \end{bmatrix}.$$

Конечным результатом будет треугольник $D^*E^*F^*$, а промежуточным $D'E'F'$ (рис. 2-10). Оба результата различны, тем самым снова подтверждается важность порядка применения преобразований. Отметим также, что для определителей справедливы равенства $\det[T_3] = -1$, и $\det[T_4] = -1$ и поэтому оба результата могут быть получены с помощью единственного отражения. Треугольник $A^*B^*C^*$ можно получить из ABC путем отражения относительно оси Y (матрица $[T_3]$ и уравнение (2-34), $D^*E^*F^*$ получается из ABC при отражении относительно оси X (матрица $[T_4]$ и уравнение (2-33)).

2-13 ПРЕОБРАЗОВАНИЕ ЕДИНИЧНОГО КВАДРАТА

До сих пор мы рассматривали поведение точек и линий для определения результатов простых матричных преобразований. Однако можно корректно рассматривать применение матрицы к любой точке плоскости. Как было показано ранее, единственная точка, остающаяся инвариантной при воздействии матричных преобразований, — это точка начала координат. Все другие точки плоскости подвержены преобразованию, которое можно представить как растяжение исходной плоскости, системы координат и перевод в новую форму. Формально принято считать, что преобразование вызывает переход от одного координатного пространства к другому.

Рассмотрим координатную сетку, состоящую из единичных квадратов на координатной плоскости xy (рис. 2-11). Четыре координатных вектора вершин единичного квадрата, проходящие под одним углом к началу координат, имеют следующий вид:

$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$	начало координат — A
	единичная точка B на оси x
	внешний угол C
	единичная точка D на оси y

Такой идентичный квадрат изображен на рис. 2-11a. Применяя к нему (2×2) -матрицу общего преобразования, получаем

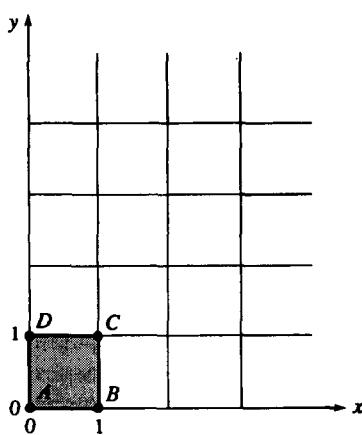
$$\begin{array}{l} A \\ B \\ C \\ D \end{array} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} \cdot 0 & 0 \\ a & b \\ a+c & b+d \\ c & d \end{bmatrix} \begin{array}{l} A^* \\ B^* \\ C^* \\ D^* \end{array} \quad (2-38)$$

Результаты этого преобразования показаны на рис. 2-11b. Из выражения (2-38) следует, что начало координат не подвергается преобразованию, т. е. $[A] = [A^*] = [0 \ 0]$. Далее отметим, что координаты B^* равны первой строке матрицы преобразования, а координаты D^* — второй. Таким образом, матрица преобразования является определенной, если определены координаты B^* и D^* (преобразование единичных векторов $[1 \ 0]$, $[0 \ 1]$). Поскольку стороны единичного квадрата первоначально параллельны и ранее было показано, что параллельные линии преобразуются снова в параллельные, то результирующая фигура является параллелограммом.

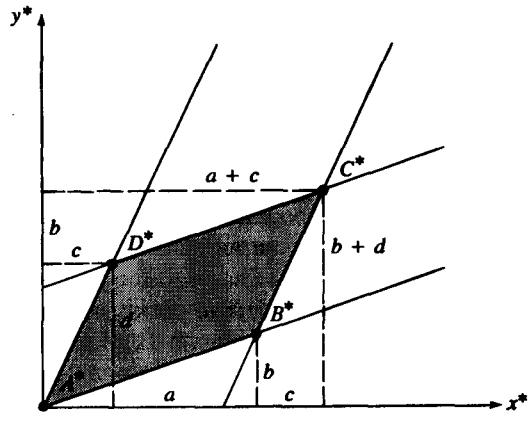
Влияние элементов a , b , c и d матрицы 2×2 может быть установлено отдельно. Элементы b и c , как видно из рис. 2-11b, вызывают сдвиг (см. разд. 2-4) исходного квадрата в направлениях y и x соответственно. Как отмечалось ранее, элементы a и d играют роль масштабных множителей. Таким образом, 2×2 -матрица задает комбинацию сдвига и масштабирования.

Несложно определить также площадь параллелограмма $A^*B^*C^*D^*$ из рис. 2-11b, которую можно вычислить следующим образом:

$$A_p = (a + c)(b + d) - \frac{1}{2}(ab) - \frac{1}{2}(cd) - \frac{c}{2}(b + b + d) - \frac{b}{2}(c + a + c).$$



(a)



(b)

Рис. 2-11 Общее преобразование единичного квадрата: a) до преобразования; b) после преобразования.

В результате получаем

$$A_p = ad - bc = \det \begin{bmatrix} a & b \\ c & d \end{bmatrix}. \quad (2-39)$$

Можно показать, что площадь любого параллелограмма A_p , образованного путем преобразования квадрата, есть функция от определителя матрицы преобразования и связана с площадью исходного квадрата A_s простым отношением

$$A_p = A_s(ad - bc) = A_s \det [T]. \quad (2-40)$$

Фактически, так как площадь всей фигуры равна сумме площадей единичных квадратов, то площадь любой преобразованной фигуры A_t зависит от площади исходной фигуры A_i

$$A_t = A_i(ad - bc). \quad (2-41)$$

Это полезный способ определения площадей произвольных фигур.

Пример 2-5 Масштабирование области

Треугольник ABC с координатными векторами $[1 \ 0]$, $[0 \ 1]$ и $[-1 \ 0]$ преобразуется матрицей

$$[T] = \begin{bmatrix} 3 & 2 \\ -1 & 2 \end{bmatrix},$$

образуя новый треугольник $A^*B^*C^*$ (рис. 2-12).

Площадь треугольника ABC равна

$$A_i = \frac{1}{2} (\text{основание}) (\text{высота}) = \frac{1}{2}(2)(1) = 1.$$

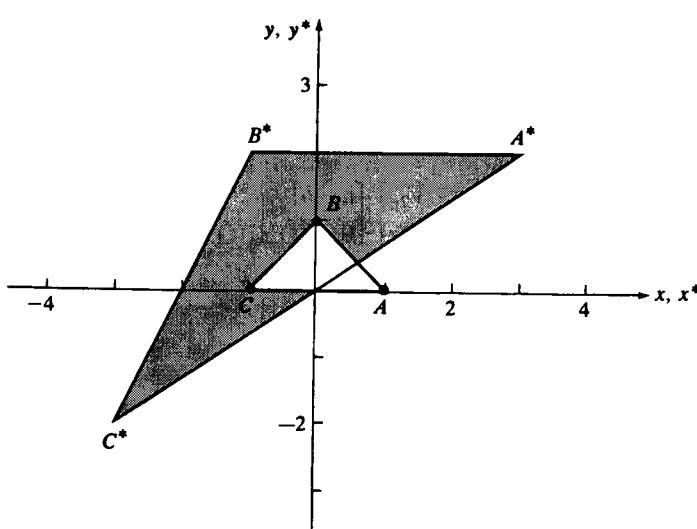


Рис. 2-12 Масштабирование области.

Воспользуемся уравнением (2-41), тогда площадь преобразованного треугольника $A^*B^*C^*$ будет равна

$$A_t = A_i(ad - bc) = 1(6 + 2) = 8.$$

Векторы преобразованного треугольника $A^*B^*C^*$ теперь равны

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ -1 & 2 \\ -3 & -2 \end{bmatrix}.$$

Вычислим площадь, образованную результирующими вершинами:

$$A_t = \frac{1}{2} (\text{основание}) (\text{высота}) = \frac{1}{2}(4)(4) = 8.$$

Это совпадает с полученным ранее результатом.

2-14 ПРЕОБРАЗОВАНИЯ ЖЕСТКИХ КОНСТРУКЦИЙ

Пришло время вернуться к поставленному в разд. 2-8 вопросу: когда перпендикулярные прямые преобразуются в перпендикулярные прямые? Рассмотрим сначала более общий вопрос: в каких случаях угол между пересекающимися прямыми сохраняется?

Напомним, что скалярное произведение двух векторов равно

$$\bar{V}_1 \cdot \bar{V}_2 = V_{1x}V_{2x} + V_{1y}V_{2y} = |\bar{V}_1| |\bar{V}_2| \cos \theta, \quad (2-42)$$

а векторное произведение двух векторов, принадлежащих плоскости xy , определяется как

$$\bar{V}_1 \times \bar{V}_2 = (V_{1x}V_{2y} - V_{2x}V_{1y})\bar{k} = |\bar{V}_1| |\bar{V}_2| \bar{k} \sin \theta, \quad (2-43)$$

где индексы x и y относятся к компонентам x и y вектора, θ — острый угол между векторами, а \bar{k} — единичный вектор, перпендикулярный к плоскости xy .

Проведем преобразование \bar{V}_1 и \bar{V}_2 , используя (2×2) -матрицу общего преобразования

$$\begin{bmatrix} \bar{V}_1 \\ \bar{V}_2 \end{bmatrix} [T] = \begin{bmatrix} V_{1x} & V_{2y} \\ V_{2x} & V_{2y} \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} aV_{1x} + cV_{1y} & bV_{1x} + dV_{1y} \\ aV_{2x} + cV_{2y} & bV_{2x} + dV_{2y} \end{bmatrix} = \begin{bmatrix} \bar{V}_1^* \\ \bar{V}_2^* \end{bmatrix}. \quad (2-44)$$

Векторным произведением векторов \bar{V}_1^* и \bar{V}_2^* будет

$$\bar{V}_1^* \times \bar{V}_2^* = (ad - cb)(V_{1x}V_{2y} - V_{2x}V_{1y})\bar{k} = |\bar{V}_1^*| |\bar{V}_2^*| \bar{k} \sin \theta. \quad (2-45)$$

Аналогично, скалярное произведение будет равно

$$\begin{aligned} \bar{V}_1^* \cdot \bar{V}_2^* &= (a^2 + b^2)V_{1x}V_{2x} + (c^2 + d^2)V_{1y}V_{2y} + (ac + bd)(V_{1x}V_{2y} + V_{1y}V_{2x}) = \\ &= |\bar{V}_1^*| |\bar{V}_2^*| \cos \theta. \end{aligned} \quad (2-46)$$

Требуется, чтобы значения векторов, как и угол между ними, оставались постоянными. Сравнивая уравнения (2-42), (2-46) и (2-43), (2-45), а также приравняв коэффициенты подобных членов, получим

$$a^2 + b^2 = 1, \quad (2-47a)$$

$$c^2 + d^2 = 1, \quad (2-47b)$$

$$ac + bd = 0, \quad (2-47c)$$

$$ad - bc = +1. \quad (2-48)$$

Выражения (2-47a, b, c) соответствуют условиям ортогональности матрицы, т. е.

$$[T][T]^{-1} = [T][T]^T = [I]$$

или

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} a^2 + b^2 & ac + bd \\ ac + bd & c^2 + d^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Выражение (2-48) требует, чтобы определитель матрицы преобразования был равен +1.

Таким образом, при полном повороте углы между пересекающимися прямыми сохраняются. Данный результат распространяется также и на операцию отражения, ортогональная матрица которого имеет определитель, равный -1 . В этом случае величины векторов сохраняются, но угол между преобразованными векторами в действительности равен $2\pi - \theta$. (Следовательно, в общем случае угол не сохраняется. Однако перпендикулярные прямые преобразуются в перпендикулярные прямые. Поскольку $\sin(2\pi - \theta) = -\sin\theta$, $ad - bc = -1$, полные повороты и отражения называются преобразованиями жесткой конструкции. Кроме того, несколько минут анализа или экспериментирования приводят к выводу, что равномерное масштабирование также сохраняет неизменным угол между пересекающимися прямыми, но не величину преобразуемых векторов. Поскольку ортогональная матрица сохраняет угол между векторами и их величины, матрица однородного масштабирования не является ортогональной.)

2-15 ПЕРЕМЕЩЕНИЯ И ОДНОРОДНЫЕ КООРДИНАТЫ

В предыдущих разделах был рассмотрен ряд преобразований, совершаемых с помощью (2×2) -матрицы общего преобразования. Среди них поворот, отражение, масштабирование, сдвиг и другие. Ранее отмечалось, что исходная система координат инвариантна по отношению ко всем перечисленным преобразованиям. Однако возникает необходимость изменять положение начала координат, т. е. преобразовывать каждую точку на плоскости. Этого можно достичь путем перемещения точки начала координат или любой другой точки на плоскости

$$x^* = ax + cy + m,$$

$$y^* = bx + dy + n.$$

К сожалению, нельзя ввести константы перемещения m и n в (2×2) -матрицу преобразования, так как это не пространство!

Данное затруднение можно преодолеть, используя однородные координаты. Однородные координаты неоднородного координатного вектора $[x \ y]$ представляют собой тройку $[x' \ y' \ h]$, где $x = x'/h$, $y = y'/h$, а h – некоторое вещественное число. Заметим, что случай $h = 0$ является особым. Всегда существует один набор однородных координат вида $[x \ y \ 1]$. Мы выбрали эту форму, чтобы представить координатный вектор $[x \ y]$ на физической плоскости xy . Все остальные однородные координаты представляются в виде $[hx \ hy \ h]$. Данные координаты не сохраняют однозначности, например, все следующие координаты $[6 \ 4 \ 2]$, $[12 \ 8 \ 4]$, $[3 \ 2 \ 1]$ представляют физическую точку $(3,2)$.

Матрица преобразования для однородных координат имеет размер 3×3 . В частности,

$$[T] = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ m & n & 1 \end{bmatrix}, \quad (2-49)$$

где действие элементов a , b , c и d верхней части (2×2) -матрицы точно соответствует действиям, рассмотренным ранее. Элементы m и n являются коэффициентами перемещения в направлениях x и y соответственно. Полная двумерная матрица преобразования имеет вид

$$[x^* \ y^* \ 1] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = [x + m \ y + n \ 1]. \quad (2-50)$$

Отметим, что каждая точка плоскости и даже начало координат $x = y = 0$ теперь могут быть преобразованы.

2-16 ПОВОРОТ ВОКРУГ ПРОИЗВОЛЬНОЙ ТОЧКИ

Ранее мы рассматривали вращение, совершающееся вокруг начала координат. Однородные координаты предусматривают механизм выполнения поворотов вокруг точек, отличных от начала координат. В общем случае поворот вокруг произвольной точки может быть реализован посредством ее перемещения в начало координат, выполнения требуемого поворота и последующего перемещения результата обратно в исходный центр вращения. Таким образом, поворот вектора $[x \ y \ 1]$ вокруг точки m, n на произвольный угол можно осуществить следующим образом:

$$[x^* \ y^* \ 1] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}. \quad (2-51)$$

Выполняя действия над двумя внутренними матрицами, можно записать

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ \left\{ \begin{array}{l} -m(\cos \theta - 1) \\ +n \sin \theta \end{array} \right\} & \left\{ \begin{array}{l} -n(\cos \theta - 1) \\ -m \sin \theta \end{array} \right\} & 1 \end{bmatrix}. \quad (2-52)$$

Рассмотрим пример, иллюстрирующий данный результат.

Пример 2-6 Поворот относительно произвольной точки

Предположим, что центр объекта лежит в точке [4 3]. Требуется повернуть объект на прямой угол против часовой стрелки вокруг центра. Используя матрицу

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

проведем поворот вокруг начала координат, не совпадающего с центром объекта. Обязательной процедурой преобразования является прежде всего такое перемещение объекта, чтобы желаемый центр вращения оказался в начале координат. Это достигается с помощью следующей матрицы перемещения:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -3 & 1 \end{bmatrix}.$$

Далее применяем матрицу поворота и наконец с помощью матрицы перемещения приведем результаты поворота обратно к первоначальному центру. Вся операция

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -4 & -3 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 4 & 3 & 1 \end{bmatrix}$$

может быть реализована одной матрицей путем простого перемножения отдельных матриц, т. е.

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 7 & -1 & 1 \end{bmatrix}.$$

2-17 ОТРАЖЕНИЕ ОТНОСИТЕЛЬНО ПРОИЗВОЛЬНОЙ ПРЯМОЙ

Ранее в разд. 2-10 обсуждалось отражение относительно прямых, проходящих через начало координат. Иногда требуется выполнить отражение объекта относительно прямой, не проходящей через точку начала координат. Это можно сделать, воспользовавшись процедурой, аналогичной вращению вокруг произвольной точки. Конкретно выполняются следующие действия:

- перемещение линии и объекта таким образом, чтобы линия прошла через начало координат;
- поворот линии и объекта вокруг точки начала координат до совпадения с одной из координатных осей;
- отражение относительно координатной оси;
- обратный поворот вокруг начала координат;
- перемещение в исходное положение.

В матричном виде данное преобразование имеет представление

$$[T] = [T'][R][R'][R]^{-1}[T']^{-1}, \quad (2-53)$$

где T' — матрица перемещения, R — матрица поворота вокруг начала координат, R' — матрица отражения.

Перемещения, повороты и отражения также применяются для преобразования произвольных фигур. Рассмотрим следующий пример.

Пример 2-7 Отражение относительно произвольной прямой

Рассмотрим прямую L и треугольник ABC (рис. 2-13a). Уравнение прямой L имеет вид

$$y = \frac{1}{2}(x + 4).$$

Координатные векторы $[2 \ 4 \ 1]$, $[4 \ 6 \ 1]$ и $[2 \ 6 \ 1]$ задают вершины треугольника ABC .

Прямая L пройдет через начало координат при перемещении ее на -2 единицы в направлении оси y . В результате этого при повороте вокруг начала координат на $-\operatorname{tg}^{-1}(1/2) = -26.57^\circ$ прямая совпадет с осью x . Выражение (2-33) используется для отражения треугольника относительно оси x , затем преобразованные координатные векторы треугольника поворачиваются и перемещаются к исходной ориентации. Комбинация преобразований будет иметь вид

$$\begin{aligned} [T] &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 2/\sqrt{5} & -1/\sqrt{5} & 0 \\ 1/\sqrt{5} & 2/\sqrt{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \\ &\quad \times \begin{bmatrix} 2/\sqrt{5} & 1/\sqrt{5} & 0 \\ -1/\sqrt{5} & 2/\sqrt{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}, \\ [T] &= \begin{bmatrix} 3/5 & 4/5 & 0 \\ 4/5 & -3/5 & 0 \\ -8/5 & 16/5 & 1 \end{bmatrix}, \end{aligned}$$

и конкретно для координатных векторов треугольника $A^*B^*C^*$ имеем

$$\begin{bmatrix} 2 & 4 & 1 \\ 4 & 6 & 1 \\ 2 & 6 & 1 \end{bmatrix} \begin{bmatrix} 3/5 & 4/5 & 0 \\ 4/5 & -3/5 & 0 \\ -8/5 & 16/5 & 1 \end{bmatrix} = \begin{bmatrix} 14/5 & 12/5 & 1 \\ 28/5 & 14/5 & 1 \\ 22/5 & 6/5 & 1 \end{bmatrix}$$

(рис. 2-13a). Рис. 2-13 b, c, d, e иллюстрируют различные этапы данного преобразования.

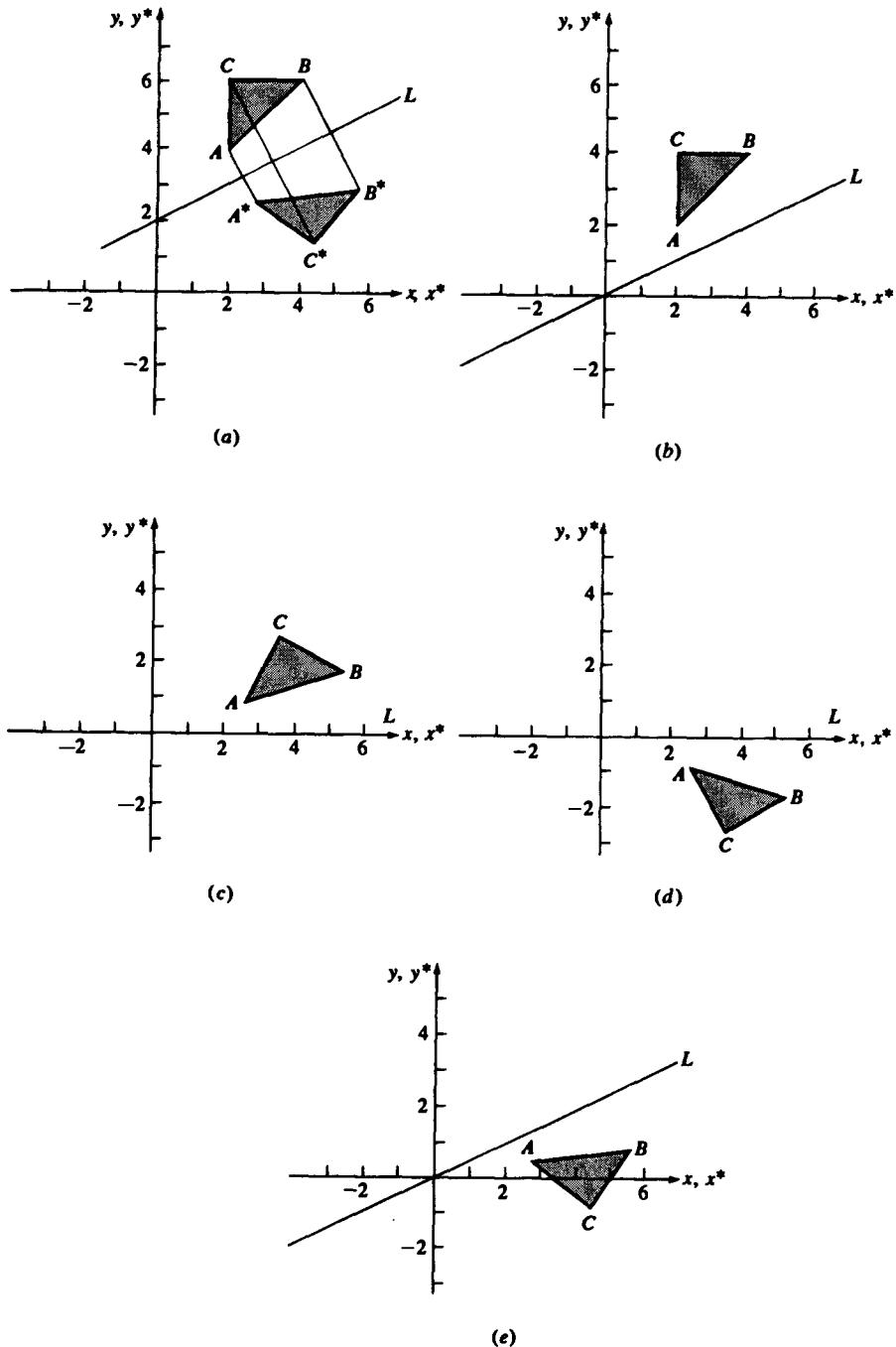


Рис. 2-13 Отражение относительно произвольной прямой: а) исходное и конечное положение; б) перенос прямой в начало координат; в) поворот до совпадения с осью x ; г) отражение относительно оси x ; д) обратный поворот; е) обратный перенос; а) обратный перенос.

2-18 ПРОЕЦИРОВАНИЕ – ГЕОМЕТРИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ ОДНОРОДНЫХ КООРДИНАТ

Матрицу преобразования размером 3×3 для двумерных однородных координат можно разбить на четыре части

$$[T] = \begin{bmatrix} a & b & : & p \\ c & d & : & q \\ \dots & \dots & \dots & \dots \\ m & n & : & s \end{bmatrix}. \quad (2-54)$$

Напомним, что a, b, c и d – коэффициенты масштабирования, вращения, отражения и сдвига соответственно. Элементы m и n задают перемещение. В двух предыдущих разделах коэффициенты имели значения $p = q = 0$ и $s = 1$. Установим величины p и q не равными 0. Какой эффект мы получим? В данном случае полезно рассмотреть геометрическую интерпретацию.

При $p = q = 0$ и $s = 1$ однородные координаты преобразованных векторов всегда равны $h = 1$. Геометрически данный результат интерпретируется как ограничение преобразования физической плоскостью $h = 1$.

Для иллюстрации эффекта преобразования при p и q , отличных от нуля, рассмотрим следующее выражение:

$$\begin{aligned} [X \ Y \ h] &= [hx \ hy \ h] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & p \\ 0 & 1 & q \\ 0 & 0 & 1 \end{bmatrix} = \\ &= [x \ y \ (px + qy + 1)]. \end{aligned} \quad (2-55)$$

Здесь $X = hx$, $Y = hy$ и $h = px + qy + 1$. Преобразованный координатный вектор, выраженный в однородных координатах, лежит теперь в трехмерном пространстве, определенном как $h = px + qy + 1$. Это преобразование показано на рис. 2-14, где отрезок AB , принадлежащий физической плоскости $h = 1$, преобразуется в CD со значением $h \neq 1$, т. е. $pX + qY - h + 1 = 0$.

Однако представляют интерес результаты, принадлежащие физической плоскости с $h = 1$, которые можно получить путем геометрического проецирования прямой CD с плоскости $h \neq 1$ обратно на плоскость $h = 1$ с использованием для этого проецирующих лучей, проходящих через начало координат. Из рис. 2-14, используя правило подобия треугольников, получим

$$x^* = \frac{X}{h} \quad y^* = \frac{Y}{h}$$

или в однородных координатах

$$[x^* \ y^* \ 1] = \left[\frac{Y}{h} \quad \frac{Y}{h} \quad 1 \right].$$

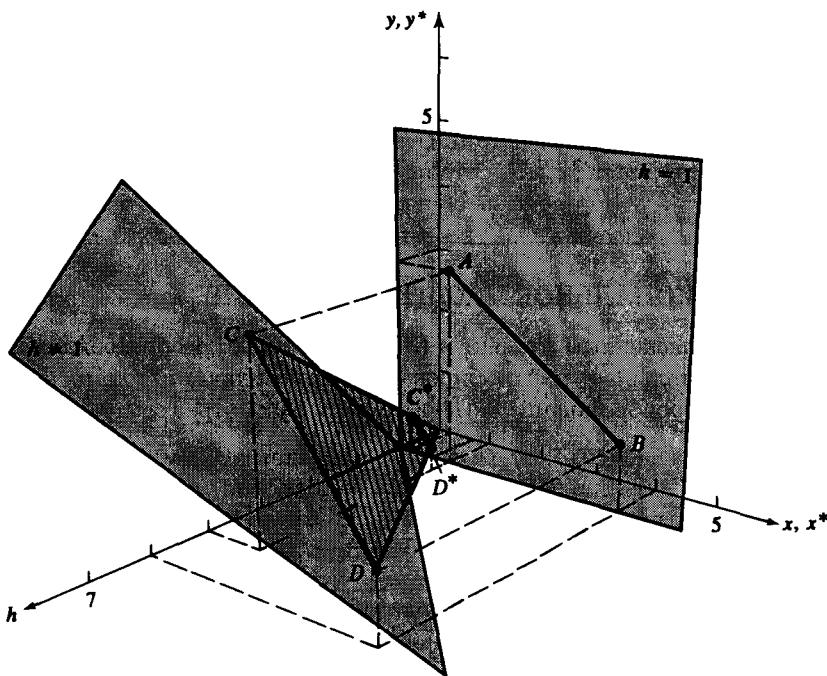


Рис. 2-14 Преобразование из физической плоскости ($h = 1$) на плоскость $h \neq 1$ и проецирование обратно на физическую плоскость.

После этого, нормализуя выражение (2-55) делением однородных координат на величину h , получаем

$$[x^* \quad y^* \quad 1] = \left[\frac{X}{h} \quad \frac{Y}{h} \quad 1 \right] = \left[\frac{x}{px + qy + 1} \quad \frac{y}{px + qy + 1} \quad 1 \right] \quad (2-56)$$

или

$$x^* = \frac{X}{h} = \frac{x}{px + qy + 1}, \quad (2-57a)$$

$$y^* = \frac{Y}{h} = \frac{y}{px + qy + 1}. \quad (2-57b)$$

Детально действие преобразования рассмотрим на следующем примере.

Пример 2-8 Проецирование в однородных координатах

Для отрезка AB из рис. 2-14 имеем $p = q = 1$, $[A] = [1 \quad 3 \quad 1]$ и $[B] = [4 \quad 1 \quad 1]$,

$$\begin{bmatrix} C \\ D \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix} [T] = \begin{bmatrix} 1 & 3 & 1 \\ 4 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 5 \\ 4 & 1 & 6 \end{bmatrix}.$$

Таким образом, $[C] = [1 \ 3 \ 5]$ и $[D] = [4 \ 1 \ 6]$ на плоскости $h=x+y+1$. Проецируя обратно на плоскость $h=1$ путем деления на коэффициент однородных координат, проведем двумерное преобразование точек

$$\begin{aligned}[C^*] &= [1 \ 3 \ 5] = [1/5 \ 3/5 \ 1], \\ [D^*] &= [4 \ 1 \ 6] = [2/3 \ 1/6 \ 1].\end{aligned}$$

Результат показан на рис. 2-14.

2-19 ПРОПОРЦИОНАЛЬНОЕ МАСШТАБИРОВАНИЕ

Оставшийся необъясненным элемент s (3×3)-матрицы преобразования соответствует пропорциональному масштабированию, при котором все компоненты вектора изменяются пропорционально. Покажем это, рассмотрев следующее преобразование:

$$[X \ Y \ h] = [x \ y \ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix} = [x \ y \ s]. \quad (2-58)$$

где $X = x$, $Y = y$ и $h = s$. После нормализации получим $X^* = x/s$ и $Y^* = y/s$. Таким образом, преобразование $[x \ y \ 1][T] = [x/s \ y/s \ 1]$ является равномерным масштабированием координатного вектора. Если $s < 1$, то происходит растяжение, а если $s > 1$ — сжатие.

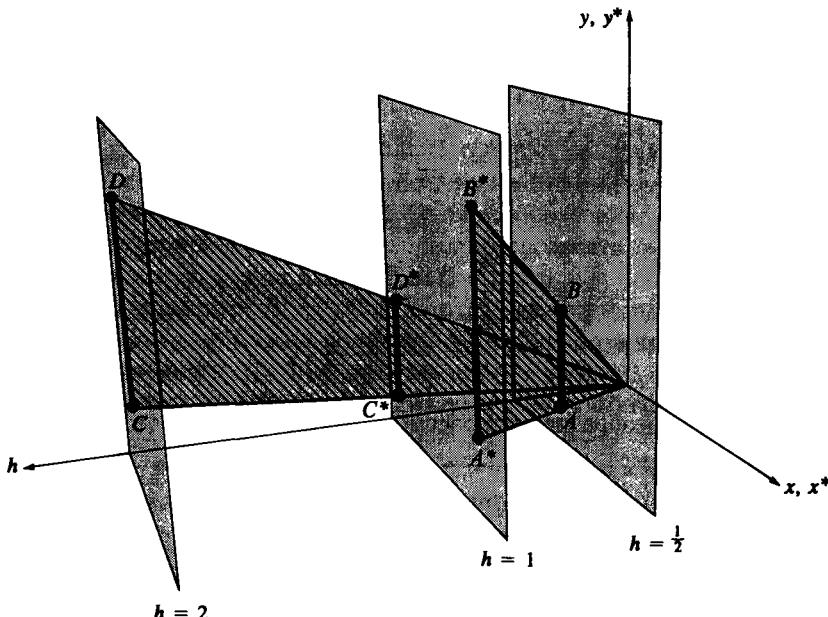


Рис. 2-15 Геометрическая интерпретация пропорционального масштабирования.

Заметим, что это преобразование осуществляется также в плоскости $h = 1$. Здесь $h = s = \text{const}$, и поэтому плоскость $h \neq 1$ параллельна плоскости $h = 1$. Геометрическая интерпретация данного эффекта показана на рис. 2-15. Если $s < 1$, то $h = \text{const}$ задает плоскость, лежащую между плоскостями $h = 1$ и $h = 0$. Следовательно, когда преобразуемая прямая AB проецируется обратно на плоскость $h = 1$, то A^*B^* увеличивается. Аналогично, если $s > 1$, то $h = \text{const}$ определяет плоскость, расположенную за плоскостью $h = 1$ и проходящую вдоль оси h . В случае проецирования прямой CD на плоскость $h = 1$ происходит уменьшение прямой C^*D^* .

2-20 ТОЧКИ БЕСКОНЕЧНОСТИ

Однородные координаты предоставляют удобный и эффективный способ нанесения точек из одной системы координат в соответствующие точки альтернативной координатной системы. Бесконечная область в одной координатной системе часто преобразуется в конечную область в альтернативной системе. При некорректном выборе переноса параллельность прямых может не сохраняться. Однако точки пересечения после преобразования оказываются снова в точках пересечения. Данное свойство используется для определения однородных координат представления точек бесконечности.

Рассмотрим пару пересекающихся прямых, заданных уравнениями

$$\begin{aligned}x + y &= 1, \\2x - 3y &= 0.\end{aligned}$$

Прямые пересекаются в точке с координатами $x = 3/5$, $y = 2/5$. Запишем уравнения в виде $x + y - 1 = 0$, $2x - 3y = 0$ и представим их в матричной форме

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & -3 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

или

$$[X][M'] = [R].$$

Если матрица $[M']$ квадратная, то пересечение может быть получено путем обращения матрицы. Изменим систему исходных уравнений следующим образом:

$$\begin{aligned}x + y - 1 &= 0, \\2x - 3y &= 0, \\1 &= 1,\end{aligned}$$

или в матричной форме

$$[X][M] = [R],$$

т. е.

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 \\ 1 & -3 & 0 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

Квадратная матрица, обратная данной [2-1], имеет следующий вид:

$$[M]^{-1} = \begin{bmatrix} 3/5 & 2/5 & 0 \\ 1/5 & -1/5 & 0 \\ 3/5 & 2/5 & 1 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 3 & 2 & 0 \\ 1 & -1 & 0 \\ 3 & 2 & 5 \end{bmatrix}.$$

Умножая обе части уравнения на $[M]^{-1}$ и учитывая, что $[M][M]^{-1} = [I]$ является тождественной матрицей, получим

$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 2 & 0 \\ 1 & -1 & 0 \\ 3 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 3/5 & 2/5 & 1 \end{bmatrix}.$$

Таким образом, точка пересечения опять имеет координаты $x = 3/5$, $y = 2/5$.

Рассмотрим теперь две параллельные прямые, заданные следующим образом:

$$\begin{aligned} x + y &= 1, \\ x + y &= 0. \end{aligned}$$

По определению геометрии Евклида, точка пересечения двух параллельных прямых расположена в бесконечности. Продолжая предыдущие рассуждения, вычислим точку пересечения этих прямых, заданных в матричной форме,

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

Однако несмотря на то что матрица квадратная, она не имеет обратной, так как две ее строки тождественны. Такая матрица называется сингулярной. Возможна иная формулировка с обратимой матрицей. Получим ее, переписывая систему уравнений следующим образом:

$$\begin{aligned} x + y - 1 &= 0, \\ x + y &= 0, \\ x &= x, \end{aligned}$$

или в матричной форме

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & x \end{bmatrix}.$$

Таблица 2-1 Однородные координаты для точки [4 3]

h	x^*	y^*	X	Y
1	4	3	4	3
1/2	8	6	4	3
1/3	12	9	4	3
:				
1/10	40	30	4	3
:				
1/100	400	300	4	3
:				

В данном случае матрица не является сингулярной и существует обратная ей

$$[M]^{-1} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 1 \\ 1 & -1 & 0 \end{bmatrix}.$$

Умножая обе части выражения на обратную матрицу, получаем

$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & x \end{bmatrix} \begin{bmatrix} 0 & 1 & -1 \\ 0 & 1 & 1 \\ 1 & -1 & u \end{bmatrix} = \begin{bmatrix} x & -x & 0 \end{bmatrix} = x \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}.$$

Результирующие однородные координаты $[1 \ -1 \ 0]$ определяют точку пересечения двух параллельных прямых, т. е. точку бесконечности. В частности, они представляют данную точку в направлении $[1 \ -1]$ двумерного пространства. В общем виде двумерный координатный вектор $[a \ b \ 0]$ представляет точку бесконечности на прямой $ay - bx = 0$. Приведем несколько примеров:

- $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$ точка на положительной оси x ,
- $\begin{bmatrix} -1 & 0 & 0 \end{bmatrix}$ точка на отрицательной оси x ,
- $\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$ точка на положительной оси y ,
- $\begin{bmatrix} 0 & -1 & 0 \end{bmatrix}$ точка на отрицательной оси y ,
- $\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$ вдоль прямой $y = x$ в направлении $[1 \ 1]$.

Вектор с однородной компонентой $h = 0$ действительно представляет точку бесконечности и может быть также интерпретирован как движение к пределу (табл. 2-1).

Рассмотрим прямую $y^* = (3/4)x^*$ и точку $[X \ Y \ h] = [4 \ 3 \ 1]$. Напомним, что в однородных координатах не существует единственного представления координатного вектора (табл. 2-1). Точка $[4 \ 3 \ 1]$ представлена в однородных координатах по всем направлениям. Заметим, что в этой таблице при $h \rightarrow 0$ отношение y^*/x^* остается равным $3/4$, как и требуется для сохранения уравнения. Кроме этого, обратим внимание на то, что следующая пара $(x^* \ y^*)$, все точки которой расположаются на линии $y^* = (3/4)x^*$, быстро приближается к бесконечности. Таким

образом, предел при $h \rightarrow 0$ есть точка бесконечности, заданная в однородных координатах как $[X \ Y \ h] = [4 \ 3 \ 0]$.

Обратившись снова к рис. 2-15, легко продемонстрировать геометрическую интерпретацию процесса движения к пределу при $h \rightarrow 0$. Рассмотрим отрезок единичной длины, проходящий от точки начала координат в направлении $[1 \ 0]$ на плоскости $h = s$ ($s < 1$). При $s \rightarrow 0$ проекция этой прямой обратно на физическую плоскость $h = 1$ в направлении лучей, проходящих через начало координат, становится бесконечной длины. Следовательно, конечная точка прямой должна представляться точкой бесконечности на оси X .

2-21 ПРАВИЛА ВЫПОЛНЕНИЯ ПРЕОБРАЗОВАНИЙ

Для представления данных и выполнения преобразований с помощью умножения матриц используются различные соглашения. Наибольшее внимание нужно уделять формулировке задач и интерпретации результатов. Например, перед выполнением поворота необходимо получить ответы на следующие вопросы.

В правосторонней или левосторонней системе координат определяются поворачиваемые координатные векторы?

Вращается объект или система координат?

Как определяются положительный и отрицательный повороты?

Координаты записываются в виде строки или столбца матрицы?

Вокруг какой линии или оси осуществляется поворот?

В данном изложении используется правосторонняя система координат, объект вращается в неподвижной координатной системе, положительный поворот определяется правилом правой руки, т. е. поворот по часовой стрелке осуществляется вокруг оси при наблюдении от начала вдоль положительной оси, и координатные векторы представляются в виде строки матрицы.

Выражение (2-29) задает преобразование для положительного поворота вокруг начала координат или оси z . Так как вектор задается строкой матрицы, то матрицу преобразования следует разместить *после* данных или матрицы координатных векторов. Это преобразование задается путем умножения справа. В случае однородных координат для положительного поворота объекта на угол θ вокруг начала координат (оси z) использование умножения справа приводит к следующему результату:

$$[X^*] = [X][R],$$

$$[x^* \ y^* \ 1] = [x \ y \ 1] \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2-59)$$

Если мы подставим координатные векторы, заданные в однородных координатах в виде столбца матрицы, то поворот можно выполнить следующим образом:

$$[X^*] = [R]^{-1}[X],$$

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (2-60)$$

Выражение (2-60) называется преобразованием с умножением слева, так как матрица преобразования расположена перед столбцом координатного вектора или данных.

Заметим, что (3×3) -матрица в выражении (2-60) есть транспозиция (3×3) -матрицы из выражения (2-59). Это свидетельствует о независимости строк и столбцов матрицы.

Для того чтобы повернуть систему координат и оставить неизмененными координатные векторы, необходимо в выражении (2-59) заменить θ на $-\theta$. Вспомним, что $\sin \theta = -\sin(-\theta)$, а $\cos \theta = \cos(-\theta)$.

Теперь выражение (2-59) будет иметь вид

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2-61)$$

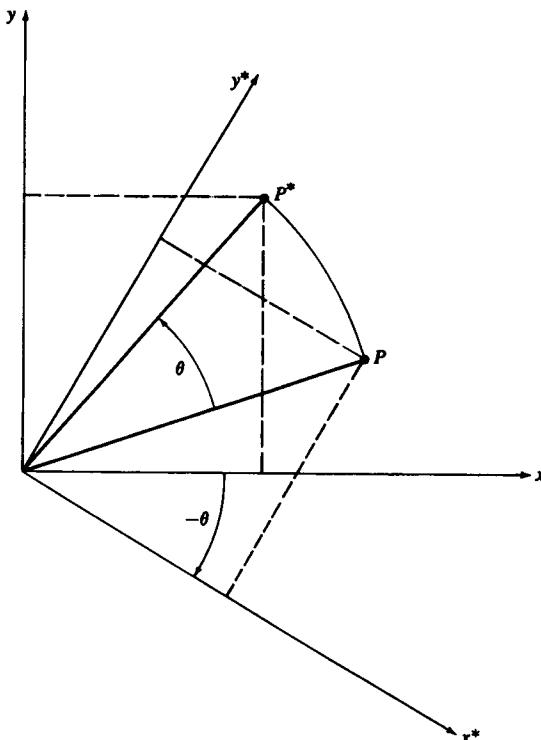


Рис. 2-16 Эквивалентность преобразования координатных векторов и систем координат.

Заметим, что (3×3) -матрица опять имеет обратную и также транспонируется в матрицу из (2-59).

Если вращается система координат и используется левосторонняя координатная система, то замену θ на $-\theta$ надо производить *дважды*, а уравнение (2-59) снова оказывается справедливым при допущении, что применяется последующее умножение на строку матрицы данных.

Заметим, что, как показано на рис. 2-16, вращение против часовой стрелки векторов, задающих объект, идентично повороту в том же направлении координатных осей при неподвижном объекте. Опять нет необходимости в изменении содержимого матрицы преобразования 3×3 , если нет других причин для ее редактирования. Эти несколько примеров показывают, насколько аккуратно необходимо выполнять матричные преобразования.

2-22 ЛИТЕРАТУРА

- 2-1 Fox L. *An Introduction to Numerical Linear Algebra*, Oxford University Press, London, 1964.
- 2-2 Forrest A. R. Coordinates, Transformations, and Visualization Techniques, CAD Group Document №. 23, Cambridge University, June 1969.

Глава 3

ПРОСТРАНСТВЕННЫЕ ПРЕОБРАЗОВАНИЯ И ПРОЕКЦИИ

3-1 ВВЕДЕНИЕ

Способность визуализировать или изображать пространственный объект является основой для понимания формы этого объекта. Кроме того, во многих случаях для этого важна способность вращать, переносить и строить виды проекций объекта. Все это легко демонстрируется на примере нашего знакомства с относительно сложным незнакомым объектом. Чтобы понять его форму, мы тут же начинаем вращать объект, отодвигать на расстояние вытянутой руки, передвигать вверх и вниз, вперед и назад и т. д. Чтобы сделать то же самое с помощью компьютера, мы должны распространить наш предшествующий двумерный анализ на три измерения. Основываясь на полученном опыте, мы немедленно вводим однородные координаты. Таким образом, точка в трехмерном пространстве $[x \ y \ z]$ представляется четырехмерным вектором

$$[x' \ y' \ z' \ h] = [x \ y \ z \ 1] [T],$$

где $[T]$ является матрицей некоего преобразования. Как и ранее, преобразование из однородных координат в обычные задается формулой

$$[x^* \ y^* \ z^* \ 1] = \left[\begin{array}{cccc} x' & y' & z' & 1 \\ \hline h & h & h & \end{array} \right]. \quad (3-1)$$

Обобщенную матрицу преобразования размерности 4×4 для трехмерных однородных координат можно представить в следующем виде:

$$[T] = \left[\begin{array}{cccc} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ l & m & n & s \end{array} \right]. \quad (3-2)$$

Матрицу преобразования 4×4 из (3-2) можно разделить на четыре отдельные части:

$$\left[\begin{array}{ccccc} & & & 3 & \\ 3 \times 3 & & & \vdots & 3 \\ & & & \vdots & \times \\ & & & \vdots & 1 \\ \dots & \dots & \dots & \vdots & \dots \\ 1 \times 3 & & & \vdots & 1 \times 1 \end{array} \right].$$

Верхняя левая (3×3) -подматрица задает линейное преобразование¹ в форме масштабирования, сдвига, отражения и вращения. Левая нижняя (1×3) -подматрица задает перемещение, а правая верхняя (3×1) -подматрица — перспективное преобразование. Последняя правая нижняя (1×1) -подматрица задает общее масштабирование. Общее преобразование, полученное после применения этой (4×4) -матрицы к однородному вектору и вычисления обычных координат, называется билинейным преобразованием². В общем случае данное преобразование осуществляет комбинацию сдвига, локального масштабирования, вращения, отражения, перемещения, перспективного преобразования и общего масштабирования.

3-2 ТРЕХМЕРНОЕ МАСШТАБИРОВАНИЕ

Диагональные элементы (4×4) -матрицы обобщенного преобразования задают локальное и общее масштабирование. Для иллюстрации этого рассмотрим преобразование

$$\begin{aligned} [X][T] &= [x \ y \ z \ 1] \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & j & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= [ax \ ey \ jz \ 1] = [x^* \ y^* \ z^* \ 1], \end{aligned} \quad (3-3)$$

которое показывает действие локального масштабирования. Ниже приводится пример.

Пример 3-1 Локальное масштабирование

Рассмотрим прямоугольный параллелепипед (*RPP*), показанный на рис. 3-1 *a* со следующими однородными координатами вершин:

$$[X] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 0 & 3 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 0 & 3 & 0 & 1 \end{bmatrix}.$$

Чтобы получить единичный куб из *RPP* с помощью локального масштабирования, необходимы масштабные множители $1/2, 1/3, 1$ вдоль осей x, y, z соответственно. Преобразование локального масштабирования задается матрицей

$$[T] = \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

¹ Линейное преобразование — это преобразование, переводящее линейную комбинацию векторов в ту же самую линейную комбинацию преобразованных векторов.

² Билинейное преобразование — это результат двух последовательных линейных преобразований.

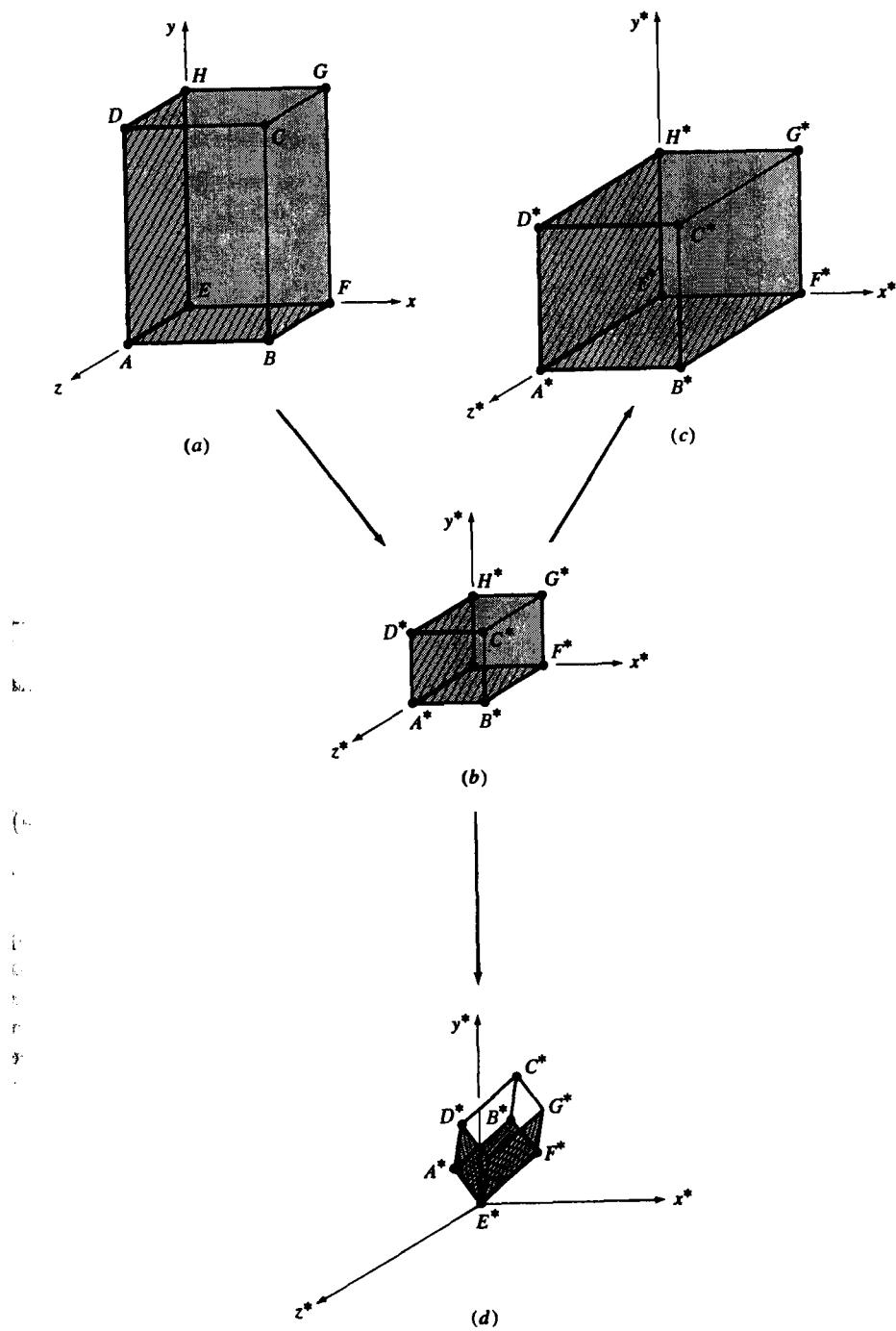


Рис. 3-1 Трехмерные масштабирования.

Результирующий куб имеет следующие однородные координаты вершин:

$$\begin{aligned} [X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 0 & 3 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 0 & 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \end{aligned}$$

Заметим, что однородный координатный множитель h равен единице для каждой из преобразованных вершин. Результат масштабирования показан на рис. 3-1 b.

Общее масштабирование можно осуществить, воспользовавшись четвертым диагональным элементом, т. е.

$$[X][T] = [x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} = [x' \ y' \ z' \ s]. \quad (3-4)$$

Обычные или физические координаты имеют вид

$$[x^* \ y^* \ z^* \ 1] = \left[\frac{x'}{s} \ \frac{y'}{s} \ \frac{z'}{s} \ 1 \right].$$

Этот результат снова иллюстрируется на примере.

Пример 3-2 Общее масштабирование

Для общего масштабирования единичного куба, изображенного на рис. 3-1 b, на множитель два (удвоение размера), необходимо преобразование (см. (3-4))

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 \end{bmatrix}.$$

Полученный в результате параллелепипед RP' имеет следующие однородные координаты вершин:

$$\begin{aligned} [X'] &= [X^*] [T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1/2 \end{bmatrix} = \\ &= \begin{bmatrix} 0 & 0 & 1 & 0.5 \\ 1 & 0 & 1 & 0.5 \\ 1 & 1 & 1 & 0.5 \\ 0 & 1 & 1 & 0.5 \\ 0 & 0 & 0 & 0.5 \\ 1 & 0 & 0 & 0.5 \\ 1 & 1 & 0 & 0.5 \\ 0 & 1 & 0 & 0.5 \end{bmatrix}. \end{aligned}$$

Заметим, что однородный координатный множитель h для каждой из преобразованных вершин равен 0.5. Таким образом, для того чтобы получить обычные или физические координаты, каждый вектор необходимо разделить на h . Результат, показанный на рис. 3-1 с, равен

$$[X^*] = \begin{bmatrix} 0 & 0 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix}.$$

Заметим здесь, что, как и в случае двумерного общего масштабирования, однородный координатный множитель не равен единице. По аналогии с предыдущим обсуждением (см. разд. 2-18) это означает преобразование из физического объема $h = 1$ в другой объем в 4-мерном пространстве. Преобразованные физические координаты получаются проецированием через центр 4-мерной координатной системы обратно в физический объем $h = 1$. Как и ранее, если $s < 1$, происходит однородное расширение. Если $s > 1$, происходит однородное сжатие координатного вектора.

Такой же результат можно получить, используя одинаковые коэффициенты локальных масштабирований. В этом случае матрица преобразования имеет вид

$$[T] = \begin{bmatrix} 1/s & 0 & 0 & 0 \\ 0 & 1/s & 0 & 0 \\ 0 & 0 & 1/s & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Отметим, что здесь однородный координатный множитель равен единице, т. е. $h = 1$. Таким образом, все преобразование происходит в физическом объеме $h = 1$.

3-3 ТРЕХМЕРНЫЕ СДВИГИ

Недиагональные элементы в верхней левой 3×3 -подматрице обобщенной матрицы преобразования размером 4×4 задают сдвиги в трех измерениях, т. е.

$$\begin{aligned} [X][T] &= [x \ y \ z \ 1] \begin{bmatrix} 1 & b & c & 0 \\ d & 1 & f & 0 \\ g & i & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= [x + yd + gz \ bx + y + iz \ cx + fy + z \ 1]. \end{aligned} \quad (3-5)$$

Поясним это на примере.

Пример 3-3 Сдвиги

Снова рассмотрим единичный куб, изображенный на рис. 3-1 б. Применив преобразование сдвигов

$$[T] = \begin{bmatrix} 1 & -0.85 & 0.25 & 0 \\ -0.75 & 1 & 0.7 & 0 \\ 0.5 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

получаем

$$\begin{aligned} [X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -0.85 & 0.25 & 0 \\ -0.75 & 1 & 0.7 & 0 \\ 0.5 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 0.5 & 1 & 1 & 1 \\ 1.5 & 0.15 & 1.25 & 1 \\ 0.75 & 1.15 & 1.95 & 1 \\ -0.25 & 2 & 1.7 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & -0.85 & 0.25 & 1 \\ 0.25 & 0.15 & 0.95 & 1 \\ -0.75 & 1 & 0.7 & 1 \end{bmatrix}. \end{aligned}$$

Этот результат показан на рис. 3-1 д. Заметим, что во всех трех примерах остается неизменной вершина параллелепипеда, находящаяся в начале координат.

3-4 ТРЕХМЕРНОЕ ВРАЩЕНИЕ

Прежде чем переходить к трехмерному вращению вокруг произвольной оси, рассмотрим вращение вокруг каждой из координатных осей. При вращении вокруг оси x остаются неизменными x -координаты координатного вектора. Фактически

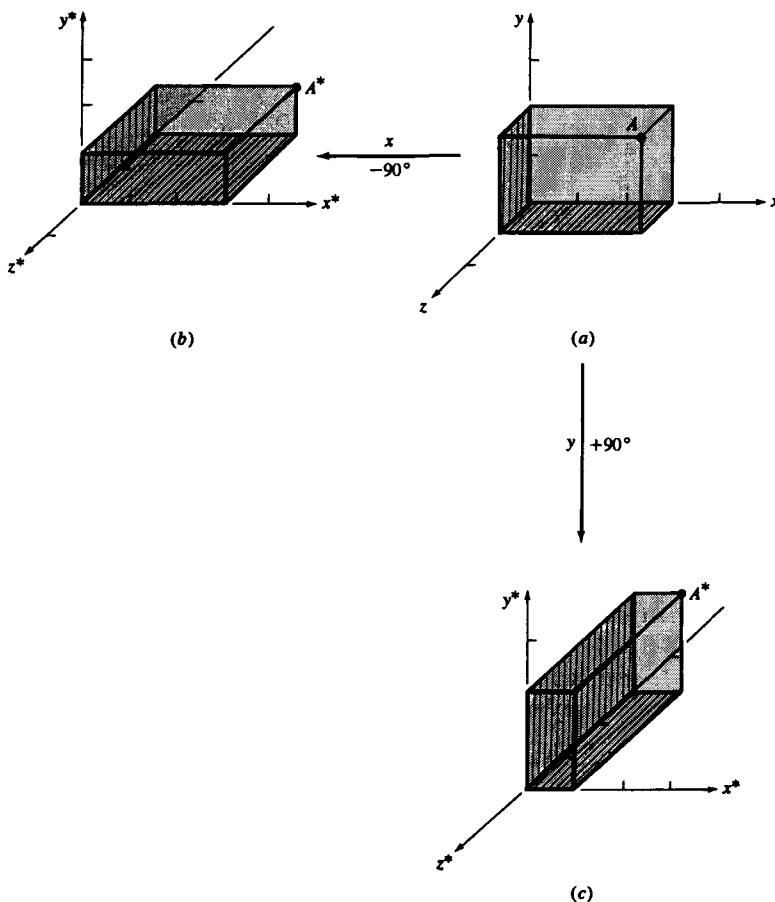


Рис. 3-2 Трехмерные повороты.

вращение происходит в плоскостях, перпендикулярных оси x . Аналогичным образом вращение вокруг осей y и z происходит в плоскостях, перпендикулярных осям y и z соответственно. Преобразование координатного вектора в каждой из этих плоскостей задается указанной в (2-29) матрицей двумерного вращения. Эта матрица и неизменность координаты x при вращении вокруг оси x позволяют записать 4×4 -преобразование однородных координат при повороте на угол θ в виде

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-6)$$

Вращение считается положительным в смысле правила правой руки, т. е. по часовой стрелке, если смотреть из начала координат в положительном направле-

нии оси вращения¹. На рис. 3-2 *b* показан параллелепипед, полученный поворотом на -90° вокруг оси *x* параллелепипеда с рис. 3-2 *a*.

Аналогично матрица преобразования для вращения вокруг оси *z* на угол ψ имеет вид

$$[T] = \begin{bmatrix} \cos \psi & \sin \psi & 0 & 0 \\ -\sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-7)$$

При вращении на угол ϕ вокруг оси *y* преобразование имеет вид

$$[T] = \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-8)$$

Заметим, что в (3-8) знаки у синусов противоположны знакам этих членов в равенствах (3-6) и (3-7). Это нужно для того, чтобы выполнялось соглашение о положительном направлении по правилу правой руки.

Из равенств (3-6)–(3-8) следует, что детерминант каждой из матриц преобразований равен +1, что и необходимо для чистого вращения. Более полно эти результаты проиллюстрирует пример.

Пример 3-4 Вращение

Рассмотрим прямоугольный параллелепипед, изображенный на рис. 3-2 *a*. Матрица $[X]$ координатного вектора имеет вид

$$[X] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} A.$$

В данном случае строка, отмеченная буквой *A* в матрице $[X]$, соответствует точке *A* на рис. 3-2. Равенство (3-6) для поворота на $\theta = -90^\circ$ вокруг оси *x* приводит к следующему преобразованию:

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

¹ Правило правой руки для вращения формулируется следующим образом: яправьте большой палец правой руки в положительном яправлении оси вращения. Тогда естественное спиральное движение пальцев указывает положительное яправление вращения.

Применение этого преобразования дает новые координаты:

$$\begin{aligned}
 [X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 3 & 1 & 0 & 1 \\ 3 & 1 & -2 & 1 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 0 & -2 & 1 \\ 0 & 0 & -2 & 1 \end{bmatrix} A^*
 \end{aligned}$$

Отметим, что, как и должно быть, x -компоненты $[X]$ и $[X^*]$ идентичны. Результат данного поворота изображен на рис. 3-2 б.

Для поворота на угол $\phi = +90^\circ$ вокруг оси y уравнение (3-7) дает следующую матрицу преобразования:

$$[T'] = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Снова применив преобразование к исходному параллелепипеду, получим новые координаты:

$$\begin{aligned}
 [X^{**}] &= [X][T'] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & -3 & 1 \\ 1 & 2 & -3 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -3 & 1 \\ 0 & 2 & -3 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} A^*
 \end{aligned}$$

Заметим, что в этом случае идентичны y -компоненты $[X]$ и $[X^{**}]$. Результат изображен на рис. 3-2 с.

Так как трехмерные вращения получаются с помощью перемножения матриц, то они не коммутативны; т. е. порядок перемножения влияет на конечный результат.

тат (см. разд. 2-12). Чтобы показать это, рассмотрим два последовательных поворота на один и тот же угол — сначала вокруг оси x , затем вокруг оси y . Используя уравнения (3-6) и (3-8) с $\theta = \phi$, мы получим

$$\begin{aligned} [T] &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos^2 \theta & 0 & -\sin \theta & 0 \\ \sin^2 \theta & \cos \theta & \cos \theta \sin \theta & 0 \\ \cos \theta \sin \theta & -\sin \theta & \cos^2 \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3-9)$$

С другой стороны, обратная операция, т. е. поворот вокруг оси y , а потом вокруг оси x с углом $\theta = \phi$ дает

$$\begin{aligned} [T] &= \begin{bmatrix} \cos \theta & 0 & -\sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos \theta & \sin^2 \theta & -\cos \theta \sin \theta & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ \sin \theta & -\cos \theta \sin \theta & \cos^2 \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3-10)$$

Сравнивая правые части (3-9) и (3-10), видим, что они не одинаковы. Если надо сделать более одного поворота, то следует помнить о некоммутируемости трехмерных вращений.

На рис. 3-3 *c* и 3-3 *d* штриховой линией изображен результат преобразования, состоящего из двух поворотов на 90° при помощи произведения матриц из (3-9) для объекта, показанного на рис. 3-3 *a*. Осуществляя повороты, заданные (3-10), в обратном порядке, получим фигуры, нарисованные сплошными линиями на рис. 3-3 *b* и 3-3 *d*. Рис. 3-3 *d* наглядно показывает, что при изменении порядка вращения получаются различные результаты. Приведенный ниже численный пример иллюстрирует это.

Пример 3-5 Комбинированные повороты

Объект на рис. 3-2 *a* имеет следующие координатные векторы:

$$[X] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} A.$$

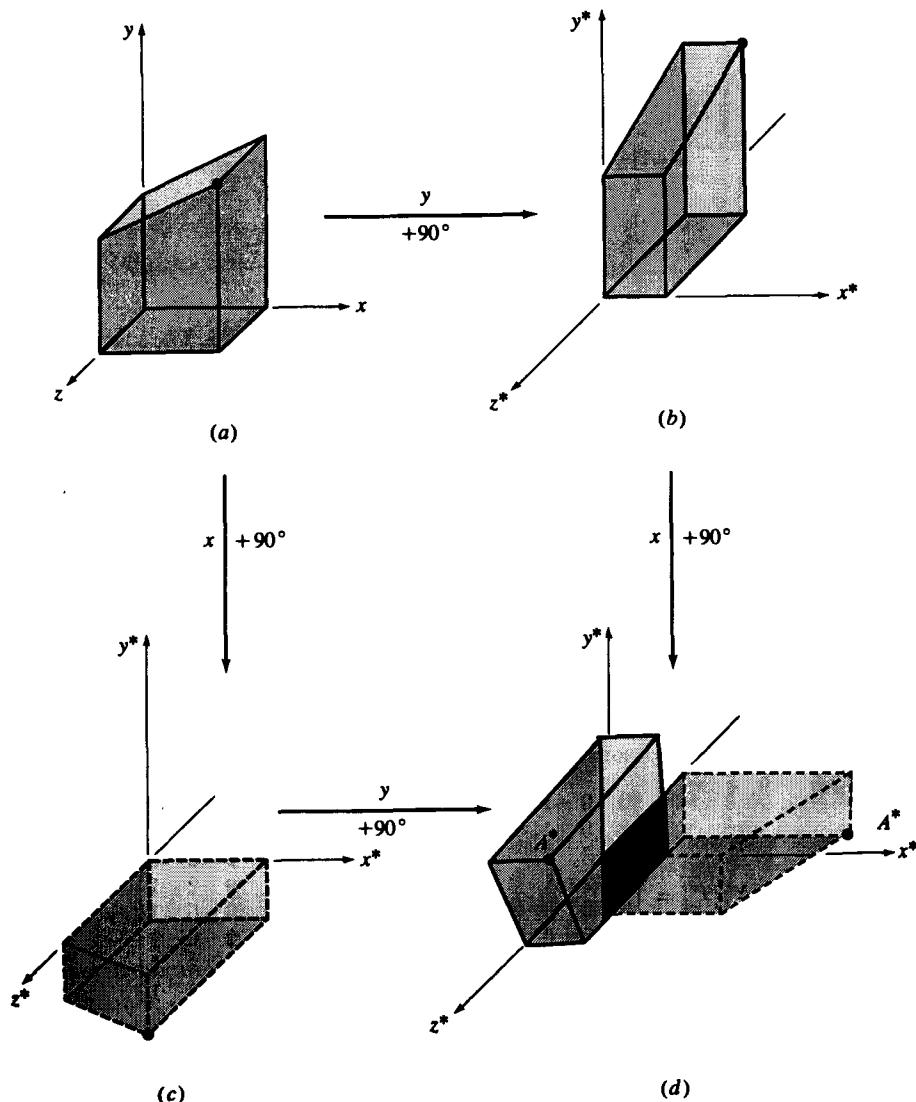


Рис. 3-3 Некоммутативность трехмерных поворотов.

Общая матрица для вращения сначала вокруг оси x на угол $\theta = 90^\circ$, а затем вокруг оси y на угол $\phi = 90^\circ$ задается уравнением (3-9) в виде

$$[T] = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Преобразованные координатные векторы равны

$$[X^*] = [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & -1 & 0 & 1 \\ 0 & -1 & -2 & 1 \\ 3 & -1 & -2 & 1 \\ 2 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -2 & 1 \\ 3 & 0 & -2 & 1 \\ 2 & 0 & 0 & 1 \end{bmatrix} A^*$$

Преобразованный объект изображен штриховой линией на рис. 3-3 d.

Общая матрица для вращения сначала вокруг оси y на угол $\phi = 90^\circ$, а затем вокруг оси x на угол $\theta = 90^\circ$ задается уравнением (3-10) в виде

$$[T'] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

В этом случае преобразованные координатные векторы равны

$$[X^{**}] = [X][T'] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 2 & 3 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 0 & 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 0 & 2 & 3 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix} A^{**}$$

Преобразованный объект изображен сплошными линиями на рис. 3-3 d.

Сравнение двух числовых результатов также ясно показывает, что ориентация преобразованных объектов совершенно различна. Следовательно, порядок перемножения матриц очень важен.

3-5 ТРЕХМЕРНОЕ ОТРАЖЕНИЕ

Некоторые ориентации трехмерного объекта нельзя получить одними вращениями, требуются преобразования отражения. В трехмерном пространстве отражение происходит относительно плоскости. По аналогии с обсуждавшимся ранее двумерным отражением (см. разд. 2-10), трехмерное отражение относительно плоскости эквивалентно вращению вокруг оси в трехмерном пространстве в четырехмерное пространство и обратно в исходное трехмерное пространство. Для чистого отражения детерминант матрицы равен -1 .

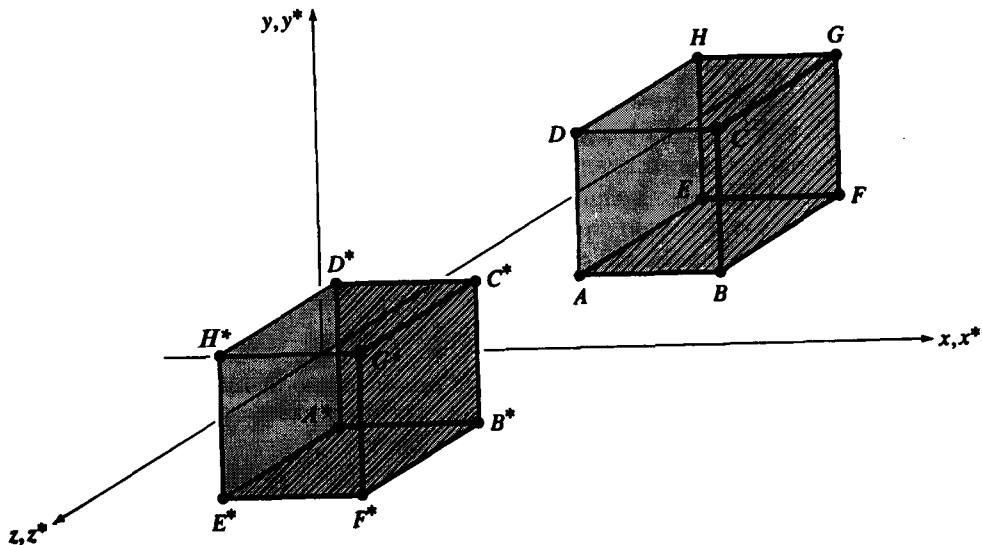


Рис. 3-4 Трехмерное отражение относительно плоскости xy .

При отражении относительно плоскости xy изменяются только значения z -координаты координатного вектора объекта. В самом деле, они изменяют знак. Таким образом, матрица преобразования для отражения относительно плоскости xy равна

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-11)$$

На рис. 3-4 изображено отражение единичного куба относительно плоскости xy . При отражении относительно плоскости yz

$$[T] = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3-12)$$

а для отражения относительно плоскости xz

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-13)$$

Эти результаты иллюстрируются в приведенном ниже численном примере.

Пример 3-6 Отражение

Параллелепипед $ABCDEFGH$, изображенный на рис. 3-4, имеет координатный вектор

$$[X] = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 2 & 0 & -1 & 1 \\ 2 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 0 & -2 & 1 \\ 2 & 0 & -2 & 1 \\ 2 & 1 & -2 & 1 \\ 1 & 1 & -2 & 1 \end{bmatrix}.$$

Матрица преобразования для отражения относительно плоскости xy задается посредством (3-11). После отражения преобразованные координатные векторы равны

$$[X^*] = [X][T] = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 2 & 0 & -1 & 1 \\ 2 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 0 & -2 & 1 \\ 2 & 0 & -2 & 1 \\ 2 & 1 & -2 & 1 \\ 1 & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 1 & 2 & 1 \end{bmatrix}.$$

Результат $A^*B^*C^*D^*E^*F^*G^*H^*$ показан на рис. 3-4.

3-6 ПРОСТРАНСТВЕННЫЙ ПЕРЕНОС

Матрица пространственного переноса имеет вид:

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix}. \quad (3-14)$$

Перемещенные однородные координаты получаются с помощью преобразования

$$\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix}.$$

Выполнив умножение, получим

$$\begin{bmatrix} x' & y' & z' & h \end{bmatrix} = [(x + l) \quad (y + m) \quad (z + n) \quad 1]. \quad (3-15)$$

Из этого следует, что преобразованные физические координаты равны

$$\begin{aligned} x^* &= x + l, \\ y^* &= y + m, \\ z^* &= z + n. \end{aligned}$$

3-7 КОМПОЗИЦИИ ПРЕОБРАЗОВАНИЙ

Последовательные преобразования могут быть скомбинированы или объединены в одно (4×4)-преобразование, дающее тот же самый результат. Так как перемножение матриц является некоммутативной операцией, то важен порядок ее выполнения (в общем случае $[A][B] \neq [B][A]$). Правильный порядок определяется положением конкретной матрицы преобразования относительно матрицы координатного вектора. Матрица, ближайшая к матрице координатного вектора, задает первое преобразование, а последняя — последнее преобразование. Математически это можно записать следующим образом:

$$[X][T] = [X][T_1][T_2][T_3][T_4]\dots,$$

где

$$[T] = [T_1][T_2][T_3][T_4]\dots$$

и $[T_i]$ являются произвольной комбинацией матриц масштабирования, сдвига, вращения, отражения, переноса, перспективного преобразования и проецирования. Так как перспективные преобразования искажают геометрические объекты (см. разд. 3-15), а преобразования проецирования приводят к потере информации (см. разд. 3-12), то в случае наличия этих матриц они должны быть расположены соответственно предпоследней и последней по порядку.

Эти идеи проиллюстрированы в приведенном ниже примере.

Пример 3-7 Композиции преобразований

Рассмотрим для заданного в однородных координатах координатного вектора $[1 \ 2 \ 1 \ 1]$ результат переноса в направлениях x, y, z на $-1, -1, -1$ соответственно, а затем поворота на $+30^\circ$ вокруг оси x и поворота на $+45^\circ$ вокруг оси y .

Сначала получим матрицу комбинированного преобразования. Из равенств (3-14), (3-6) и (3-8) следует, что

$$\begin{aligned} [T] &= [Tr][R_x][R_y] = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ \sin \phi \sin \theta & \cos \theta & \cos \phi \sin \theta & 0 \\ \sin \phi \cos \theta & -\sin \theta & \cos \phi \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ \sin \phi \sin \theta & \cos \theta & \cos \phi \sin \theta & 0 \\ \sin \phi \cos \theta & -\sin \theta & \cos \phi \cos \theta & 0 \\ l \cos \phi & m \cos \theta & -l \sin \phi & 1 \\ +m \sin \phi \sin \theta & -n \sin \theta & +m \cos \phi \sin \theta & \\ +n \sin \phi \cos \theta & +n \cos \theta & +n \cos \phi \cos \theta & \end{bmatrix}, \end{aligned} \tag{3-16}$$

где θ и ϕ — соответственно углы вращения относительно осей x и y ; а l, m, n — величины переноса в направлениях x, y, z .

В общем случае для координатного вектора мы имеем

$$\begin{aligned}
 [X][T] &= [x \ y \ z \ 1] \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ \sin \phi \sin \theta & \cos \theta & \cos \phi \sin \theta & 0 \\ \sin \phi \cos \theta & -\sin \theta & \cos \phi \cos \theta & 0 \\ l \cos \phi & m \cos \theta & -l \sin \phi & 1 \\ +m \sin \phi \sin \theta & -n \sin \theta & +m \cos \phi \sin \theta & \\ +n \sin \phi \cos \theta & & +n \cos \phi \cos \theta & \end{bmatrix} = \\
 &= \begin{bmatrix} (x+l) \cos \phi & (y+m) \cos \theta & -(x+l) \sin \phi & 1 \\ +(y+m) \sin \phi \sin \theta & -(z+n) \sin \theta & +(y+m) \cos \phi \sin \theta & \\ +(z+n) \sin \phi \cos \theta & & +(z+n) \cos \phi \cos \theta & \end{bmatrix}.
 \end{aligned}$$

Для конкретных значений $\theta = +30^\circ$, $\phi = +45^\circ$, $l = -1$, $m = 1$, $n = -1$ преобразуемый координатный вектор имеет вид $[3 \ 2 \ 1 \ 1]$.

$$[X][T] = [3 \ 2 \ 1 \ 1] \begin{bmatrix} 0.707 & 0 & -0.707 & 0 \\ 0.354 & 0.866 & 0.354 & 0 \\ 0.612 & -0.5 & 0.612 & 0 \\ -1.673 & -0.366 & -0.259 & 1 \end{bmatrix},$$

$$[X][T] = [1.768 \ 0.866 \ -1.061 \ 1].$$

Чтобы убедиться, что общая матрица дает тот же самый результат, как и последовательное применение матриц, рассмотрим

$$\begin{aligned}
 [X'] &= [X][Tr] = \\
 &= [3 \ 2 \ 1 \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix} = \\
 &= [2 \ 1 \ 0 \ 1],
 \end{aligned}$$

$$\begin{aligned}
 [X''] &= [X'][R_x] = [2 \ 1 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & -0.5 & 0.866 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= [2 \ 0.866 \ 0.5 \ 1],
 \end{aligned}$$

$$\begin{aligned}
 [X'''] &= [X''][R_y] = [2 \ 0.866 \ 0.5 \ 1] \begin{bmatrix} 0.707 & 0 & -0.707 & 0 \\ 0 & 1 & 0 & 0 \\ 0.707 & 0 & 0.707 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= [1.768 \ 0.866 \ -1.061 \ 1],
 \end{aligned}$$

что доказывает наше утверждение.

3-8 ПОВОРОТЫ ВОКРУГ ОСИ, ПАРАЛЛЕЛЬНОЙ КООРДИНАТНОЙ ОСИ

Преобразования, заданные равенствами (3-6)–(3-8), описывают вращение вокруг координатных осей x , y и z . Однако часто бывает необходимо вращать объект вокруг оси, не совпадающей с этими тремя. Мы рассмотрим частный случай для оси, параллельной одной из координатных осей x , y или z . На рис. 3-5 изображено тело в локальной системе осей $x'y'z'$, параллельных фиксированной глобальной системе xyz . Вращение тела вокруг любой из локальных осей x' , y' или z' выполняется с помощью следующей процедуры:

- переместить тело так, чтобы локальная ось совпала с координатной;
- повернуть вокруг указанной оси;
- переместить преобразованное тело в исходное положение.

Математически это можно записать так:

$$[X^*] = [X][Tr][R_x][Tr]^{-1},$$

где

- $[X^*]$ – преобразованное тело,
- $[X]$ – исходное тело,
- $[Tr]$ – матрица перемещения,
- $[R_x]$ – соответствующая матрица поворота,
- $[Tr]^{-1}$ – матрица, обратная к матрице перемещения.

Ниже приводится иллюстративный пример.

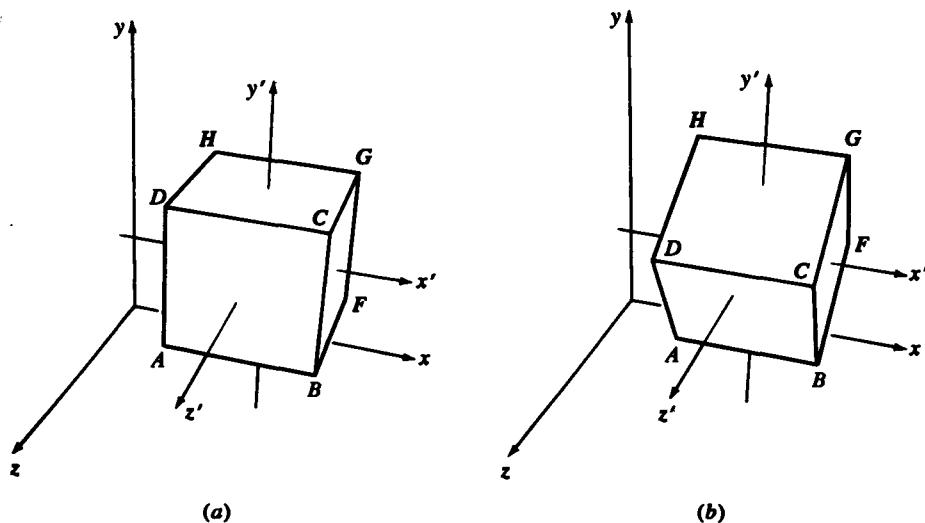


Рис. 3-5 Поворот вокруг оси, параллельной одной из координатных осей.

Пример 3-8 Единственное относительное вращение

Рассмотрим параллелепипед, изображенный на рис. 3-5 а, заданный координатными векторами

$$[X] = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix} \begin{array}{l} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array}$$

относительно глобальной системы координат xyz . Повернем параллелепипед на $\theta = +30^\circ$ относительно локальной оси x' , проходящей через центр параллелепипеда. Предполагается, что начало локальной системы координат находится в центре параллелепипеда. Координаты этого центра равны $[x_c \ y_c \ z_c \ 1] = [3/2 \ 3/2 \ 3/2 \ 1]$. Вращение осуществляется следующим образом:

$$[X^*] = [X][Tr][R][Tr]^{-1},$$

где

$$[Tr] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -y_c & -z_c & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -3/2 & -3/2 & 1 \end{bmatrix},$$

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & -0.5 & 0.866 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

и

$$[Tr]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & y_c & z_c & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 3/2 & 3/2 & 1 \end{bmatrix}.$$

Первая матрица $[Tr]$ сдвигает параллелепипед параллельно плоскости $x = 0$ до тех пор, пока ось x' не совпадет с осью x . Вторая матрица $[R_x]$ выполняет требуемое вращение вокруг оси x , третья матрица $[Tr]^{-1}$ переносит ось x' , а следовательно, и повернутый параллелепипед, обратно в исходное положение.

Объединив эти три матрицы, получим

$$\begin{aligned} [T] &= [Tr][R_x][Tr]^{-1} = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & y_c(1 - \cos \theta) + z_c \sin \theta & z_c(1 - \cos \theta) - y_c \sin \theta & 1 \end{bmatrix}. \end{aligned}$$

После подстановки числовых значений преобразованные координаты примут вид:

$$[X'] = [X][T] = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & -0.5 & 0.866 & 0 \\ 0 & 0.951 & -0.549 & 1 \end{bmatrix},$$

$$[X'] = \begin{bmatrix} 1 & 0.817 & 1.683 & 1 \\ 2 & 0.817 & 1.683 & 1 \\ 2 & 1.683 & 2.183 & 1 \\ 1 & 1.683 & 2.183 & 1 \\ 1 & 1.317 & 0.817 & 1 \\ 2 & 1.317 & 0.817 & 1 \\ 2 & 2.183 & 1.317 & 1 \\ 1 & 2.183 & 1.317 & 1 \end{bmatrix} \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{matrix}.$$

Результат изображен на рис. 3-5 б.

В предыдущем примере требовалось только вращение вокруг единственной оси, параллельной одной из координатных осей. Таким образом надо только было сделать так, чтобы ось вращения совпала с соответствующей координатной осью. Для того чтобы совершить несколько поворотов в локальной системе осей, параллельных осям глобальной системы координат, надо совместить начала локальной и глобальной систем. Конкретнее, повороты могут быть выполнены с помощью следующей процедуры:

- переместить локальную систему осей так, чтобы начала локальной и глобальной систем совпали;
- выполнить требуемые повороты;
- переместить локальную систему осей обратно в исходное положение. Ниже эта процедура иллюстрируется на примере.

Пример 3-9 Несколько относительных вращений

Рассмотрим снова параллелепипед, изображенный на рис. 3-5 а. Пусть его требуется повернуть сначала на угол $\phi = -45^\circ$ вокруг оси y' , а затем на угол $\theta = +30^\circ$ вокруг оси x' . Для этого надо совместить начала систем координат $x'y'z'$ и xyz , выполнить необходимые повороты, а результат вернуть обратно в исходное положение.

Объединенное преобразование имеет вид:

$$[X'] = [X][T] = [X][Tr][R_y][R_x][Tr]^{-1}.$$

Конкретнее,

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_c & -y_c & -z_c & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_c & y_c & z_c & 1 \end{bmatrix},$$

где ϕ и θ обозначают углы поворотов вокруг осей y' и x' , соответственно.
Объединив эти матрицы, получим

$$[T] = \begin{bmatrix} \cos \phi & \sin \phi \sin \theta & -\sin \phi \cos \theta & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ \sin \phi & -\cos \phi \sin \theta & \cos \phi \cos \theta & 0 \\ x_c(1 - \cos \phi) & -x_c \sin \phi \sin \theta & x_c \sin \phi \cos \theta & 1 \\ -z_c \sin \phi & +y_c(1 - \cos \theta) & -y_c \sin \theta & 0 \\ & +z_c \cos \phi \sin \theta & +z_c(1 - \cos \phi \cos \theta) & 0 \end{bmatrix}. \quad (3-17)$$

Тогда преобразованные координатные векторы равны

$$[X'] = \begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 1 & 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.707 & -0.354 & 0.612 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ -0.707 & -0.354 & 0.612 & 0 \\ 1.5 & 1.262 & -1.087 & 1 \end{bmatrix},$$

$$[X'] = \begin{bmatrix} 0.793 & 1.067 & 1.25 & 1 \\ 1.5 & 0.713 & 1.862 & 1 \\ 1.5 & 1.579 & 2.362 & 1 \\ 0.793 & 1.933 & 1.75 & 1 \\ 1.5 & 1.421 & 0.638 & 1 \\ 2.207 & 1.067 & 1.25 & 1 \\ 2.207 & 1.933 & 1.75 & 1 \\ 1.5 & 2.287 & 1.138 & 1 \end{bmatrix}.$$

Результат изображен на рис. 3-6.

3-9 ПОВОРОТ ВОКРУГ ПРОИЗВОЛЬНОЙ ОСИ В ПРОСТРАНСТВЕ

Обобщенный случай поворота вокруг произвольной оси в пространстве встречается часто, например в робототехнике, мультиплексии, моделировании. Следуя логике предыдущего обсуждения, поворот вокруг произвольной оси в пространстве выполняется с помощью переноса и простых поворотов вокруг координатных осей. Так как метод поворота вокруг координатной оси известен, то основная идея заключается в том, чтобы совместить произвольную ось вращения с одной из координатных осей.

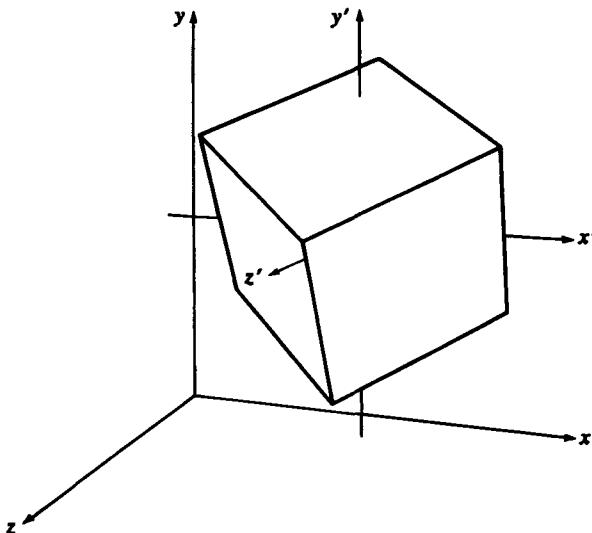


Рис. 3-6 Несколько поворотов относительно локальной системы координат.

Предположим, что произвольная ось в пространстве проходит через точку (x_0, y_0, z_0) с направляющим вектором (c_x, c_y, c_z) . Поворот вокруг этой оси на некоторый угол δ выполняется по следующему правилу:

- выполнить перенос так, чтобы точка (x_0, y_0, z_0) находилась в начале системы координат;
- выполнить соответствующие повороты так, чтобы ось вращения совпала с осью z ¹;
- выполнить поворот на угол δ вокруг оси z ;
- выполнить преобразование, обратное тому, что позволило совместить ось вращения с осью z ;
- выполнить обратный перенос.

В общем случае для того, чтобы произвольная ось, проходящая через начало координат, совпала с одной из координатных осей, необходимо сделать два последовательных поворота вокруг двух других координатных осей. Для совмещения произвольной оси вращения с осью z сначала выполним поворот вокруг оси x , а затем вокруг оси y . Чтобы определить угол поворота α вокруг оси x , используемый для перевода произвольной оси в плоскость xz , спроектируем сначала на плоскость yz направляющий единичный вектор этой оси (рис. 3-7 а). Компоненты y и z спроектированного вектора равны c_y - и c_z -компонентам единичного направляющего вектора оси вращения.

Из рис. 3-7 а следует, что

$$d = \sqrt{c_y^2 + c_z^2} \quad (3-18)$$

¹ Выбор координатной оси (в данном случае z) произволен.

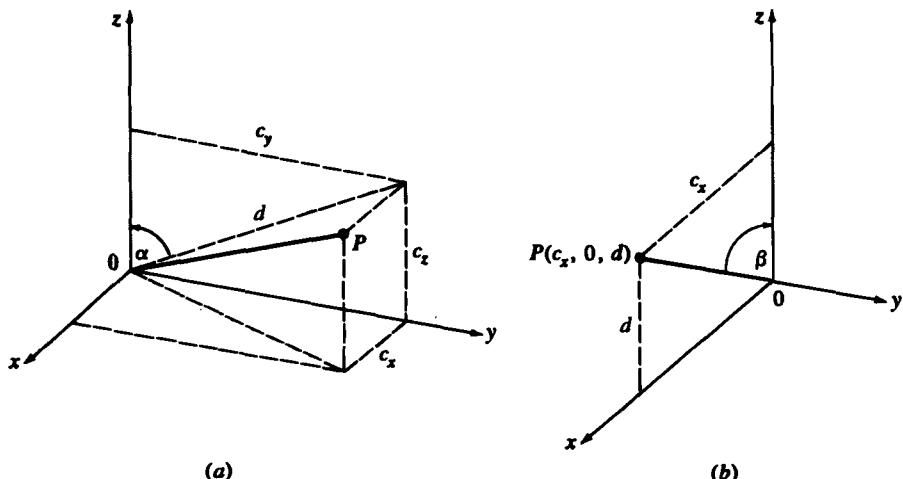


Рис. 3-7 Повороты, необходимые для совмещения с осью z единичного вектора OP .
(a) Поворот вокруг x ; (b) поворот вокруг y .

и

$$\cos \alpha = \frac{c_z}{d}, \quad \sin \alpha = \frac{c_y}{d}. \quad (3-19)$$

После перевода в плоскость xz с помощью поворота вокруг оси x , z -компоненту единичного вектора равна d , а x -компонента равна c_x , т. е. x -компоненте направляющего вектора, как это показано на рис. 3-7 b. Длина единичного вектора равна, конечно, 1. Таким образом, угол поворота β вокруг оси y , необходимый для совмещения произвольной оси с осью z , равен

$$\cos \beta = d, \quad \sin \beta = c_x. \quad (3-20)$$

Тогда полное преобразование можно представить в виде

$$[M] = [T] [R_x] [R_y] [R_\delta] [R_y]^{-1} [R_x]^{-1} [T]^{-1}, \quad (3-21)$$

где матрица переноса равна

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & -z_0 & 1 \end{bmatrix}. \quad (3-22)$$

Матрица преобразования поворота вокруг оси x

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_z/d & c_y/d & 0 \\ 0 & -c_y/d & c_z/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-23)$$

и вокруг оси y

$$[R_y] = \begin{bmatrix} \cos(-\beta) & 0 & -\sin(-\beta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(-\beta) & 0 & \cos(-\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} d & 0 & c_x & 0 \\ 0 & 1 & 0 & 0 \\ -c_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-24)$$

И, наконец, вращение вокруг произвольной оси задается матрицей поворота вокруг оси z

$$[R]_z = \begin{bmatrix} -\cos \delta & \sin \delta & 0 & 0 \\ -\sin \delta & \cos \delta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-25)$$

На практике углы α и β не вычисляются явным образом. Элементы матриц поворотов $[R_x]$ и $[R_y]$ в (3-21) получаются из уравнений (3-18)–(3-20) за счет выполнения двух операций деления и извлечения квадратного корня. Хотя данные результаты были разработаны для произвольной оси в первом квадранте, они применимы во всех квадрантах.

Если компоненты направляющего вектора произвольной оси неизвестны, то, зная вторую точку (x_1, y_1, z_1) на оси, их можно определить, нормализовав вектор, соединяющий первую и вторую точки. Более точно вектор оси из (x_0, y_0, z_0) в (x_1, y_1, z_1) равен

$$[V] = [(x_1 - x_0) (y_1 - y_0) (z_1 - z_0)].$$

Нормализация дает компоненты направляющего вектора:

$$[c_x \ c_y \ c_z] = \frac{[(x_1 - x_0) \ (y_1 - y_0) \ (z_1 - z_0)]}{[(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2]^{\frac{1}{2}}}. \quad (3-26)$$

Более полно этот метод иллюстрируется в следующем примере.

Пример 3-10 Поворот вокруг произвольной оси

Рассмотрим куб с одним отсеченным углом, изображенный на рис. 3-8 а. Координатные векторы вершин равны

$$[X] = \left[\begin{array}{cccc} 2 & 1 & 2 & 1 \\ 3 & 1 & 2 & 1 \\ 3 & 1.5 & 2 & 1 \\ 2.5 & 2 & 2 & 1 \\ 2 & 2 & 2 & 1 \\ 2 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 \\ 3 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ 3 & 2 & 1.5 & 1 \end{array} \right] \begin{array}{l} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ I \\ J \end{array}$$

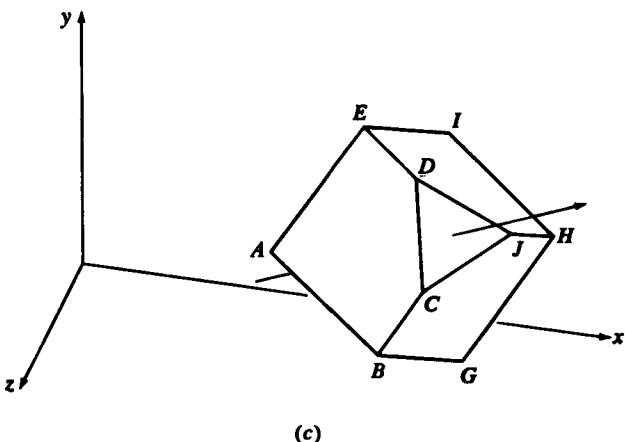
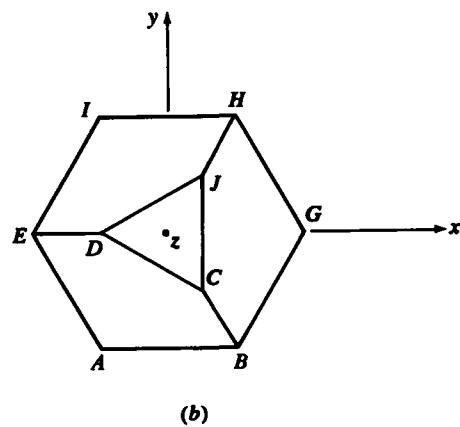
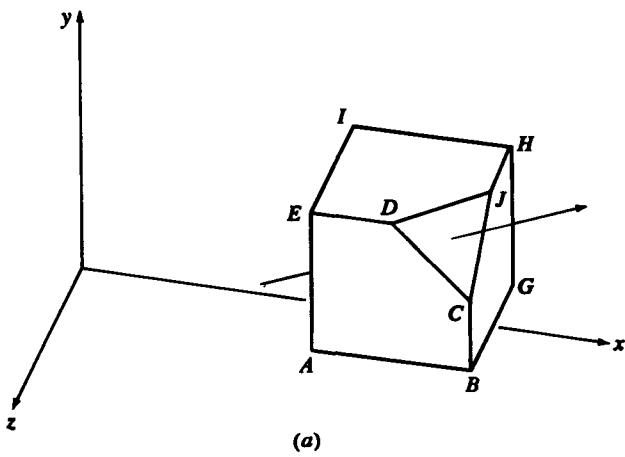


Рис. 3-8 Поворот вокруг произвольной оси.

Куб необходимо повернуть на -45° вокруг оси, проходящей через точку F и противоположный угол на диагонали. Ось направлена из точки F в противоположный угол и проходит через центр угловой грани.

Сначала определим компоненты направляющего единичного вектора оси вращения. Учитывая, что угол, отрезанный треугольником CDJ также лежит на оси, из (3-26) следует, что

$$\begin{bmatrix} c_x & c_y & c_z \end{bmatrix} = \frac{[(3-2) \quad (2-1) \quad (2-1)]}{((3-2)^2 + (2-1)^2 + (2-1)^2)^{1/2}} = \\ = [1/\sqrt{3} \quad 1/\sqrt{3} \quad 1/\sqrt{3}].$$

С помощью уравнений (3-18)–(3-20) получаем

$$d = \sqrt{(1/\sqrt{3})^2 + (1/\sqrt{3})^2} = \sqrt{2/3}$$

и

$$\alpha = \cos^{-1}(1/\sqrt{3}/\sqrt{2/3}) = \cos^{-1}(1/\sqrt{2}) = 45^\circ,$$

$$\beta = \cos^{-1}(\sqrt{2/3}) = 35, 26^\circ.$$

Так как точка F лежит на оси вращения, матрица переноса равна

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -2 & -1 & -1 & 1 \end{bmatrix}.$$

Тогда матрицы поворотов, совмещающих произвольную ось вращения с осью z , имеют вид:

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

и

$$[R_y] = \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} & 0 \\ 0 & 1 & 0 & 0 \\ -1/\sqrt{3} & 0 & 2/\sqrt{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Обратные матрицы $[R_x]^{-1}$, $[R_y]^{-1}$ и $[T]^{-1}$ получаются с помощью подстановки значений $-\alpha$, $-\beta$ и (x_0, y_0, z_0) для углов α , β и $(-x_0, -y_0, -z_0)$, соответственно, в уравнения (3-22)–(3-24).

Объединив $[T]$, $[R_x]$, $[R_y]$, получим

$$[M] = [T][R_x][R_y] = \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} & 0 \\ -2/\sqrt{6} & 0 & -4/\sqrt{3} & 1 \end{bmatrix}.$$

Преобразованные промежуточные координаты равны

$$[X][M] = \begin{bmatrix} -0.408 & -0.707 & 0.577 & 1 \\ 0.408 & -0.707 & 1.155 & 1 \\ 0.204 & -0.354 & 1.443 & 1 \\ -0.408 & 0 & 1.443 & 1 \\ -0.816 & 0 & 0.155 & 1 \\ 0 & 0 & 0 & 1 \\ 0.816 & 0 & 0.577 & 1 \\ 0.408 & 0.707 & 1.155 & 1 \\ -0.408 & 0.707 & 0.577 & 1 \\ 0.204 & 0.354 & 1.443 & 1 \end{bmatrix}.$$

Этот промежуточный результат изображен на рис. 3-8 б. Заметим, что точка F равна $(0, 0, 0)$.

Вращение вокруг произвольной оси теперь эквивалентно вращению вокруг оси z . Следовательно (см. (3-7)),

$$[R_\delta] = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Преобразованный объект возвращается в его «исходное» местоположение в пространстве с помощью преобразования

$$[M]^{-1} = [R_y]^{-1}[R_x]^{-1}[T]^{-1} = \begin{bmatrix} 2/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} & 0 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 0 \\ 2 & 1 & 1 & 1 \end{bmatrix}.$$

Этот результат можно получить либо обращением и объединением матриц, составляющих $[M]$, либо обращением самой матрицы $[M]$. Заметим, между прочим, что $[R_x][R_y]$ представляют только повороты. Верхняя левая (3×3) -подматрица $[M]^{-1}$ является точным транспонированием верхней левой (3×3) -подматрицы $[M]$.

Координаты результата равны

$$[X][M][R_\delta][M]^{-1} = \begin{bmatrix} 1.689 & 1.506 & 1.805 & 1 \\ 2.494 & 1.195 & 2.311 & 1 \\ 2.747 & 1.598 & 2.155 & 1 \\ 2.598 & 2.155 & 1.747 & 1 \\ 2.195 & 2.311 & 1.494 & 1 \\ 2 & 1 & 1 & 1 \\ 2.805 & 0.689 & 1.506 & 1 \\ 3.311 & 1.494 & 1.195 & 1 \\ 2.506 & 1.805 & 0.689 & 1 \\ 3.155 & 1.747 & 1.598 & 1 \end{bmatrix},$$

где

$$[M][R_\delta][M]^{-1} = \begin{bmatrix} 0.805 & -0.311 & 0.506 & 0 \\ 0.506 & 0.805 & -0.311 & 0 \\ -0.311 & 0.506 & 0.805 & 0 \\ 0.195 & 0.311 & -0.506 & 1 \end{bmatrix}.$$

На рис. 3-8 с изображен преобразованный объект.

3-10 ОТРАЖЕНИЕ ОТНОСИТЕЛЬНО ПРОИЗВОЛЬНОЙ ПЛОСКОСТИ

Преобразования, заданные в уравнениях (3-11)–(3-13), осуществляют отражение относительно координатных плоскостей $x = 0$, $y = 0$, $z = 0$ соответственно. Часто возникает необходимость отразить объект относительно произвольной плоскости. И снова это можно сделать с помощью процедуры, объединяющей ранее определенные простые преобразования. Один из возможных методов состоит в следующем:

- перенести точку P , принадлежащую плоскости отражения, в начало системы координат;
- повернуть вектор нормали к плоскости отражения в начале координат до совпадения с осью $+z$ (см. разд. 3-9, уравнения (3-23) и (3-24)), теперь плоскость отражения будет совпадать с координатной плоскостью $z = 0$;
- применяя уже известные преобразования, отразить объект относительно координатной плоскости $z = 0$ (см. (3-11));
- чтобы получить результаты, необходимо выполнить преобразования, обратные к описанным в первых двух пунктах.

Тогда общее преобразование описывается матрицей

$$[M] = [T] [R_x] [R_y] [Rf_{lt_z}] [R_y]^{-1} [R_x]^{-1} [T]^{-1},$$

где матрицы $[T]$, $[R_x]$, $[R_y]$ задаются уравнениями (3-22)–(3-24) соответственно, Rf_{lt_z} – матрица отражения относительно плоскости $z = 0$, $(x_0, y_0, z_0) = (P_x, P_y, P_z)$ – координаты точки P на плоскости отражения; а (c_x, c_y, c_z) есть вектор нормали к плоскости отражения¹.

Пример более подробно проиллюстрирует данный метод.

Пример 3-11 Отражение

Снова рассмотрим куб с одним отсеченным углом, как это показано на рис. 3-8 а. Отразим куб относительно плоскости, содержащей треугольник CDJ .

Выбрав точку C для перемещения в начало координат и вспомнив координатные векторы для куба, получим матрицу переноса

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & -3/2 & -2 & 1 \end{bmatrix}.$$

¹ Если известно уравнение плоскости отражения, $ax + by + cz + d = 0$, то единичный вектор нормали к плоскости равен

$$[\hat{n}] = [c_x \quad c_y \quad c_z] = \frac{[a \quad b \quad c]}{\sqrt{a^2 + b^2 + c^2}}.$$

Подробности можно найти в [3-1].

Нормаль к плоскости отражения получим, используя координатные векторы C, D, J (см. [3-1]). Конкретнее, взяв векторное произведение векторов CJ и CD до переноса, получим

$$\begin{aligned} n &= ([J] - [C]) \times ([D] - [C]) = \\ &= [(3 - 3) \quad (2 - 1.5) \quad (1.5 - 2)] \times [(2.5 - 3) \quad (2 - 1.5) \quad (2 - 2)] = \\ &= [0 \quad 1/2 \quad -1/2] \times [-1/2 \quad 1/2 \quad 0] = [1/4 \quad 1/4 \quad 1/4]. \end{aligned}$$

Нормализация дает

$$\hat{n} = [1/\sqrt{3} \quad 1/\sqrt{3} \quad 1/\sqrt{3}].$$

Используя уравнения (3-19) и (3-20), получим

$$d = \sqrt{n_y^2 + n_z^2} = \sqrt{(1/\sqrt{3})^2 + (1/\sqrt{3})^2} = \sqrt{2/3}$$

и $\alpha = 45^\circ$, $\beta = 35.26^\circ$. Матрицы поворотов для совмещения нормали в точке C с осью z имеют вид (см. (3-23) и (3-24))

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad [R_y] = \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} & 0 \\ 0 & 1 & 0 & 0 \\ -1/\sqrt{3} & 0 & 2/\sqrt{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Матрицы $[R_x]^{-1}$, $[R_y]^{-1}$ и $[T]^{-1}$ можно получить подстановкой $-\alpha$, $-\beta$ и $[x_0 \quad y_0 \quad z_0] = [C]$ в (3-22)–(3-24).

Объединение $[T]$, $[R_x]$ и $[R_y]$ дает

$$[M] = [T][R_x][R_y] = \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} & 0 \\ -5/2\sqrt{6} & 1/2\sqrt{2} & -13/2\sqrt{3} & 1 \end{bmatrix}.$$

Преобразованные промежуточные точки равны

$$[X][M] = \begin{bmatrix} -0.612 & -0.354 & -0.876 & 1 \\ 0.204 & -0.354 & -0.287 & 1 \\ 0 & 0 & 0 & 1 \\ -0.612 & 0.354 & 0 & 1 \\ -1.021 & 0.354 & -0.287 & 1 \\ -0.204 & 0.354 & -1.443 & 1 \\ 0.612 & 0.354 & -0.876 & 1 \\ 0.204 & 1.061 & -0.287 & 1 \\ -0.612 & 1.061 & -0.876 & 1 \\ 0 & 0.707 & 0 & 1 \end{bmatrix}.$$

Этот промежуточный результат изображен на рис. 3-9 б. Заметим, что точка C находится в начале системы координат, а ось z направлена на нас.

Отражение относительно произвольной плоскости теперь эквивалентно отражению относительно плоскости $z = 0$. Следовательно (см. (3-11)),

$$[Rflt] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

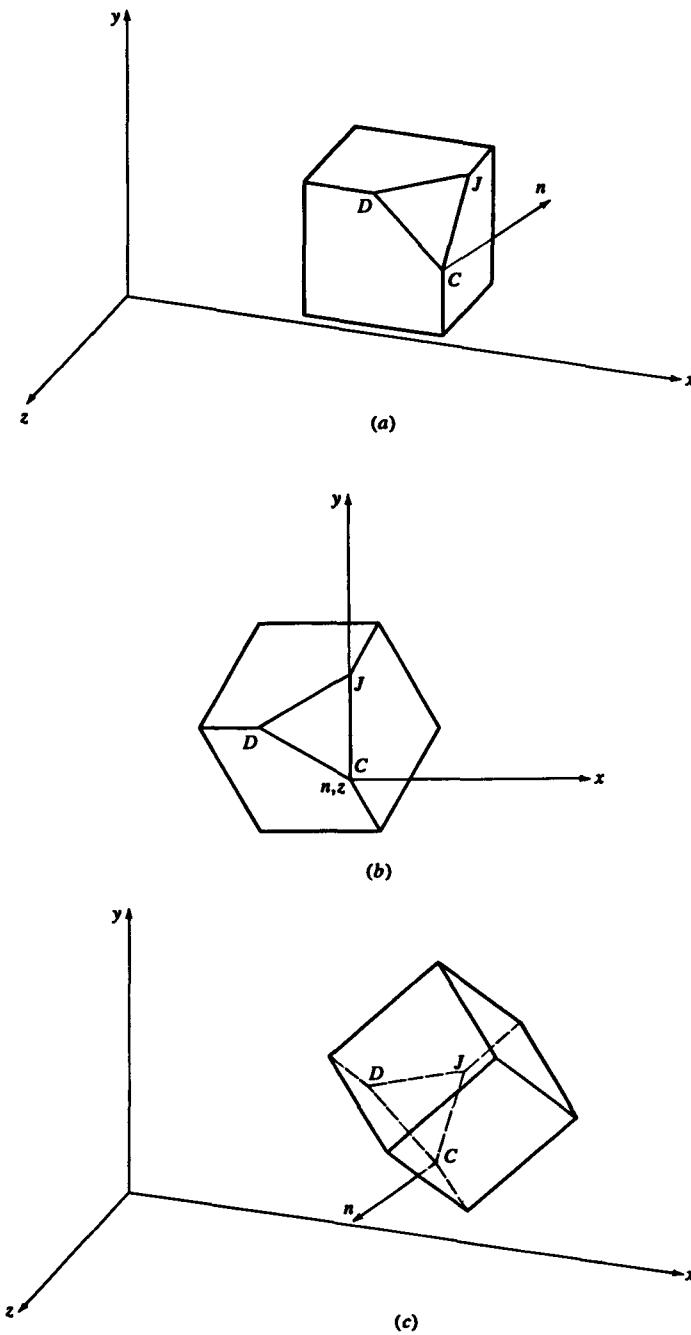


Рис. 3-9 Отражение относительно произвольной плоскости.

Для возврата преобразованного объекта в «исходное» положение в пространстве требуется преобразование

$$[M]^{-1} = [R_y]^{-1}[R_x]^{-1}[T]^{-1} = \begin{bmatrix} 2/\sqrt{6} & -1/\sqrt{6} & -1/\sqrt{6} & 0 \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 0 \\ 3 & 3/2 & 2 & 1 \end{bmatrix}.$$

Результатирующие координатные векторы таковы:

$$[X][M][Rf\!lt][M]^{-1} = \begin{bmatrix} 3 & 2 & 3 & 1 \\ 10/3 & 4/3 & 7/3 & 1 \\ 3 & 3/2 & 2 & 1 \\ 5/2 & 2 & 2 & 1 \\ 7/3 & 7/3 & 7/3 & 1 \\ 11/3 & 8/3 & 8/3 & 1 \\ 4 & 2 & 2 & 1 \\ 10/3 & 7/3 & 4/3 & 1 \\ 3 & 3 & 2 & 1 \\ 3 & 2 & 3/2 & 1 \end{bmatrix},$$

где

$$[M][Rf\!lt][M]^{-1} = \begin{bmatrix} 1/3 & -2/3 & -2/3 & 0 \\ -2/3 & 1/3 & -2/3 & 0 \\ -2/3 & -2/3 & 1/3 & 0 \\ 13/3 & 13/3 & 13/3 & 1 \end{bmatrix}.$$

На рис. 3-9 с показан преобразованный объект.

Как показано в этом и предыдущих разделах, сложные преобразования можно легко построить с помощью простых базовых преобразований. Такой подход даже предпочтительнее, так как он уменьшает вероятность появления ошибок и более эффективен с вычислительной точки зрения, нежели прямой математический подход.

3-11 АФФИННАЯ И ПЕРСПЕКТИВНАЯ (НАЧЕРТАТЕЛЬНАЯ) ГЕОМЕТРИЯ

К настоящему времени разработана математическая теория как для перспективной (начертательной), так и для аффинной геометрии. Теоремы аффинной геометрии идентичны теоремам геометрии Евклида. И в той и в другой науках важным понятием является параллелизм. В перспективной геометрии прямые в общем случае не параллельны.

Аффинное преобразование есть комбинация линейных преобразований, например поворота и последующего переноса. Для аффинного преобразования последний столбец в обобщенной 4×4 -матрице равен $[0 \ 0 \ 0 \ 1]^T$. В противном случае, как это показано далее в разделе 3-15, преобразованная однородная координата h не равна единице и нет взаимно однозначного соответствия между аффинным преобразованием и (4×4) -матричным оператором. Аффинные преобразования образуют полезное подмножество билинейных преобразований, так



Рис. 3-10 Иерархия плоских геометрических проекций.

как произведение двух аффинных преобразований также аффинно. Это свойство позволяет скомбинировать общее преобразование множества точек относительно произвольной системы координат при сохранении значения единицы для однородной координаты h .

Так как евклидова геометрия изучается в школах многие годы, то методы рисования и черчения, основывающиеся на евклидовой геометрии, стали стандартными методами графического сообщения. Хотя художниками и архитекторами часто используются перспективные виды для создания более реалистического изображения, в технической работе они используются редко из-за трудности их ручного конструирования. Однако при использовании для задания объекта однородных координат, аффинные и перспективные преобразования вычисляются одинаково легко.

И аффинные, и перспективные преобразования трехмерны, т. е. являются преобразованиями одного трехмерного пространства в другое. Однако для наблюдения результатов на двумерной поверхности требуется проецирование из трехмерного пространства в двумерное. Результат этого проецирования называется плоской геометрической проекцией. На рис. 3-10 изображена иерархия плоских геометрических проекций. Матрица проецирования из трехмерного пространства в двумерное всегда содержит столбец из нулей, следовательно, детерминант этого преобразования всегда равен нулю.

Плоские геометрические проекции объектов образуются пересечением прямых, называемых проекторами, с плоскостью, называемой плоскостью проекции. Проекторы — это прямые, проходящие через произвольную точку, называемую центром проекции, и каждую точку объекта. Если центр проекции расположен в конечной точке трехмерного пространства, получается перспективная проекция. Если центр расположен в бесконечности, то все проекторы параллельны и результат является параллельной проекцией. Плоские геометрические проекции пред-

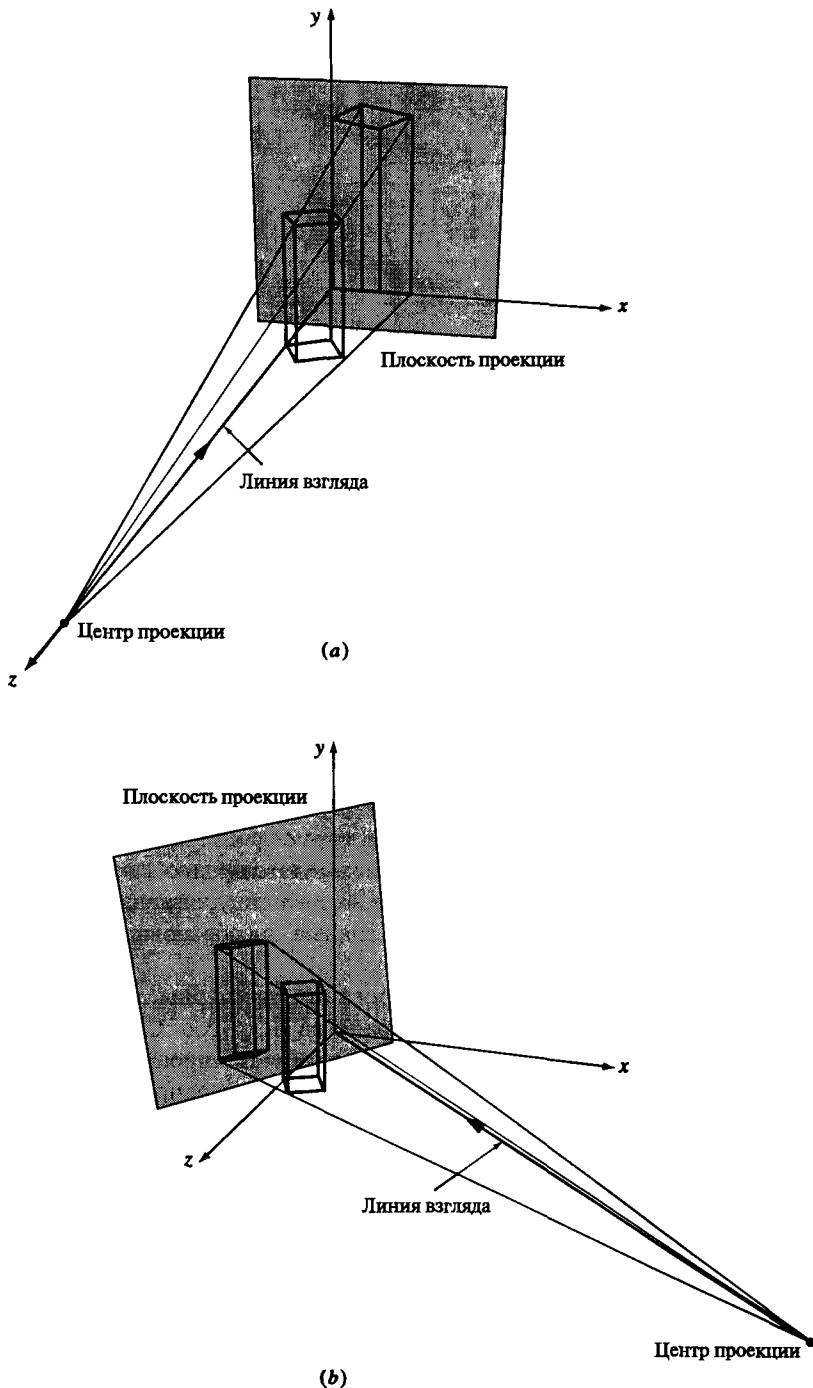


Рис. 3-11 Плоские проекции. (a) Фиксирован центр проекции; (b) фиксирован объект.

ставляют основу описательной геометрии. Неплоские и негеометрические проекции также полезны; они широко используются в картографии.

Для разработки различных преобразований, представленных на рис. 3-10, можно использовать два разных подхода. В первом предполагается, что центр проекции или точка зрения фиксирована, а плоскость проекции перпендикулярна каждому проектору, как это показано на рис. 3-11 *a*. Для получения требуемого вида манипулируют объектом. Во втором подходе предполагается, что объект фиксирован, центр проекции может как угодно перемещаться в трехмерном пространстве, а плоскость проекции не обязательно перпендикулярна направлению взгляда. На рис. 3-11 *b* приведен пример этого. Оба подхода математически эквивалентны.

Первый подход напоминает действия наблюдателя, которого попросили описать небольшой объект, например, книгу. Он берет объект в руки, поворачивает и перемещает его для того, чтобы изучить все его стороны. В этом случае центр проекции фиксирован и манипулируют объектом. Второй подход напоминает действия наблюдателя, которого попросили описать большой объект, например автомобиль. Наблюдатель ходит вокруг объекта, чтобы осмотреть его с разных сторон, взбирается на лестницу для осмотра верха и опускается на колени для осмотра его нижней части. В этом случае объект фиксирован, а центр проекции и точка зрения перемещаются.

В процессе конструирования или изображения объекта на графическом дисплее компьютера местоположение глаза обычно фиксировано, а плоскость проекции, т. е. поверхность, обычно перпендикулярна направлению взгляда. Следовательно, в данном случае более подходит первый подход. Тем не менее, если графический дисплей используется для представления движения какого-нибудь транспортного средства или наблюдатель движется в сгенерированной компьютером модели, как в случае использования тренажера, или если наблюдатель прогуливается в архитектурной модели, тогда больше подходит второй подход.

В этой книге используются фиксированный центр проекции и движущийся объект. Метод с фиксированным объектом и движущимся центром проекции хорошо проработан Калбомом и Пейсиореком ([3-2]).

Мы начнем наше обсуждение плоских геометрических проекций (см. рис. 3-10) с рассмотрения параллельных проекций.

3-12 ОРТОГРАФИЧЕСКИЕ ПРОЕКЦИИ

Самой простой из параллельных проекций является ортографическая проекция, используемая обычно в инженерных чертежах. В этом случае точно изображаются правильные или «истинные» размер и форма одной плоской грани объекта. Ортографические проекции — это проекции на одну из координатных плоскостей $x = 0$, $y = 0$ или $z = 0$. Матрица проекции на плоскость $z = 0$ имеет вид

$$[P_z] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-27)$$

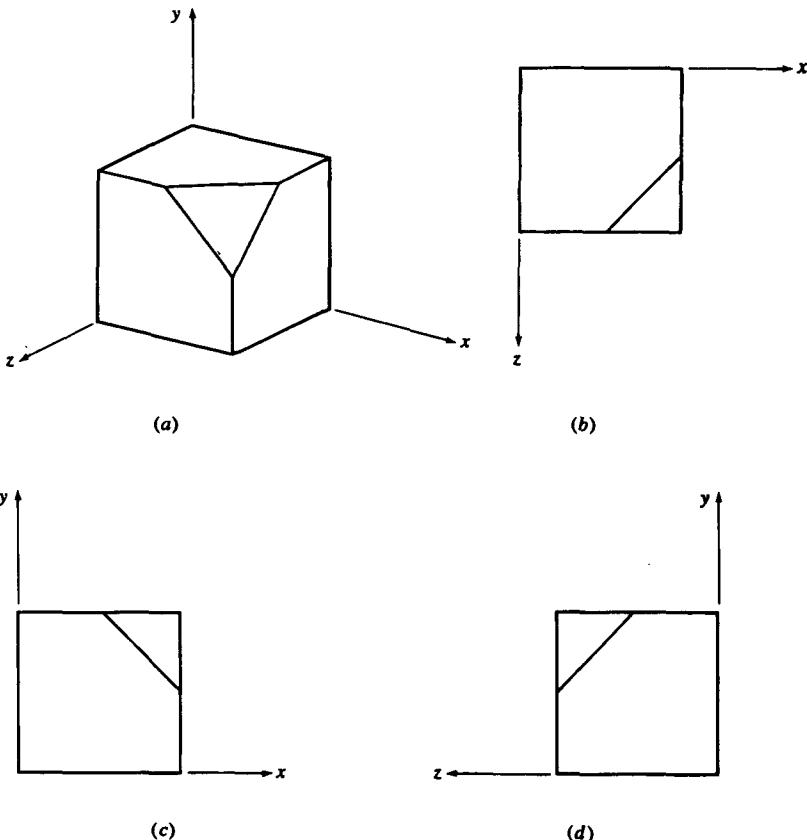


Рис. 3-12 Ортографические проекции объекта на рис. 3-12а на плоскости: (b) $y = 0$, (c) $z = 0$ и (d) $x = 0$.

Заметим, что в третьем столбце (столбце z) все элементы нулевые. Следовательно, в результате преобразования z -координата координатного вектора станет равной нулю.

Аналогичным образом матрицы для проекций на плоскости $x = 0$ и $y = 0$ равны

$$[P_z] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-28)$$

и

$$[P_y] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-29)$$

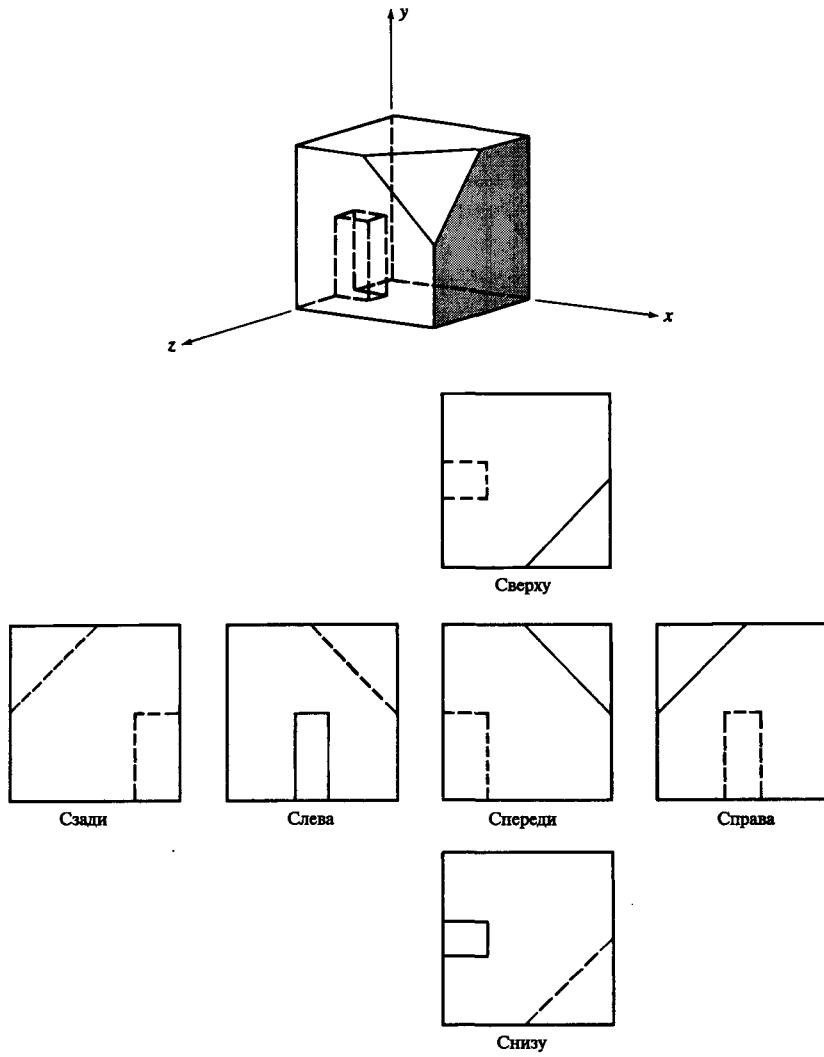


Рис. 3-13 Многовидовая ортографическая проекция.

Ортографические проекции объекта на рис. 3-12 *a* на плоскости $x = 0$, $y = 0$ и $z = 0$ из центров проекций, расположенных в бесконечности на $+x$ -, $+y$ - и $+z$ -осях соответственно, изображены на рис. 3-12 *b*, 3-12 *c* и 3-12 *d* соответственно.

Одна ортографическая проекция не дает достаточной информации для визуального и практического воссоздания формы объекта. Необходимо, следовательно, наличие нескольких ортографических проекций. Эти многовидовые ортографические проекции обычно¹ организуются так, как это показано на рис. 3-13.

¹ Это соглашение относится к Соединенным Штатам.

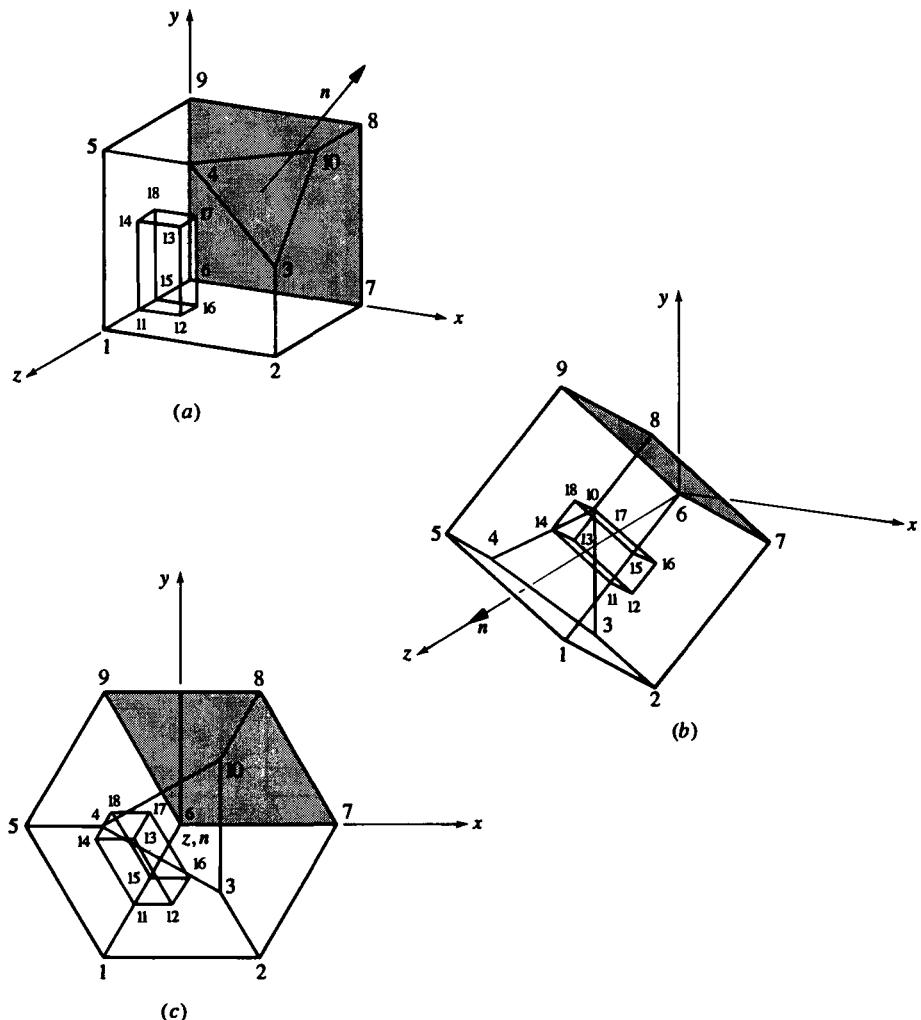


Рис. 3-14 Построение вспомогательного вида. (а) Триметрический вид; (б) нормаль совмещена с осью z ; (с) спроецировано на плоскость $z = 0$.

Виды спереди, справа и сверху получаются проецированием на плоскости $z = 0$, $x = 0$ и $y = 0$ из центров проекции, расположенных в бесконечности на $+z$, $+x$ и $+y$ осях. Виды сзади, слева и снизу получаются проецированием на плоскости $z = 0$, $x = 0$, $y = 0$ из центров проекции, расположенных в бесконечности на $-z$, $-x$ и $-y$ осях. Координатные оси обычно не изображаются на этих видах.

Как показано на рис. 3.13, невидимые линии принято изображать пунктиром. Для адекватного представления формы объекта обычно не требуются все шесть видов. Наиболее часто используются виды спереди, сверху и справа. Даже если используются не все шесть видов, оставшиеся располагаются в указанных местах.

Виды спереди и с боков иногда называют главным и боковыми фасадами, вид сверху иногда называют планом.

Интересно и важно отметить, что все шесть видов могут быть получены комбинациями отражения, вращения и переноса с последующим проецированием на плоскость $z = 0$ из центра проекции, расположенного в бесконечности на оси z . Например, вид сзади получается отражением относительно плоскости $z = 0$ и проецированием на эту плоскость. Аналогичным образом вид слева получается вращением вокруг оси y на угол $+90^\circ$ и проецированием на плоскость $z = 0$.

Для объектов с гранями, не параллельными одной из координатных плоскостей, стандартные ортографические виды не показывают правильную или истинную форму этих граней. Для этих целей используются вспомогательные виды. Вспомогательный вид образуется с помощью вращения и перемещения объекта так, чтобы нормаль к грани совпала с одной из координатных осей (см. разд. 3-9). Результат затем проецируется на координатную плоскость, перпендикулярную этой оси. На рис. 3-14 с изображена вспомогательная плоскость, показывающая истинную форму треугольного угла объекта на рис. 3.13.

Пример более полно объясняет эти построения.

Пример 3-12 Вспомогательный вид

Построим вспомогательный вид, изображающий правильную форму треугольного угла объекта на рис. 3-14 а. Координатные векторы объекта равны

$$[X] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0.5 & 1 & 1 \\ 0.5 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0.5 & 1 \\ 0 & 0 & 0.6 & 1 \\ 0.25 & 0 & 0.6 & 1 \\ 0.25 & 0.5 & 0.6 & 1 \\ 0 & 0.5 & 0.6 & 1 \\ 0 & 0 & 0.4 & 1 \\ 0.25 & 0 & 0.4 & 1 \\ 0.25 & 0.5 & 0.4 & 1 \\ 0 & 0.5 & 0.4 & 1 \end{bmatrix}.$$

Номера вершин, изображенные на рис. 3-14, соответствуют номерам строк матрицы $[X]$ координатных векторов.

Внешняя нормаль к треугольной грани имеет следующие направляющие косинусы

$$[c_x \quad c_y \quad c_z] = [1/\sqrt{3} \quad 1/\sqrt{3} \quad 1/\sqrt{3}]$$

и проходит через начало координат и точку [0.83333 0.83333 0.83333]. Вспомнив результаты разд. 3-9 и примера 3-10, получим, что нормаль совмещается с осью z с помощью поворота вокруг оси x на угол

$$\alpha = \arccos(c_z/d) = \arccos(1/\sqrt{2}) = +45^\circ$$

и последующего поворота вокруг оси y на угол

$$\beta = \arccos(d) = \arccos(2/\sqrt{6}) = +35, 26^\circ.$$

Теперь общая матрица преобразования равна

$$[T] = \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Преобразованные координатные векторы равны

$$[X'] = [X] [T] = \begin{bmatrix} -0.408 & -0.707 & 0.577 & 1 \\ 0.408 & -0.707 & 1.155 & 1 \\ 0.204 & -0.354 & 1.443 & 1 \\ -0.408 & 0 & 1.443 & 1 \\ -0.816 & 0 & 1.155 & 1 \\ 0 & 0 & 0 & 1 \\ 0.816 & 0 & 0.577 & 1 \\ 0.408 & 0.707 & 0.155 & 1 \\ -0.408 & 0.707 & 0.577 & 1 \\ 0.204 & 0.354 & 1.443 & 1 \\ -0.245 & -0.424 & 0.354 & 1 \\ -0.041 & -0.424 & 0.491 & 1 \\ -0.245 & -0.071 & 0.779 & 1 \\ -0.449 & -0.071 & 0.635 & 1 \\ -0.163 & -0.283 & 0.231 & 1 \\ 0.041 & -0.283 & 0.375 & 1 \\ -0.163 & 0.071 & 0.664 & 1 \\ -0.367 & 0.071 & 0.52 & 1 \end{bmatrix}$$

Результат изображен на рис. 3-14 б. Вспомогательный вид создается проецированием этого промежуточного результата на плоскость $z = 0$ с помощью (3-27), т. е.

$$[P_z] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Объединение матриц $[T]$ и $[P_z]$ дает

$$[T'] = [T] [P_z] = \begin{bmatrix} 2/\sqrt{6} & 0 & 0 & 0 \\ -1/\sqrt{6} & 1/\sqrt{2} & 0 & 0 \\ -1/\sqrt{3} & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Обратите внимание на столбец из нулей. Вспомогательный вид создается затем с помощью

$$[X''] = [X] [T'].$$

Матрица $[X'']$ та же самая, что и $[X']$ за исключением нулевого третьего столбца, т. е. действие проецирования состоит в отбрасывании координаты z . Результат показан на рис. 3-14 с. Невидимые линии изображены сплошными. Заметим, что показана правильная форма грани объекта — равностороннего треугольника.

Часто бывает необходимо для сложных объектов показать внутренние детали. Это выполняется с помощью сечений. Сечение образуется проведением через объект так называемой «секущей» плоскости, удалением части объекта по одну сторону от этой плоскости и проецированием оставшейся части объекта на плоскость сечения. Как и ранее, этот вид можно сделать с помощью совмещения с одной из координатных осей нормали к плоскости сечения (см. разд. 3-9), отсечения объекта с одной стороны от плоскости сечения (см. [3-1]) и, наконец, проецирования результата на координатную плоскость, перпендикулярную этой оси.

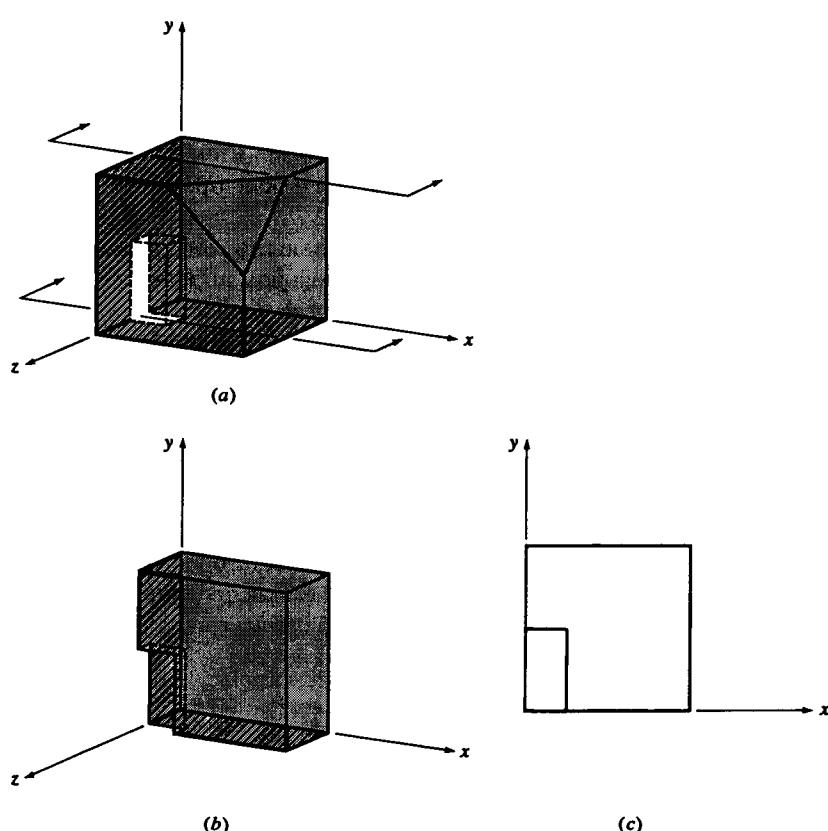


Рис. 3-15 Построение сечения. (a) Весь объект; (b) удалена часть между плоскостью сечения и центром проецирования; (c) проекция на плоскость $z = 0$.

На рис. 3-15 изображена плоскость сечения, проходящая через выемку на левой стороне объекта с рис. 3-13. Стрелки используются для показа плоскости сечения и направления взгляда.

3-13 АКСОНОМЕТРИЧЕСКИЕ ПРОЕКЦИИ

Одна ортографическая проекция не может дать представления об общей трехмерной форме объекта. Это ограничение можно преодолеть с помощью аксонометрических проекций. Аксонометрическая проекция образуется манипулированием объекта с помощью поворотов и перемещений таким образом, что бы были видны по крайней мере три соседние грани¹. Результат затем проецируется с центром проекции, расположенным в бесконечности, на одну из координатных плоскостей, обычно на плоскость $z = 0$. Если грань не параллельна плоскости проекции, то аксонометрическая проекция не показывает истинную форму этой грани. Однако остаются постоянными относительные длины параллельных в исходном пространстве линий, т. е. параллельные линии одинаково укорачиваются (искажаются). Коэффициент искажения есть отношение длины проекции отрезка к его истинной длине. Представляют интерес три аксонометрические проекции: триметрическая, диметрическая и изометрическая, как это показано на рис. 3-10. В триметрической проекции меньше всего ограничений, а в изометрической — больше всего. В самом деле, как будет показано ниже, изометрическая проекция есть частный случай диметрической, а диметрическая проекция есть частный случай триметрической.

Триметрическая проекция строится произвольными поворотами вокруг любых координатных осей, совершамыми в произвольном порядке, с последующим проецированием на плоскость $z = 0$. Большинство иллюстраций в этой книге представлены триметрическими проекциями. На рис. 3-16 изображено несколько таких проекций. Каждая из них построена с помощью поворота вокруг оси y , затем вокруг оси x и, наконец, параллельного проецирования на плоскость $z = 0$.

В общем случае для триметрической проекции коэффициенты искажения по каждой из проецируемых главных осей (x , y и z) не равны друг другу. Здесь термин главная ось используется в том смысле, что ось (или ребро) объекта в исходном пространстве параллельна одной из координатных осей x , y или z . Наложение ограничений на коэффициенты уменьшает диапазон триметрических проекций. Однако для любой конкретной триметрической проекции коэффициенты искажения вычисляют с помощью применения общей матрицы преобразования к единичным векторам вдоль главных осей. В частности,

$$[U][T] = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} [T] = \begin{bmatrix} x_x^* & y_x^* & 0 & 1 \\ x_y^* & y_y^* & 0 & 1 \\ x_z^* & y_z^* & 0 & 1 \end{bmatrix}, \quad (3-30)$$

¹ Минимальное количество граней, встречающихся для простых кубовидных объектов, подобных используемым в большинстве иллюстраций этой главы.

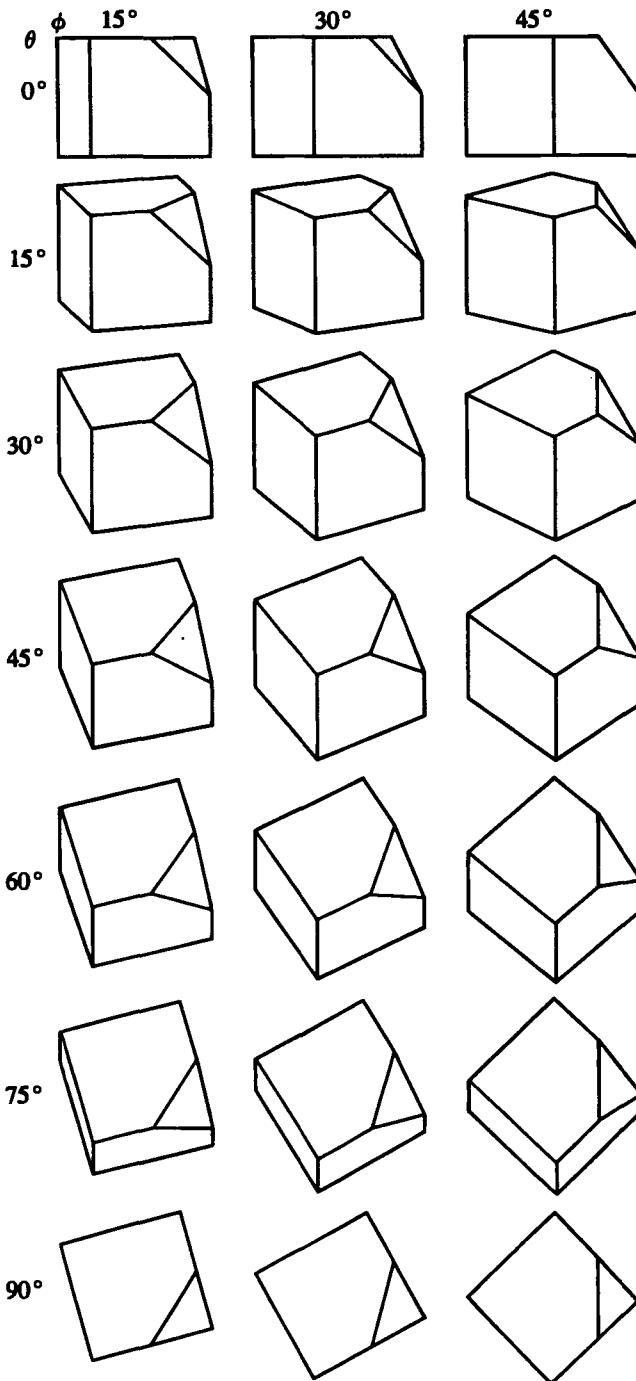


Рис. 3-16 Триметрические проекции.

где $[U]$ есть матрица единичных векторов вдоль нетрансформированных осей x , y и z соответственно, а $[T]$ — общая матрица триметрической проекции. Тогда коэффициенты искажения вдоль спроецированных главных осей равны

$$f_x = \sqrt{x_x^{*2} + y_x^{*2}}, \quad (3-31a)$$

$$f_y = \sqrt{x_y^{*2} + y_y^{*2}}, \quad (3-31b)$$

$$f_z = \sqrt{x_z^{*2} + y_z^{*2}}. \quad (3-31c)$$

В примере 3-13 триметрическая проекция рассмотрена более подробно.

Пример 3-13 Триметрическая проекция

Рассмотрим центральную иллюстрацию рис. 3-16, построенную с помощью поворота на угол $\phi = 30^\circ$ вокруг оси y , затем поворота на угол $\theta = 45^\circ$ вокруг оси x и последующего параллельного проецирования на плоскость $z = 0$. Координатный вектор куба с отсеченным углом

$$[X] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0.5 & 1 & 1 \\ 0.5 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0.5 & 1 \end{bmatrix}.$$

Общая матрица триметрической проекции равна (см. (3-8), (3-6) и (3-27)):

$$\begin{aligned} [T] &= [R_y] [R_z] [P_z] = \\ &= \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos \phi & \sin \phi \sin \theta & 0 & 0 \\ 0 & \cos \theta & 0 & 0 \\ \sin \phi & -\cos \phi \sin \theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{3}/2 & \sqrt{2}/4 & 0 & 0 \\ 0 & \sqrt{2}/2 & 0 & 0 \\ 1/2 & -\sqrt{6}/4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Таким образом, преобразованные координатные векторы имеют вид:

$$[X^*] = [X] [T] = \begin{bmatrix} 0.5 & -0.612 & 0 & 1 \\ 1.366 & -0.259 & 0 & 1 \\ 1.366 & 0.095 & 0 & 1 \\ 0.933 & 0.272 & 0 & 1 \\ 0.5 & 0.095 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.866 & 0.354 & 0 & 1 \\ 0.866 & 1.061 & 0 & 1 \\ 0 & 0.707 & 0 & 1 \\ 1.116 & 0.754 & 0 & 1 \end{bmatrix},$$

а коэффициенты искажения —

$$[U][T] = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{3}/2 & \sqrt{2}/4 & 0 & 0 \\ 0 & \sqrt{2}/2 & 0 & 0 \\ 1/2 & -\sqrt{6}/4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ = \begin{bmatrix} \sqrt{3}/2 & \sqrt{2}/4 & 0 & 1 \\ 0 & \sqrt{2}/2 & 0 & 1 \\ 1/2 & -\sqrt{6}/4 & 0 & 1 \end{bmatrix}$$

и

$$f_x = \sqrt{(\sqrt{3}/2)^2 + (\sqrt{2}/4)^2} = 0.935, \\ f_y = \sqrt{2}/2 = 0.707, \\ f_z = \sqrt{(1/2)^2 + (-\sqrt{6}/4)^2} = 0.791.$$

Диметрическая проекция — это триметрическая проекция с двумя одинаковыми коэффициентами искажения, третий коэффициент может иметь любое значение. Диметрическая проекция строится с помощью поворота на угол ϕ вокруг оси y , затем поворота на угол θ вокруг оси x и проецирования на плоскость $z = 0$ с центром проекции, расположенным в бесконечности. Точные значения углов поворотов еще не известны. Воспользовавшись (3-8), (3-6) и (3-27), получим результирующее преобразование:

$$[T] = [R_y][R_x][R_z] \\ = \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Объединение матриц дает

$$[T] = \begin{bmatrix} \cos \phi & \sin \phi \sin \theta & 0 & 0 \\ 0 & \cos \theta & 0 & 0 \\ \sin \phi & -\cos \phi \sin \theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-32)$$

Единичные векторы на главных осях x , y и z преобразуются в

$$[U^*] = [U][T] = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi \sin \theta & 0 & 0 \\ 0 & \cos \theta & 0 & 0 \\ \sin \phi & -\cos \phi \sin \theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$[U^*] = \begin{bmatrix} \cos \phi & \sin \phi \sin \theta & 0 & 1 \\ 0 & \cos \theta & 0 & 1 \\ \sin \phi & -\cos \phi \sin \theta & 0 & 1 \end{bmatrix}. \quad (3-33)$$

Теперь квадрат длины подвергнутого преобразованию единичного вектора вдоль оси x , т. е. квадрат коэффициента искажения, равен

$$f_x^2 = (x_x^{*2} + y_x^{*2}) = \cos^2 \phi + \sin^2 \phi \sin^2 \theta. \quad (3-34)$$

Аналогично квадраты коэффициентов искажения по осям y и z равны

$$f_y^2 = (x_y^{*2} + y_y^{*2}) = \cos^2 \theta, \quad (3-35)$$

$$f_z^2 = (x_z^{*2} + y_z^{*2}) = \sin^2 \phi + \cos^2 \phi \sin^2 \theta. \quad (3-36)$$

Приравнивание коэффициентов искажения по осям¹ x и y дает одно уравнение с двумя неизвестными углами поворота ϕ и θ . А именно

$$\cos^2 \phi + \sin^2 \phi \sin^2 \theta = \cos^2 \theta.$$

Используя равенства $\cos^2 \phi = 1 - \sin^2 \phi$ и $\cos^2 \theta = 1 - \sin^2 \theta$, получаем

$$\sin^2 \phi = \sin^2 \theta / (1 - \sin^2 \theta). \quad (3-37)$$

Второе соотношение между ϕ и θ получим, фиксируя коэффициент искажения f_z вдоль оси z . Объединив уравнения (3-36) и (3-37) и воспользовавшись равенством $\cos^2 \phi = 1 - \sin^2 \phi$, получаем

$$2 \sin^2 \theta - 2 \sin^4 \theta - (1 - \sin^2 \theta) f_z^2 = 0$$

или

$$2 \sin^4 \theta - (2 + f_z^2) \sin^2 \theta + f_z^2 = 0. \quad (3-38)$$

Положив $u = \sin^2 \theta$, получим решения $\sin^2 \theta = f_z^2/2$, 1. Решение $\sin^2 \theta = 1$ отбросим, так как при подстановке его в (3-37) оно дает бесконечный результат. Следовательно,

$$\theta = \arcsin(\pm f_z / \sqrt{2}). \quad (3-39)$$

Подстановка в равенство (3-37) приводит к

$$\phi = \arcsin(\pm f_z / \sqrt{2 - f_z^2}). \quad (3-40)$$

Это показывает, что диапазон коэффициентов искажения равен $0 \leq f \leq 1^2$. Заметим далее, что каждый коэффициент искажения f_z порождает четыре возможных диметрических проекции.

На рис. 3-17 показаны диметрические проекции для разных коэффициентов искажения. Для каждого коэффициента была выбрана диметрическая проекция, соответствующая положительному вращению вокруг оси y и положительному вращению вокруг x .

На рис. 3-18 показаны четыре возможных диметрических проекции для коэффициента искажения $5/8$.

Рассмотрим конкретный пример.

¹ Могут быть использованы любые оси из трех.

² Отрицательные коэффициенты искажения не имеют смысла.

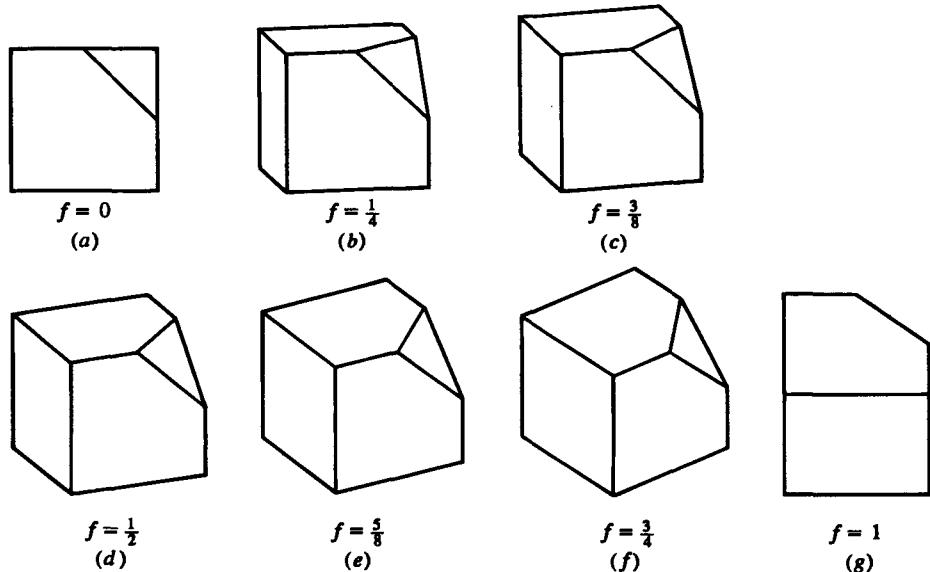


Рис. 3-17 Диметрические проекции для разных значений коэффициента искажения.
(a) 0; (b) 1/4; (c) 3/8; (d) 1/2; (e) 5/8; (f) 3/4; (g) 1.

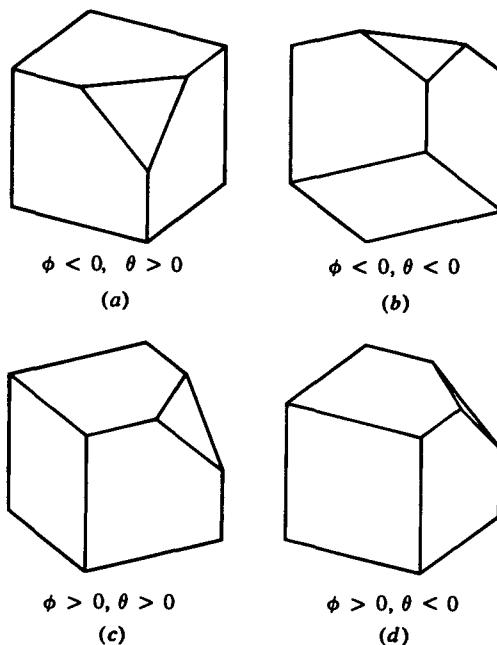


Рис. 3-18 Четыре возможных диметрических проекции для коэффициента искажения $5/8$ и углов поворотов $\phi = \pm 29.52^\circ$, $\theta = \pm 26.23^\circ$. (a) $\phi = -29.52^\circ$, $\theta = +26.23^\circ$; (b) $\phi = -29.52^\circ$, $\theta = -26.23^\circ$; (c) $\phi = +29.52^\circ$, $\theta = +26.23^\circ$; (d) $\phi = +29.52^\circ$, $\theta = -26.23^\circ$.

Пример 3-14 Диметрические проекции

Для куба с отсеченным углом построим диметрическую проекцию для коэффициента искажения по оси z , равного $1/2$.

Из равенства (3-39)

$$\begin{aligned}\theta &= \arcsin(\pm f_z / \sqrt{2}) \\ &= \arcsin(\pm 1/2\sqrt{2}) \\ &= \arcsin(\pm 0.35355) \\ &= \pm 20.705^\circ.\end{aligned}$$

Из равенства (3-40)

$$\begin{aligned}\phi &= \arcsin(\pm f_z / \sqrt{2 - f_z^2}) \\ &= \arcsin(\pm 1/2\sqrt{7/4}) \\ &= \arcsin(\pm 0.378) \\ &= \pm 22.208^\circ.\end{aligned}$$

Выбрав $\phi = +22.208^\circ$ и $\theta = +20.705^\circ$, из (3-32) получаем следующую матрицу диметрической проекции:

$$[T] = \begin{bmatrix} 0.926 & 0.134 & 0 & 0 \\ 0 & 0.935 & 0 & 0 \\ 0.378 & -0.327 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Вспоминая координатный вектор $[X]$ для куба с отсеченным углом (см. пример 3-13), получим координаты

$$[X^*] = [X][T] = \begin{bmatrix} 0.378 & -0.327 & 0 & 1 \\ 1.304 & -0.194 & 0 & 1 \\ 1.304 & 0.274 & 0 & 1 \\ 0.841 & 0.675 & 0 & 1 \\ 0.378 & 0.608 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.926 & 0.134 & 0 & 1 \\ 0.926 & 1.069 & 0 & 1 \\ 0 & 0.935 & 0 & 1 \\ 1.115 & 0.905 & 0 & 1 \end{bmatrix}.$$

Результат изображен на рис. 3-17 d.

Диметрическая проекция позволяет проводить измерения с одинаковым масштабным множителем по двум преобразованным главным осям. Измерение вдоль третьей оси требует другого масштабного множителя. Это может привести к путанице и ошибкам, если требуется точное масштабирование размеров спроектированного объекта. Изометрическая проекция решает эту проблему.

В изометрической проекции все три коэффициента искажения равны. Вспомним уравнения (3-34)–(3-36) и приравнивая (3-34) и (3-35), снова получим ра-

венство (3-37), т. е.

$$\sin^2 \phi = \frac{\sin^2 \theta}{1 - \sin^2 \theta}. \quad (3-37)$$

Приравнивание равенств (3-35) и (3-36) дает

$$\sin^2 \theta = \frac{1 - 2 \sin^2 \theta}{1 - \sin^2 \theta}. \quad (3-41)$$

Из (3-37) и (3-41) следует, что $\sin^2 \theta = 1/3$ или $\sin \theta = \pm\sqrt{1/3}$ и $\theta = \pm 35, 26^\circ$. Тогда

$$\sin^2 \phi = \frac{1}{3} : \left(1 - \frac{1}{3}\right) = \frac{1}{2}$$

$\phi = \pm 45^\circ$. Снова отметим, что существуют четыре возможных изометрических проекций. Это показано на рис. 3-19. Коэффициент искажения для изометрической проекции равен (см. (3-35))

$$f = \sqrt{\cos^2 \theta} = \sqrt{2/3} = 0.8165.$$

В самом деле, изометрическая проекция есть частный случай диметрической с $f_z = 0,8165$.

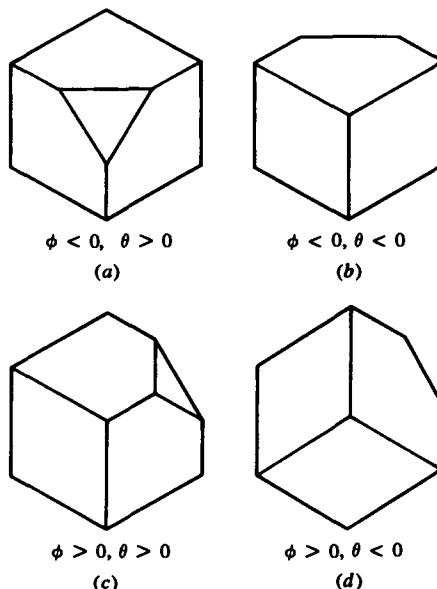


Рис. 3-19 Четыре возможных изометрических проекций с углами поворотов $\phi = \pm 45^\circ$, $\theta = \pm 35,26^\circ$. (a) $\phi = -45^\circ$, $\theta = +35,26^\circ$; (b) $\phi = -45^\circ$, $\theta = -35,26^\circ$; (c) $\phi = +45^\circ$, $\theta = +35,26^\circ$; (d) $\phi = +45^\circ$, $\theta = -35,26^\circ$.

При построении изометрических проекций вручную важен угол, который составляет проекция оси x с горизонталью. Преобразование единичного вектора вдоль оси x с помощью матрицы изометрической проекции дает

$$\begin{aligned}[U_x^*] &= [1 \ 0 \ 0 \ 1] \begin{bmatrix} \cos \phi & \sin \phi \sin \theta & 0 & 0 \\ 0 & \cos \theta & 0 & 0 \\ \sin \phi & -\cos \phi \sin \theta & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= [\cos \phi \ \sin \phi \sin \theta \ 0 \ 1].\end{aligned}$$

Тогда угол между проекцией оси x и горизонталью равен

$$\operatorname{tg} \alpha = \frac{y_x^*}{x_x^*} = \frac{\sin \phi \sin \theta}{\cos \phi} = \pm \sin \theta, \quad (3-42)$$

поскольку $\sin \phi = \cos \phi$ для $\phi = 45^\circ$. Следовательно,

$$\alpha = \operatorname{arctg}(\pm \sin 35.26439^\circ) = \pm 30^\circ.$$

Для построения изометрических проекций вручную обычно используется пластмассовый прямоугольный треугольник с углами 30° и 60° . Ниже приводится пример.

Пример 3-15 Изометрическая проекция

Снова рассмотрим куб с отсеченным углом (см. пример 3-13) и построим изометрическую проекцию для $\phi = -45^\circ$ и $\theta = +35.26439^\circ$. Из равенства (3-32) получаем, что преобразование изометрического проецирования имеет вид:

$$[T] = \begin{bmatrix} 0.707 & -0.408 & 0 & 0 \\ 0 & 0.816 & 0 & 0 \\ -0.707 & -0.408 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Вспоминая координаты $[X]$, получим

$$[X^*] = [X][T] = \begin{bmatrix} -0.707 & -0.408 & 0 & 1 \\ 0 & -0.816 & 0 & 1 \\ 0 & -0.408 & 0 & 1 \\ -0.354 & 0.204 & 0 & 1 \\ -0.707 & 0.408 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.707 & -0.408 & 0 & 1 \\ 0.707 & 0.408 & 0 & 1 \\ 0 & 0.816 & 0 & 1 \\ 0.354 & 0.204 & 0 & 1 \end{bmatrix}.$$

Результат изображен на рис. 3-19 а.

3-14 КОСОУГОЛЬНЫЕ ПРОЕКЦИИ

В противоположность ортографическим и аксонометрическим проекциям, для которых проекторы перпендикулярны плоскости проекции, косоугольная проекция формируется параллельными проекторами с центром, лежащим в бесконечности, расположенными под косым углом к плоскости проекции. Общая схема проекции изображена на рис. 3-20.

Косоугольные проекции показывают общую трехмерную форму объекта. Однако истинные размеры и форма изображаются только для граней объекта, расположенных параллельно плоскости проекции, т. е. углы и длины сохраняются только для таких граней. В самом деле, косоугольная проекция этих граней эквивалентна ортографическому виду спереди. Границы, не параллельные плоскости проекции, подвергаются искажению.

Особый интерес представляют две косоугольные проекции — кавалье и кабине. Проекция кавалье получается, когда угол между проекторами и плоскостью проекции составляет 45° . В этой проекции коэффициенты искажения для всех трех главных направлений одинаковы. Результат этой проекции выглядит неестественно утолщенным. Для «коррекции» этого недостатка используется проекция кабине.

Проекцией кабине называется такая косоугольная проекция, у которой коэффициент искажения для ребер, перпендикулярных плоскости проекции, равен $\sqrt{2}$. Как будет показано ниже, для проекции кабине угол между проекторами и плоскостью проекции составляет $\text{arcctg}(1/2) = 63.43^\circ$.

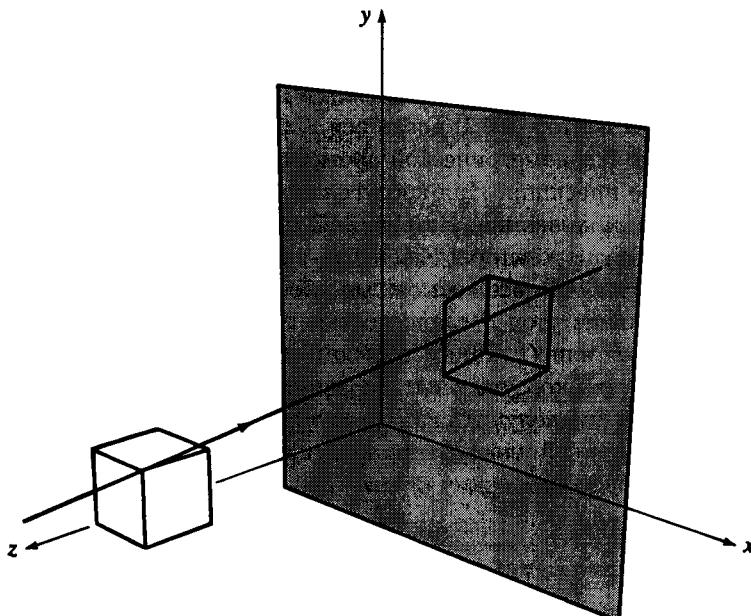


Рис. 3-20 Косоугольная проекция.

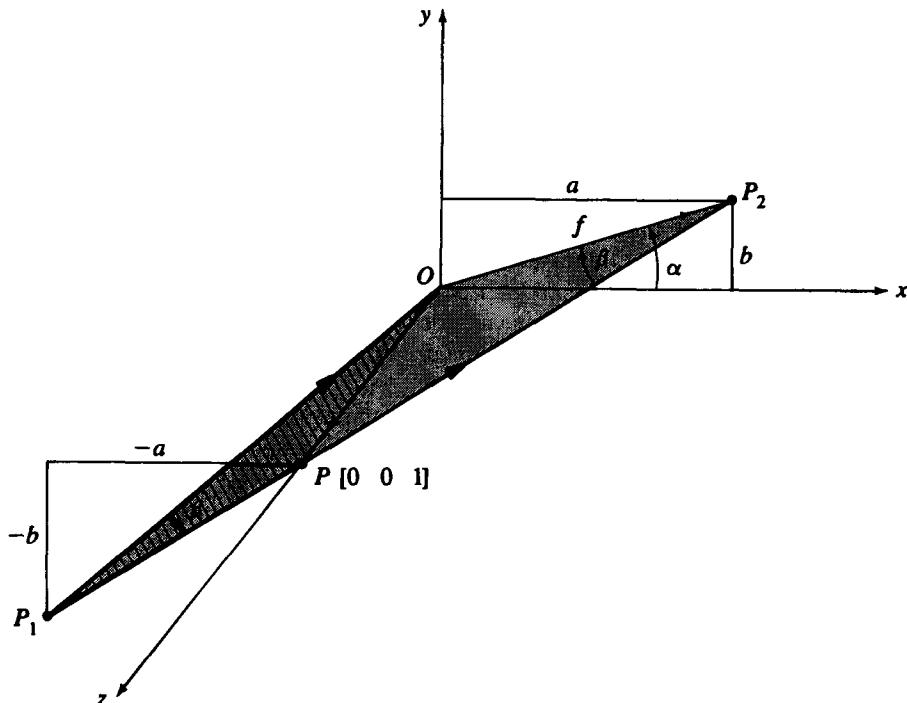


Рис. 3-21 Построение косоугольной проекции.

Чтобы построить матрицу преобразования для косоугольной проекции, рассмотрим единичный вектор $[0 \ 0 \ 1]$ вдоль оси z , показанный на рис. 3-21. Для ортографической или аксонометрической проекции на плоскость $z = 0$ вектор PO задает направление проекции. При косоугольной проекции проекторы составляют угол с плоскостью проекции. На рис. 3-21 показаны типичные косоугольные проекторы P_1O и PP_2 . Проекторы P_1O и PP_2 образуют угол β с плоскостью проекции $z = 0$. Заметим, что все возможные проекторы, проходящие через точку P или O и образующие угол β с плоскостью $z = 0$, лежат на поверхности конуса с вершиной в P или O . Таким образом, для заданного угла β существует бесконечное количество косоугольных проекций.

Проектор P_1O можно получить из PO с помощью переноса на $[-a \ -b \ 1]$ точки P в точку P_1 . В двумерной плоскости, проходящей через P перпендикулярно оси z , (3×3) -матрица преобразования равна

$$[T'] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a & -b & 1 \end{bmatrix}.$$

В трехмерном пространстве это двумерное преобразование эквивалентно сдвигу вектора PO в направлениях x и y . Для этого необходимо преобразование

$$[T''] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -a & -b & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Проецирование на плоскость $z = 0$ дает

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -a & -b & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Из рис. 3-21 получаем, что

$$\begin{aligned} a &= f \cos \alpha, \\ b &= f \sin \alpha, \end{aligned}$$

где f — длина спроцированного единичного вектора на оси z , т. е. коэффициент искажения, а α — угол между горизонталью и спроцированной осью z . Из рис. 3-21 также ясно, что β — угол между косыми проекторами и плоскостью проекции равен

$$\beta = \operatorname{arcctg}(f). \quad (3-43)$$

Таким образом, преобразование для косоугольной проекции имеет вид:

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -f \cos \alpha & -f \sin \alpha & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-44)$$

При $f = 0$, $\beta = 90^\circ$ получаем ортографическую проекцию. Если $f = 1$, то не подвергаются искажению ребра, перпендикулярные плоскости проекции. А это является условием проекции кавалье. Из равенства (3-43) имеем:

$$\beta = \operatorname{arcctg}(1) = 45^\circ.$$

Заметим, что в проекции кавалье α является все еще свободным параметром. На рис. 3-22 показаны проекции кавалье для некоторых значений α . Наиболее часто используются значения α , равные 30° и 45° . Также применяется значение $180^\circ - \alpha$.

Проекцию кабине можно получить при коэффициенте искажения $f = 1/2$. Отсюда

$$\beta = \operatorname{arcctg}(1/2) = 63.435^\circ.$$

В этом случае снова угол α является переменной величиной, как это показано на рис. 3-23. Наиболее часто встречаются значения 30° и 45° , применяется также

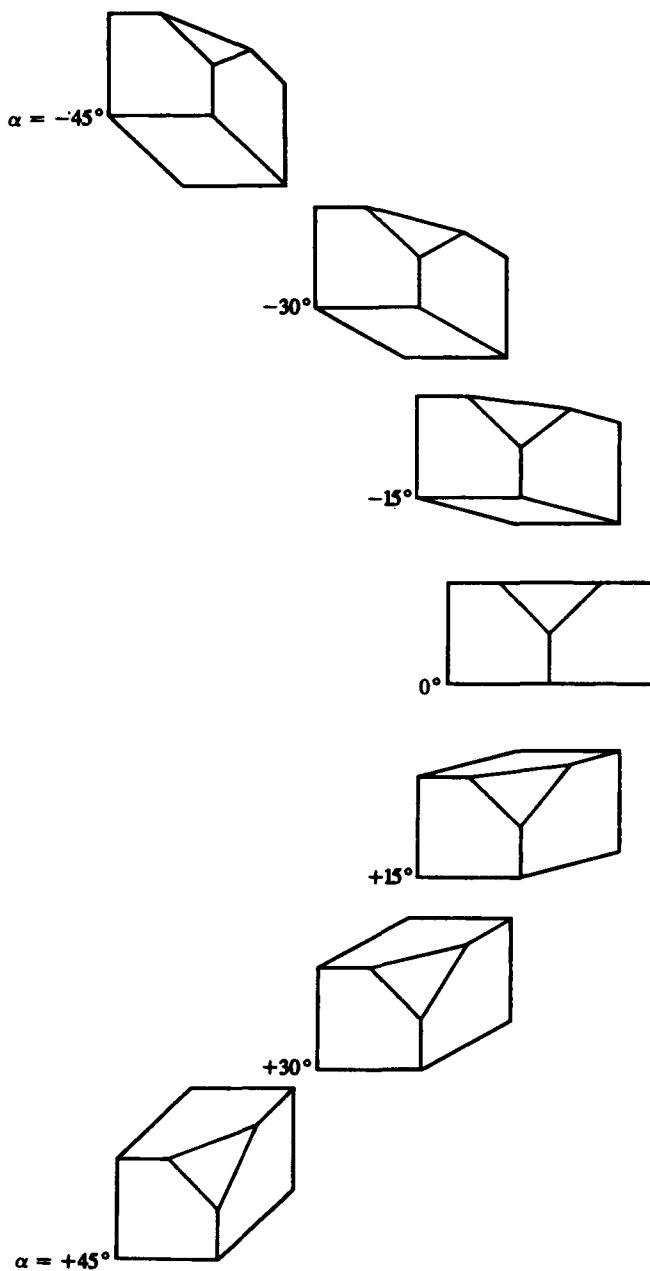


Рис. 3-22 Проекции кавалье. Сверху вниз угол α изменяется от -45° до $+45^\circ$ с интервалом 15° , угол $\beta = 45^\circ$.

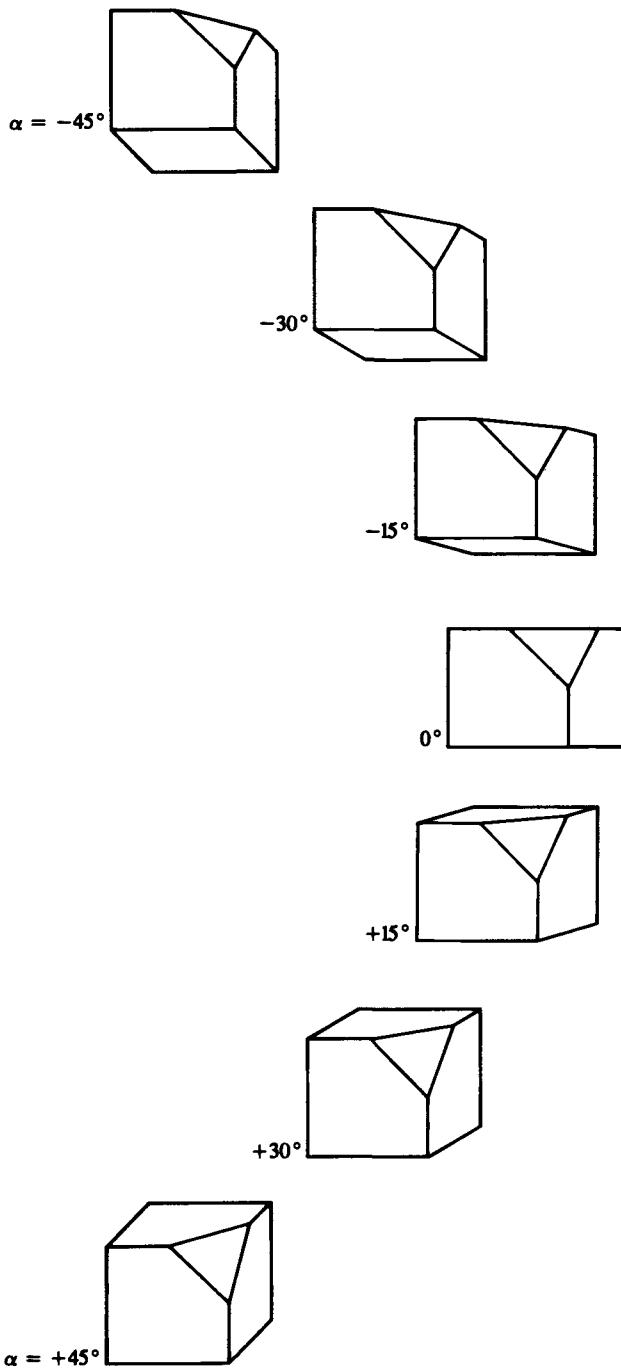


Рис. 3-23 Проекции кабине. Сверху вниз угол α изменяется от -45° до $+45^\circ$ с интервалом 15° , коэффициент искажения $f = 0.5$.

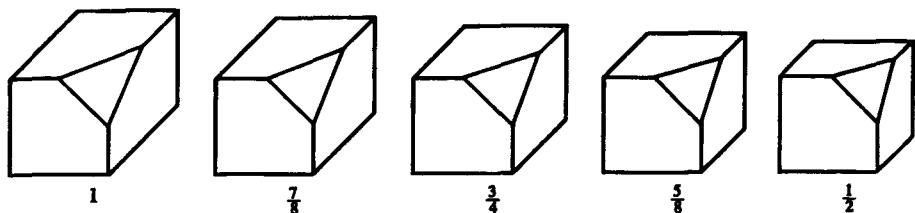


Рис. 3-24 Косоугольные проекции. Слева направо $f = 1, 7/8, 3/4, 5/8, 1/2$ при $\alpha = 45^\circ$.

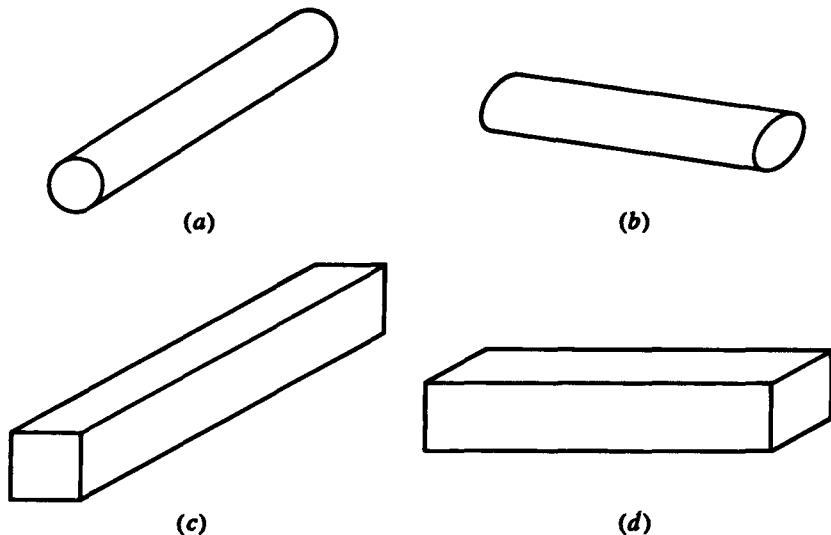


Рис. 3-25 Искажение, возникающее в косоугольных проекциях, $f = 5/8$, $\alpha = 45^\circ$.
 (a) Круглая грань параллельна плоскости проекции; (b) круглая грань перпендикулярна плоскости проекции; (c) длинная сторона перпендикулярна плоскости проекции; (d) длинная сторона параллельна плоскости проекции.

значение $180^\circ - \alpha$. На рис. 3-24 изображены косоугольные проекции для коэффициентов искажения $f = 1, 7/8, 3/4, 5/8, 1/2$ с углом $\alpha = 45^\circ$.

Поскольку изображается истинная форма одной грани, косоугольные проекции особенно подходят для иллюстрации объектов с круглыми или иными искривленными гранями. Такие грани должны быть параллельны плоскости проекции, чтобы избежать нежелательных искажений. Так же, как и в случае параллельных проекций, объекты с одним измерением, существенно превосходящим другие, подвергаются значительному искажению, если только это измерение не параллельно плоскости проекции. Такие эффекты показаны на рис. 3-25.

Ниже приводится подробный пример.

Пример 3-16 Косоугольные проекции

Построим проекции кавалье и кабине для куба с одним отсеченным углом (см. пример 3-13).

Вспомнив, что проекция кавалье – это косоугольная проекция с $\beta = 45^\circ$, т. е. коэффициент искажения $f = 1$, и выбрав угол горизонтального наклона $\alpha = 30^\circ$, из (3-44) получим матрицу преобразования

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -f \cos \alpha & -f \sin \alpha & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -0.866 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Преобразуем координаты $[X]$ куба с отсеченным углом (см. пример 3-13)

$$[X^*] = [X][T] = \begin{bmatrix} -0.866 & -0.5 & 0 & 1 \\ 0.134 & -0.5 & 0 & 1 \\ 0.134 & 0 & 0 & 1 \\ -0.366 & 0.5 & 0 & 1 \\ -0.866 & 0.5 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0.567 & 0.75 & 0 & 1 \end{bmatrix}.$$

Результат изображен на рис. 3-22.

Обратившись теперь к проекции кабине и вспомнив, что коэффициент искажения равен $1/2$, из равенства (3-43) получим

$$\beta = \operatorname{arcctg}(1/2) = \operatorname{arctg}(2) = 63.435^\circ.$$

Снова выбрав $\alpha = 30^\circ$, из (3-44), имеем:

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -f \cos \alpha & -f \sin \alpha & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -0.433 & -0.25 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Преобразованные координаты проекции кабине для этого куба равны

$$[X^*] = [X][T] = \begin{bmatrix} -0.433 & -0.25 & 0 & 1 \\ 0.567 & -0.25 & 0 & 1 \\ 0.567 & 0.25 & 0 & 1 \\ 0.067 & 0.75 & 0 & 1 \\ -0.433 & 0.75 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0.783 & 0.875 & 0 & 1 \end{bmatrix}.$$

Этот результат изображен на рис. 3-23.

Отметим, что и для проекции кавалье, и для проекции кабине треугольная грань не изображается с правильными размерами или формой, так как она не параллельна плоскости проекции ($z = 0$).

3-15 ПЕРСПЕКТИВНЫЕ ПРЕОБРАЗОВАНИЯ

Перспективное преобразование имеет место, когда не равен нулю любой из первых трех элементов четвертого столбца обобщенной (4×4) -матрицы преобразования однородных координат. Как было упомянуто ранее (см. разд. 3-11), перспективное преобразование — это преобразование одного трехмерного пространства в другое. В отличие от обсуждавшихся ранее параллельных преобразований, в данном случае параллельные прямые сходятся, размер объекта уменьшается с увеличением расстояния до центра проекции, и происходит неоднородное искажение линий объекта, зависящее от ориентации и расстояния от объекта до центра проекции. Все это помогает нашему восприятию глубины, но не сохраняет форму объекта.

Одноточечное перспективное преобразование задается равенством

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ z \ rz + 1]. \quad (3-45)$$

Здесь $h = rz + 1 \neq 1$. Обычные координаты получаются делением на h :

$$[x^* \ y^* \ z^* \ 1] = [\frac{x}{rz+1} \ \frac{y}{rz+1} \ \frac{z}{rz+1} \ 1]. \quad (3-46)$$

Перспективное проецирование на некоторую двумерную видовую плоскость можно получить, объединив ортографическую проекцию с перспективным преобразованием. Например, перспективное проецирование на плоскость $z = 0$ выполняется с помощью преобразований

$$\begin{aligned} [T] &= [P_r] [P_z] = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3-47)$$

и

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ 0 \ rz + 1]. \quad (3-48)$$

Обычные координаты равны

$$[x^* \ y^* \ z^* \ 1] = [\frac{x}{rz+1} \ \frac{y}{rz+1} \ 0 \ 1]. \quad (3-49)$$

Чтобы показать, что равенство (3-47) выполняет перспективное проецирование на плоскость $z = 0$, рассмотрим рис. 3-26 с геометрическими построениями для перспективной проекции трехмерной точки P на плоскость $z = z^* = 0$ в точку P^*

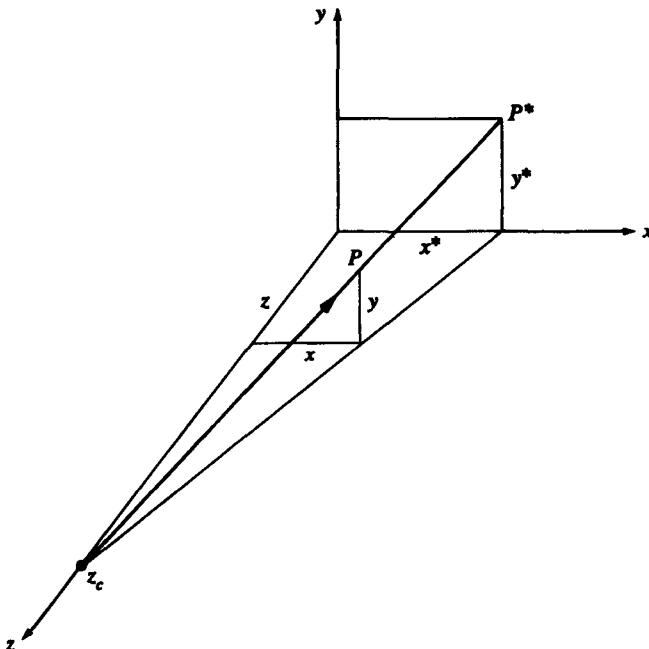


Рис. 3-26 Перспективная проекция точки.

и с центром проекции, лежащим в z_c на оси z . Координаты точки проекции P^* можно получить, используя подобие треугольников. Из рис. 3-26 следует:

$$\frac{x^*}{z_c} = \frac{x}{z_c - z}$$

или

$$x^* = x / \left(1 - \frac{z}{z_c}\right)$$

и

$$\frac{y^*}{\sqrt{x^{*2} + z_c^2}} = \frac{y}{\sqrt{x^2 + (z_c - z)^2}}$$

или

$$y^* = \frac{y}{1 - \frac{z}{z_c}},$$

z^* равно, конечно, нулю.

Полагая $r = -1/z_c$, получим результаты, аналогичные полученным с помощью (3-47). Таким образом, уравнение (3-47) осуществляет перспективное проецирование на плоскость $z = 0$ с центром проекции в точке $(-1/r)$ на оси z . Заметим,

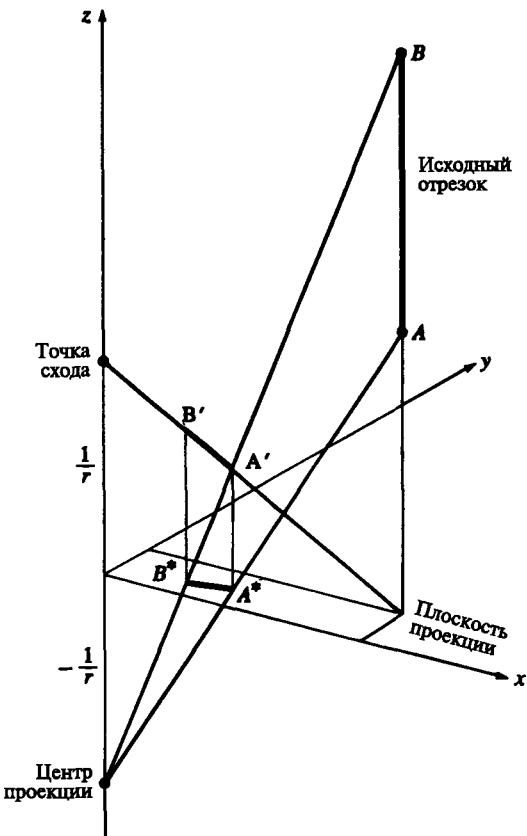


Рис. 3-27 Проекция прямой, параллельной оси z .

что при приближении z_c к бесконечности, r приближается к нулю и в результате получаем аксонометрическую проекцию на плоскость $z = 0$. Заметим далее, что на точки, лежащие в плоскости проекции, т. е. $z = 0$, перспективное преобразование не действует. Отметим также, что начало координат ($x = y = z = 0$) остается неизменным. Следовательно, если плоскость проекции ($z = 0$) проходит через объект, то эта часть объекта изображается с правильным размером и формой. Все другие части объекта искажаются.

Чтобы лучше понять действие перспективного преобразования, рассмотрим рис. 3-27. На нем показано перспективное проецирование на плоскость $z = 0$ отрезка AB , параллельного оси z , в отрезок $A'B'$ на плоскости $z = 0$ с центром проекции, расположенным в точке $-1/r$ на оси z . Преобразование можно разделить на два этапа (см. (3-47)). На первом этапе отрезок AB отображается в отрезок $A'B'$ (см. рис. 3-27). Затем с помощью ортографического проецирования отрезок $A'B'$ в трехмерном пространстве отображается в отрезок A^*B^* на плоскости $z = 0$. Центр проекции расположен в бесконечности.

Исследование рис. 3-27 показывает, что прямые $A'B'$ и AB пересекают плоскость $z = 0$ в одной и той же точке. Прямая $A'B'$ также пересекает ось z в точке $z = +1/r$. Далее, перспективное преобразование (см. (3-45) и (3-46)) отображает расположенную в бесконечности точку пересечения параллельных прямых AB и оси z в конечную точку $z = 1/r$ на оси z . Эта точка называется точкой схода¹. Заметим, что точка схода лежит на том же расстоянии от плоскости проекции, что и центр проекции, только с противоположной стороны от плоскости, например, если $z = 0$ есть плоскость проекции, а центр проекции находится в $z = -1/r$, тогда точка схода находится в $z = +1/r$.

Чтобы подтвердить это наблюдение, рассмотрим перспективное преобразование точки, находящейся в бесконечности на оси $+z$, т. е.

$$[0 \ 0 \ 1 \ 0] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = [0 \ 0 \ 1 \ r]. \quad (3-50)$$

Соответствующая ей точка $[x^* \ y^* \ z^* \ 1] = [0 \ 0 \ 1/r \ 1]$ теперь является конечной точкой на положительной оси z . Это означает, что все полу бесконечное положительное пространство ($0 \leq z \leq \infty$) отображается в ограниченную область $0 \leq z^* \leq 1/r$. Далее, все прямые, параллельные оси z , теперь проходят через точку $[0 \ 0 \ 1/r \ 1]$ – точку схода.

Прежде чем перейти к примерам, для полноты изложения укажем одноточечные перспективные преобразования с центром проекции и точкой схода, расположенными на осях x и y . Одноточечное перспективное преобразование

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ z \ (px + 1)] \quad (3-51)$$

с обычными координатами

$$[x^* \ y^* \ z^* \ 1] = \left[\frac{x}{px + 1} \quad \frac{y}{px + 1} \quad \frac{z}{px + 1} \quad 1 \right] \quad (3-52)$$

имеет центр проекции $[-1/p \ 0 \ 0 \ 1]$ и точку схода, расположенную на оси x в $[1/p \ 0 \ 0 \ 1]$.

Одноточечное перспективное преобразование

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ z \ (qy + 1)] \quad (3-53)$$

¹ Интуитивно точка схода – это такая точка «вдали», в которой параллельные прямые «кажутся» сходящимися и исчезающими. В качестве примера приведем прямолинейную, уходящую вдали железнодорожную колею.

с обычными координатами

$$\begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix} = \left[\frac{x}{qy+1} \quad \frac{y}{qy+1} \quad \frac{z}{qy+1} \quad 1 \right] \quad (3-54)$$

имеет центр проекции $[0 \ -1/q \ 0 \ 1]$ и точку схода, расположенную на оси y в $[0 \ 1/q \ 0 \ 1]$.

Пример 3-17 Перспективное преобразование прямой, параллельной оси z

Рассмотрим отрезок AB на рис. 3-27, параллельный оси z , с концевыми точками $A[3 \ 2 \ 4 \ 1]$ и $B[3 \ 2 \ 8 \ 1]$. Выполним перспективное проецирование на плоскость $z = 0$ с центром проекции в точке $z = -2$. Перспективное преобразование AB в $A'B'$ при $r = 0.5$ равно

$$\begin{aligned} A \begin{bmatrix} 3 & 2 & 4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} &= \begin{bmatrix} 3 & 2 & 4 & 3 \\ 3 & 2 & 8 & 5 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0.667 & 1.333 & 1 \\ 0.6 & 0.4 & 1.6 & 1 \end{bmatrix} \begin{bmatrix} A' \\ B' \end{bmatrix}. \end{aligned}$$

Параметрическое уравнение отрезка $A'B'$

$$P(t) = [A'] + [B' - A']t, \quad 0 \leq t \leq 1,$$

$$\text{или } P(t) = [1 \ 0.667 \ 1.333 \ 1] + [-0.4 \ -0.267 \ 0.267 \ 0]t.$$

Пересечение этого отрезка с плоскостями $x = 0$, $y = 0$ и $z = 0$ дает

$$x(t) = 0 = 1 - 0.4t \rightarrow t = 2.50$$

$$y(t) = 0 = 0.667 - 0.267t \rightarrow t = 2.50$$

$$z(t) = 0 = 1.333 + 0.267t \rightarrow t = -5.0.$$

Подставив значение $t = 2.5$ в параметрическое уравнение отрезка $A'B'$, получим

$$z(2.5) = 1.333 + (0.267)(2.5) = 2.0,$$

что является пересечением отрезка $A'B'$ с осью z в точке $z = +1/r$, точке схода. Теперь подстановка в уравнения для x - и y -компонент значения $t = -5.0$ дает пересечение с плоскостью $z = 0$, т. е.

$$x(-5.0) = 1 - (0.4)(-5.0) = 3.0$$

$$y(-5.0) = 0.667 - (0.267)(-5.0) = 2.0,$$

что совпадает с пересечением прямой AB с плоскостью $z = 0$. Проекция отрезка $A'B'$ в отрезок A^*B^* на плоскости $z = 0$ вычисляется следующим образом

$$\begin{aligned} A' \begin{bmatrix} 1 & 0.667 & 1.333 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} &= \begin{bmatrix} 1 & 0.667 & 0 & 1 \\ 0.6 & 0.4 & 0 & 1 \end{bmatrix} \begin{bmatrix} A^* \\ B^* \end{bmatrix}. \end{aligned}$$

Ниже приводится пример с простым кубом.

Пример 3-18 Одноточечное перспективное преобразование куба

Выполним перспективное проецирование на плоскость $z = 0$ единичного куба, изображенного на рис. 3-28 a, с центром проекции в точке $z_c = 10$ на оси z .

Одноточечный перспективный множитель r равен

$$r = -1/z_c = -1/10 = -0.1.$$

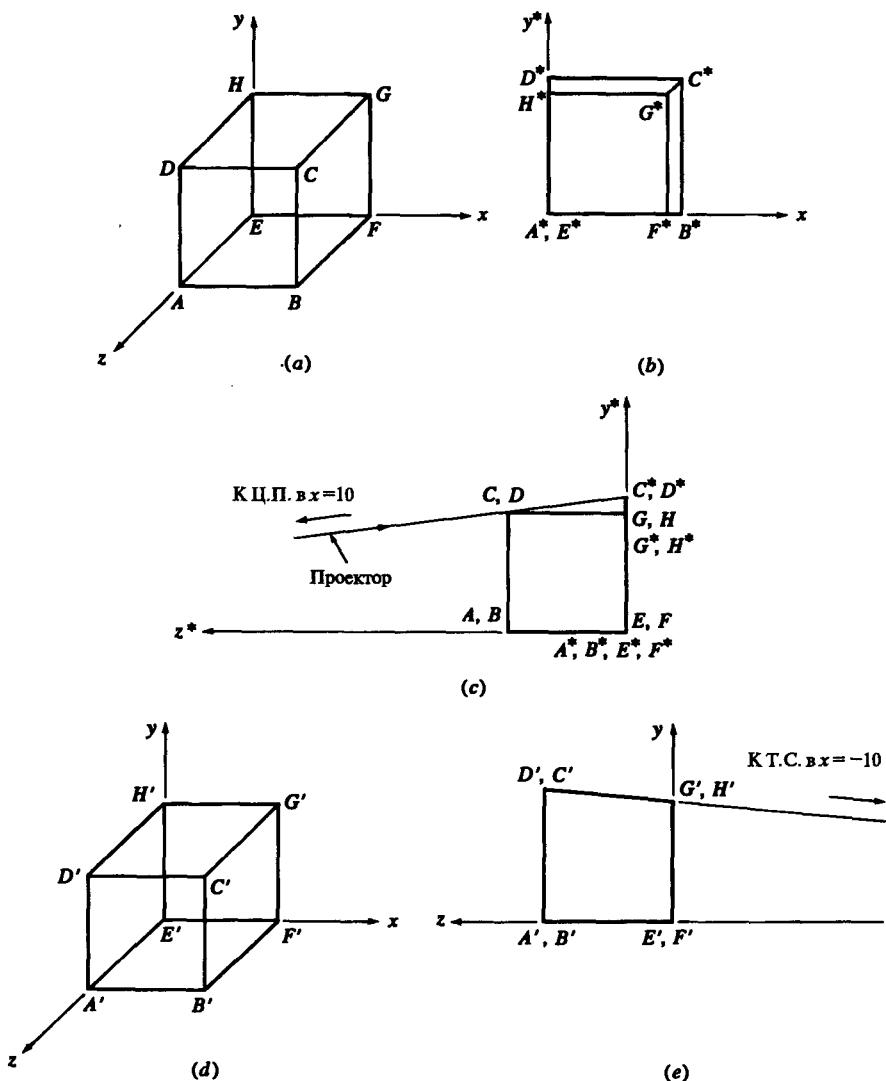


Рис. 3-28 Одноточечная перспективная проекция единичного куба.

Из уравнения (3-48) получаем, что

$$\begin{aligned}
 [T] &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 [X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 0 & 0 & 0 & 0.9 \\ 1 & 0 & 0 & 0.9 \\ 1 & 1 & 0 & 0.9 \\ 0 & 1 & 0 & 0.9 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1.11 & 0 & 0 & 1 \\ 1.11 & 1.11 & 0 & 1 \\ 0 & 1.11 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.
 \end{aligned}$$

Результат изображен на рис. 3-28 б. Отметим, что, поскольку центр проекции находится на положительной оси z , проекция передней грани $ABCD$ куба больше проекции задней грани. Почему так происходит, показано на рис. 3-28 с, на котором изображена параллельная проекция исходного куба на плоскость $x = 0$.

Отметим также, что, поскольку точка схода лежит на оси z , прямая C^*G^* на рис. 3-28 б проходит через начало координат.

В качестве альтернативного метода, эквивалентного первому, можно было бы выполнить перспективное преобразование и получить искаженный объект в трехмерном пространстве, а затем ортографически спроектировать результат на некоторую плоскость. Искаженный объект получают следующим образом:

$$\begin{aligned}
 [X'] &= [X][P_r] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 0 & 0 & 1 & 0.9 \\ 1 & 0 & 1 & 0.9 \\ 1 & 1 & 1 & 0.9 \\ 0 & 1 & 1 & 0.9 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1.11 & 1 \\ 1.11 & 0 & 1.11 & 1 \\ 1.11 & 1.11 & 1.11 & 1 \\ 0 & 1.11 & 1.11 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.
 \end{aligned}$$

Результат, полученный с помощью косоугольной проекции показан на рис. 3-28 d. Заметим, что «передняя» грань ($A'B'C'D'$) больше, чем «задняя» грань ($E'F'G'H'$). Последующее ортографическое проецирование на плоскость $z = 0$ дает тот же самый результат [X^*], что был получен ранее и который изображен на рис. 3-28 c.

На рис. 3-28 e, на котором представлена ортографическая проекция на плоскость $x = 0$ искаженного объекта с рис. 3-28 d, показано, что ребра куба, ранее параллельные оси z , теперь сходятся к точке схода $z = -10$.

Рис. 3-28 b не передает трехмерности куба. Как показано в следующем примере, более уловительный результат можно получить центрированием куба.

Пример 3-19 Одноточечное перспективное преобразование центрированного куба

Изображенный на рис. 3-28 a куб может быть центрирован на оси z путем его переноса на $-1/2$ вдоль направлений x и y . Результирующее преобразование

$$\begin{aligned}[T] &= [Tr_{xy}] [P_{rz}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.5 & -0.5 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.1 \\ -0.5 & -0.5 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Сдвинутый куб изображен на рис. 3-29 a.

Преобразованные обычные координаты равны

$$\begin{aligned}[X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.1 \\ -0.5 & -0.5 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} -0.5 & -0.5 & 0 & 0.9 \\ 0.5 & -0.5 & 0 & 0.9 \\ 0.5 & 0.5 & 0 & 0.9 \\ -0.5 & 0.5 & 0 & 0.9 \\ -0.5 & -0.5 & 0 & 1 \\ 0.5 & -0.5 & 0 & 1 \\ 0.5 & 0.5 & 0 & 1 \\ -0.5 & 0.5 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.56 & -0.56 & 0 & 1 \\ 0.56 & -0.56 & 0 & 1 \\ 0.56 & 0.56 & 0 & 1 \\ -0.56 & 0.56 & 0 & 1 \\ -0.5 & -0.5 & 0 & 1 \\ 0.5 & -0.5 & 0 & 1 \\ 0.5 & 0.5 & 0 & 1 \\ -0.5 & 0.5 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Результат изображен на рис. 3-29 b. Заметим, что ранее параллельные оси z прямые, соединяющие углы передней и задней граней, теперь сходятся к оси z ($x = 0$, $y = 0$).

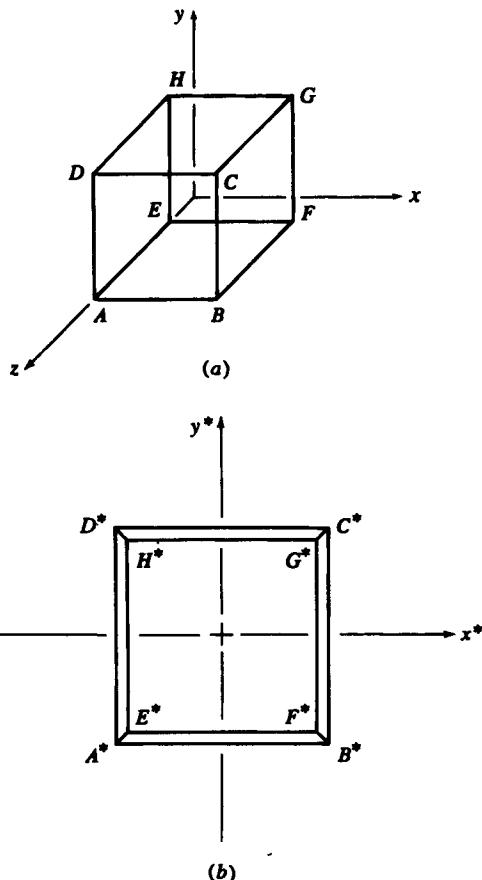


Рис. 3-29 Одноточечная перспективная проекция центрированного единичного куба.

К сожалению, результирующее изображение все еще не дает адекватного восприятия трехмерной формы объекта. Поэтому мы обратимся к более сложным перспективным преобразованиям.

Если в четвертом столбце (4×4)-матрицы преобразования два элемента из первых трех не равны нулю, то такое преобразование называется двуточечным перспективным преобразованием. Двуточечное перспективное преобразование

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ z \ (px + qy + 1)] \quad (3-55)$$

с обычными координатами

$$[x^* \ y^* \ z^* \ 1] = \left[\frac{x}{px + qy + 1} \ \frac{y}{px + qy + 1} \ \frac{z}{px + qy + 1} \ 1 \right] \quad (3-56)$$

имеет два центра проекции: первый на оси x в точке $[-1/p \ 0 \ 0 \ 1]$ и второй на оси y в точке $[0 \ -1/q \ 0 \ 1]$, и две точки схода: на оси x в точке $[1/p \ 0 \ 0 \ 1]$ и на оси y в точке $[0 \ 1/q \ 0 \ 1]$. Заметим, что заданное уравнением (3-55) двуточечное преобразование можно получить объединением двух одноточечных. Конкретнее,

$$[P_{pq}] = [P_p][P_q] = [P_q][P_p],$$

где $[P_{pq}]$ задается уравнением (3-55), $[P_p]$ — уравнением (3-53) и $[P_q]$ — уравнением (3-51). Более подробно двуточечная перспективная проекция рассмотрена в следующем примере.

Пример 3-20 Двуточечные перспективные преобразования

Снова рассмотрим куб, описанный в примере 3-18. Построим двуточечную перспективную проекцию этого куба на плоскость $z = 0$ для центров проекции, находящихся в точках $x = -10$ и $y = -10$. Нужное преобразование получим путем объединения (3-55) и (3-27). А именно,

$$\begin{aligned} [T] &= [P_{pq}][P_z] = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Здесь p и q равны

$$p = -1/(-10) = 0.1, \quad q = -1/(-10) = 0.1.$$

Преобразованные координаты куба имеют вид:

$$\begin{aligned} [X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1.1 \\ 1 & 1 & 0 & 1.2 \\ 0 & 1 & 0 & 1.1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1.1 \\ 1 & 1 & 0 & 1.2 \\ 0 & 1 & 0 & 1.1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.909 & 0 & 0 & 1 \\ 0.833 & 0.833 & 0 & 1 \\ 0 & 0.909 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.909 & 0 & 0 & 1 \\ 0.833 & 0.833 & 0 & 1 \\ 0 & 0.909 & 0 & 1 \end{bmatrix} \end{aligned}$$

Результаты изображены на рис. 3-30 а. Две точки схода находятся в $x = 10$ и $y = 10$.

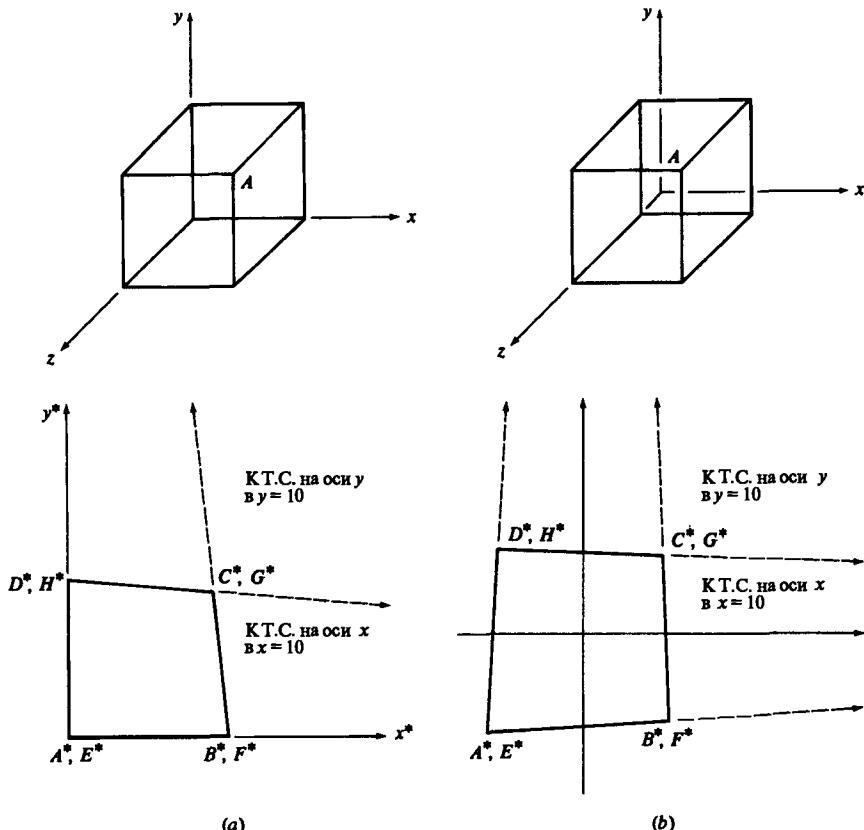


Рис. 3-30 Двуточечные перспективные проекции. (a) Нецентрированного куба; (b) центрированного куба.

Центрирование куба на оси z с помощью переноса вдоль x и y на -0.5 , аналогичное проделанному в примере 3-18, приведет к следующей общей матрице преобразования

$$\begin{aligned}
 [T] &= [T_{xy}] [P_{pq}] [P_z] = \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.5 & -0.5 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 0 & 0 \\ -0.5 & -0.5 & 0 & 0.9 \end{bmatrix},
 \end{aligned}$$

где предполагается проецирование на плоскость $z = 0$. Заметим, что в этом случае общий масштабирующий множитель больше не равен единице (см.

(3-4)), т. е. происходит очевидное масштабирование куба, вызванное его перемещением. Преобразованные координаты равны

$$\begin{aligned}
 [X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 0 & 0 \\ -0.5 & -0.5 & 0 & 0.9 \end{bmatrix} = \\
 &= \begin{bmatrix} -0.5 & -0.5 & 0 & 0.9 \\ 0.5 & -0.5 & 0 & 1 \\ 0.5 & 0.5 & 0 & 1.1 \\ -0.5 & 0.5 & 0 & 1 \\ -0.5 & -0.5 & 0 & 0.9 \\ 0.5 & -0.5 & 0 & 1 \\ 0.5 & 0.5 & 0 & 1.1 \\ -0.5 & 0.5 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -0.56 & -0.56 & 0 & 1 \\ 0.5 & -0.5 & 0 & 1 \\ 0.46 & 0.46 & 0 & 1 \\ -0.5 & 0.5 & 0 & 1 \\ -0.56 & -0.56 & 0 & 1 \\ 0.5 & -0.5 & 0 & 1 \\ 0.46 & 0.46 & 0 & 1 \\ -0.5 & 0.5 & 0 & 1 \end{bmatrix}.
 \end{aligned}$$

Результаты изображены на рис. 3-30 б.

И снова полученное изображение не дает адекватного восприятия трехмерной формы объекта. Поэтому обратимся к трехточечным перспективным преобразованиям.

Трехточечная перспектива получается, если не равны нулю три первых элемента четвертого столбца (4×4)-матрицы преобразования. Это трехточечное перспективное преобразование

$$[x \ y \ z \ 1] = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ z \ (px + qy + rz + 1)] \quad (3-57)$$

с обычными координатами

$$[x^* \ y^* \ z^* \ 1] = \left[\frac{x}{px + qy + rz + 1} \quad \frac{y}{px + qy + rz + 1} \quad \frac{z}{px + qy + rz + 1} \quad 1 \right] \quad (3-58)$$

имеет три центра проекции: на оси x в точке $[-1/p \quad 0 \quad 0 \quad 1]$, на оси y в $[0 \quad -1/q \quad 0 \quad 1]$ и на оси z в $[0 \quad 0 \quad -1/r \quad 1]$, а также три точки схода: на оси x в $[1/p \quad 0 \quad 0 \quad 1]$, на оси y в $[0 \quad 1/q \quad 0 \quad 1]$ и на оси z в $[0 \quad 0 \quad 1/r \quad 1]$.

Снова заметим, что трехточечное перспективное преобразование, заданное равенством (3-57), может быть получено конкатенацией трех одноточечных перспективных преобразований, по одному на каждую координатную ось. Создание трехточечной перспективы иллюстрируется на следующем примере.

Пример 3-21 Трехточечное перспективное преобразование

Для описанного в примере 3-18 куба рассмотрим проекцию на плоскость $z = 0$ после применения трехточечного перспективного преобразования. Центры проекции находятся в точках $x = -10, y = -10$ и $z = 10$. Точки схода находятся в $x = 10, y = 10$ и $z = -10$. Матрица преобразования равна

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Преобразованные координаты куба

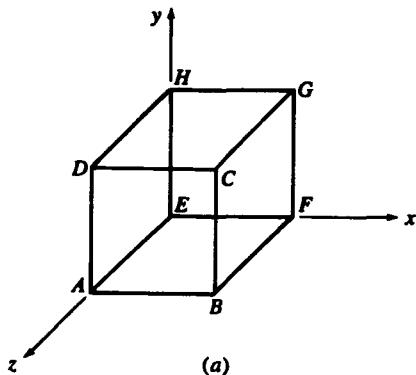
$$\begin{aligned} [X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 0 & 0 & 0 & 0.9 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1.1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1.1 \\ 1 & 1 & 0 & 1.2 \\ 0 & 1 & 0 & 1.1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0.909 & 0.909 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.909 & 0 & 0 & 1 \\ 0.833 & 0.833 & 0 & 1 \\ 0 & 0.909 & 0 & 1 \end{bmatrix} \end{aligned}$$

Результат показан на рис. 3-31 б. Искаженный в результате перспективного преобразования объект изображен на рис. 3-31 с. Отметим, что ребра на рисунке сходятся.

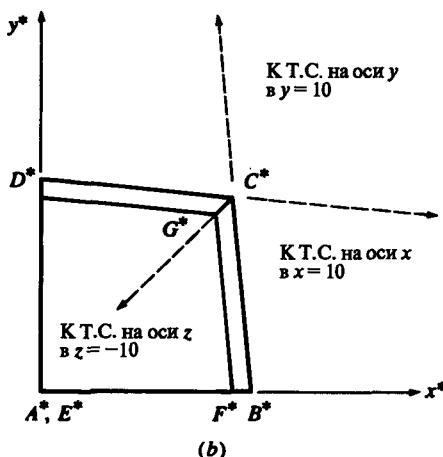
И снова получившийся вид недостаточно информативен, хотя и математически корректен. Соответствующие методы генерации перспективных видов обсуждаются в разд. 3-16.

3-16 МЕТОДЫ СОЗДАНИЯ ПЕРСПЕКТИВНЫХ ВИДОВ

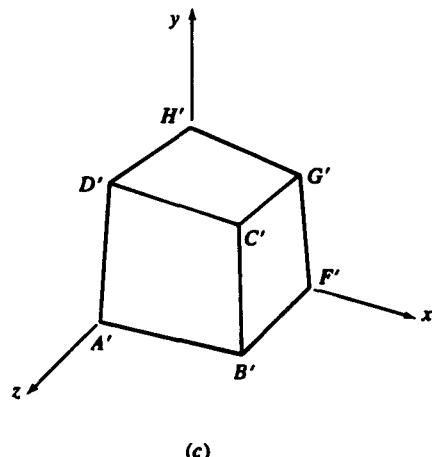
Предложенные в предыдущем разделе виды перспективной проекции были неинформативны, так как во всех случаях из каждого центра проекции была видна только одна грань куба. Для того чтобы наблюдатель воспринял трехмерную форму объекта на основании только одного вида, надо, чтобы были видны несколько граней этого объекта. Для простых объектов, подобных кубу, должны быть видны как минимум три грани. Вид с несколькими гранями можно получить из одноточечной перспективной проекции с фиксированным центром и с плоскостью проецирования, перпендикулярной направлению взгляда, если предварительно выполнен перенос и/или поворот объекта. Тогда получается реалистический вид, если только центр проекции не находится слишком близко к объекту.



(a)



(b)



(c)

Рис. 3-31 Трехточечная перспектива. (a) Исходный куб; (b) перспективная проекция на плоскость $z = 0$; (c) искаженный куб.

Для начала рассмотрим простой перенос объекта с последующим одноточечным проецированием на плоскость $z = 0$ и с центром проекции в точке $z = z_c$. Требуемое преобразование записывается в виде

$$\begin{aligned}
 [T] &= [Tr_{xyz}] [P_{rz}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ l & m & 0 & 1 + rn \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1/z_c \\ l & m & 0 & 1 - n/z_c \end{bmatrix}, \tag{3-59}
 \end{aligned}$$

где $r = -1/z_c$.

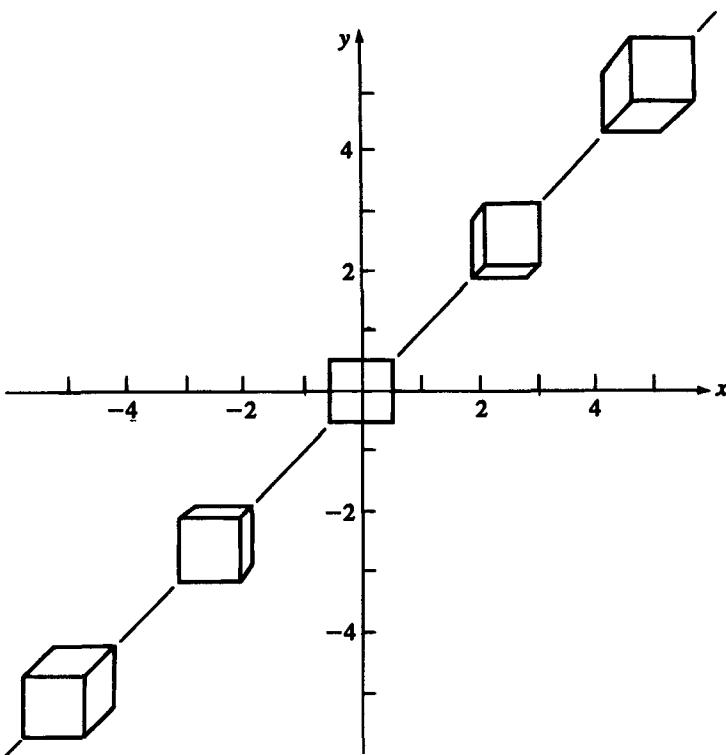


Рис. 3-32 Одноточечная перспективная проекция с переносом в x , y направлениях.

Уравнение (3-59) вместе с рис. 3-32 показывает, что перенос в направлениях x и y открывает дополнительные грани объекта. Перенос в обоих этих направлениях необходим, чтобы открыть три грани простого кубообразного объекта. На рис. 3-32 показаны результаты переноса вдоль прямой $y = x$ отцентрированного относительно начала координат куба и одноточечного проецирования на плоскость $z = 0$. Заметим, что для передней грани показываются истинные размеры и форма.

Уравнение (3-59) также показывает, что перенос вдоль оси z , т. е. к центру проекции или от него, приводит к явному изменению масштаба (из-за элемента $1 - n/z_c$). Этот эффект соответствует физической реальности, так как объекты, находящиеся дальше от наблюдателя, выглядят более мелкими. Заметим, что при приближении центра проекции к бесконечности явление масштабирования исчезает. Этот эффект схематично показан на рис. 3-33. Как изображено на этом рисунке, объект может находиться с любой стороны от центра проекции. Если объект и плоскость проекции находятся по одну сторону от центра, то, как показано на рис. 3-33, получается прямое изображение. Если же объект и плоскость проекции лежат по разные стороны от центра, то получается перевернутое изображение.

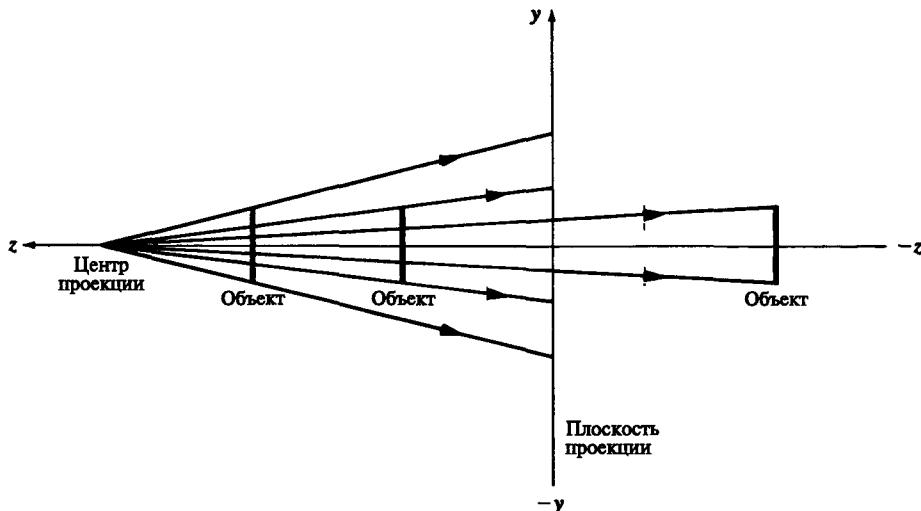


Рис. 3-33 Эффект масштабирования при перемещениях вдоль оси z для ОдноТочечной перспективной проекции.

На рис. 3-34 показаны результаты перемещения объекта во всех трех направлениях. Здесь куб перемещается вдоль трехмерной прямой от $-x = -y = -z$ к $x = y = z$. Заметно очевидное увеличение размера, а также на всех видах заметно сохранение истинной формы, но не размера передней грани.

Эти идеи более подробно изложены в примере.

Пример 3-22 Одноточечная перспективная проекция с переносом

Рассмотрим отцентрированный относительно начала координат единичный куб со следующими координатными векторами

$$[X] = \begin{bmatrix} -0.5 & -0.5 & 0.5 & 1 \\ 0.5 & -0.5 & 0.5 & 1 \\ 0.5 & 0.5 & 0.5 & 1 \\ -0.5 & 0.5 & 0.5 & 1 \\ -0.5 & -0.5 & -0.5 & 1 \\ 0.5 & -0.5 & -0.5 & 1 \\ 0.5 & 0.5 & -0.5 & 1 \\ -0.5 & 0.5 & -0.5 & 1 \end{bmatrix}.$$

Переместим куб на 5 единиц в направлениях x и y и построим перспективную проекцию на плоскость $z = 0$ с центром проекции в $z = z_c = 10$.

Из уравнения (3-59) получаем общую матрицу преобразования

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.1 \\ 5 & 5 & 0 & 1 \end{bmatrix}.$$

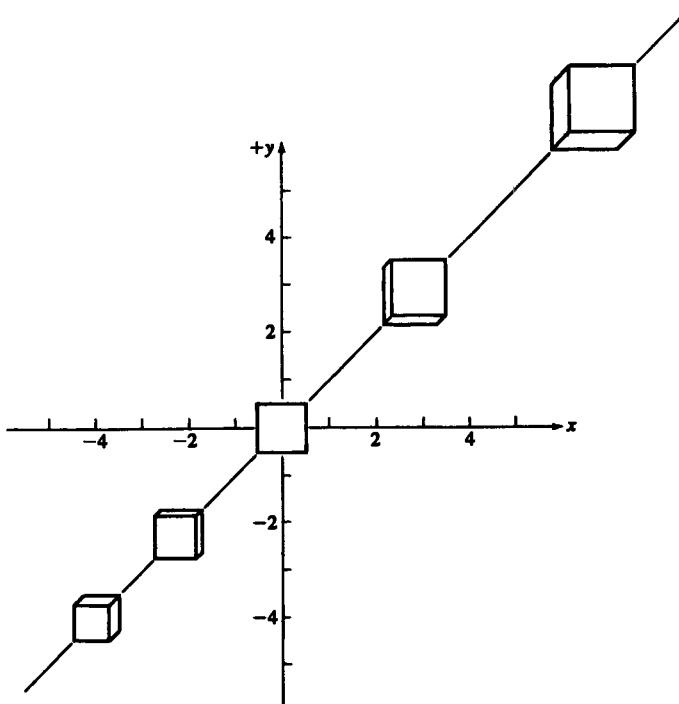


Рис. 3-34 Одноточечная перспективная проекция, объединенная с переносами в x , y , z направлениях.

Преобразованные координаты

$$[X^*] = [X][T] = \begin{bmatrix} 4.5 & 4.5 & 0 & 0.95 \\ 5.5 & 4.5 & 0 & 0.95 \\ 5.5 & 5.5 & 0 & 0.95 \\ 4.5 & 5.5 & 0 & 0.95 \\ 4.5 & 4.5 & 0 & 1.05 \\ 5.5 & 4.5 & 0 & 1.05 \\ 5.5 & 5.5 & 0 & 1.05 \\ 4.5 & 5.5 & 0 & 1.05 \end{bmatrix} = \begin{bmatrix} 4.737 & 4.737 & 0 & 1 \\ 5.789 & 4.737 & 0 & 1 \\ 5.789 & 5.789 & 0 & 1 \\ 4.737 & 5.789 & 0 & 1 \\ 4.286 & 4.286 & 0 & 1 \\ 5.238 & 4.286 & 0 & 1 \\ 5.238 & 5.238 & 0 & 1 \\ 4.286 & 5.238 & 0 & 1 \end{bmatrix}.$$

Правый верхний объект на рис. 3-32 изображает этот результат.

Если исходный объект был перемещен на 5 единиц в направлениях x , y , z и была построена одноточечная перспективная проекция на плоскость $z = 0$ с центром проекции в $z = z_c = 20$, тогда из (3-59) следует, что общая матрица преобразования записывается в виде

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.05 \\ 5 & 5 & 0 & 0.75 \end{bmatrix}.$$

Отметим общее масштабирование, задаваемое значением 0.75 в правом нижнем элементе матрицы преобразования.

Преобразованные координаты равны

$$[X^*] = [X][T] = \begin{bmatrix} 4.5 & 4.5 & 0 & 0.725 \\ 5.5 & 4.5 & 0 & 0.725 \\ 5.5 & 5.5 & 0 & 0.725 \\ 4.5 & 5.5 & 0 & 0.725 \\ 4.5 & 4.5 & 0 & 0.775 \\ 5.5 & 4.5 & 0 & 0.775 \\ 5.5 & 5.5 & 0 & 0.775 \\ 4.5 & 5.5 & 0 & 0.775 \end{bmatrix} = \begin{bmatrix} 6.207 & 6.207 & 0 & 1 \\ 7.586 & 6.207 & 0 & 1 \\ 7.586 & 7.586 & 0 & 1 \\ 6.207 & 7.586 & 0 & 1 \\ 5.806 & 5.806 & 0 & 1 \\ 7.097 & 5.806 & 0 & 1 \\ 7.097 & 7.097 & 0 & 1 \\ 5.806 & 7.097 & 0 & 1 \end{bmatrix}.$$

Результат показан в виде верхнего правого объекта на рис. 3-34.

Несколько граней также будет видно, если использовать вращение объекта. Один поворот откроет по крайней мере две грани объекта, тогда как два и более поворотов вокруг разных осей откроют, как минимум, три грани.

Матрица преобразования для поворота вокруг оси y на угол ϕ и последующего одноточечного перспективного проецирования на плоскость $z = 0$ с центром проекции в $z = z_c$:

$$\begin{aligned} [T] = [R_y][P_{rz}] &= \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos \phi & 0 & 0 & \frac{\sin \phi}{z_c} \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & 0 & -\frac{\cos \phi}{z_c} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3-60)$$

Аналогичным образом матрица преобразования для поворота вокруг оси x на угол θ и последующего одноточечного перспективного проецирования на плоскость $z = 0$ с центром проекции в точке $z = z_c$ имеет вид:

$$\begin{aligned} [T] = [R_x][P_{rz}] &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & 0 & -\frac{\sin \theta}{z_c} \\ 0 & -\sin \theta & 0 & -\frac{\cos \theta}{z_c} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3-61)$$

В обоих уравнениях (3-60) и (3-61) не равны нулю два отвечающих за перспективное преобразование (перспективных) элемента в четвертом столбце матрицы преобразования. Таким образом, один поворот вокруг главной оси, перпендикулярной той оси, на которой лежит центр проекции, эквивалентен двуточечному

перспективному преобразованию. При повороте вокруг оси, на которой лежит центр проекции, такого эффекта нет. Заметим, что для одного поворота перспективный элемент для оси вращения остается неизменным, например, в уравнениях (3-60) и (3-61) элементы q и r соответственно равны нулю.

В общем случае вращение вокруг главной оси не открывает необходимого для адекватного трехмерного представления числа граней — как минимум, трех. Для этого оно должно быть скомбинировано с перемещением вдоль оси. Следующий пример иллюстрирует это.

Пример 3-23 Двуточечное перспективное проецирование с использованием поворота вокруг одной главной оси

Рассмотрим проекцию на плоскость $z = 0$ с центром в точке $z = z_c = 2.5$ куба с рис. 3-35 а, повернутого вокруг оси y на угол $\phi = 60^\circ$, чтобы открылась левая грань, и перемещенного на -2 единицы вдоль y , чтобы открылась верхняя грань.

Используя уравнение (3-38) с $\phi = 60^\circ$, уравнение (3-47) с $z_c = 2.5$ и уравнение (3-14) с $n = 1 = 0$, $m = -2$, получим

$$\begin{aligned} [T] &= [R_y][Tr][P_{rz}] = \\ &= \begin{bmatrix} 0.5 & 0 & -0.866 & 0 \\ 0 & 1 & 0 & 0 \\ 0.866 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.4 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 0.5 & 0 & 0 & 0.346 \\ 0 & 1 & 0 & 0 \\ 0.866 & 0 & 0 & -0.2 \\ 0 & -2 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Преобразованные координаты равны

$$\begin{aligned} [X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0 & 0.346 \\ 0 & 1 & 0 & 0 \\ 0.866 & 0 & 0 & -0.2 \\ 0 & -2 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 0.866 & -2 & 0 & 0.8 \\ 1.366 & -2 & 0 & 1.146 \\ 1.366 & -1 & 0 & 1.146 \\ 0.866 & -1 & 0 & 0.8 \\ 0 & -2 & 0 & 1 \\ 0.5 & -2 & 0 & 1.346 \\ 0.5 & -1 & 0 & 1.346 \\ 0 & -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1.083 & -2.5 & 0 & 1 \\ 1.192 & -1.745 & 0 & 1 \\ 1.192 & -0.872 & 0 & 1 \\ 1.083 & -1.25 & 0 & 1 \\ 0 & -2 & 0 & 1 \\ 0.371 & -0.485 & 0 & 1 \\ 0.371 & -0.743 & 0 & 1 \\ 0 & -1 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Результат показан на рис. 3-35 б. Искажение появляется из-за того, что центр проекции расположен слишком близко к кубу. Отметим схождение параллельных осей x и z прямых линий к точкам схода, лежащим на оси x . Эти точки схода определяются в примере 3-25 из разд. 3-17.

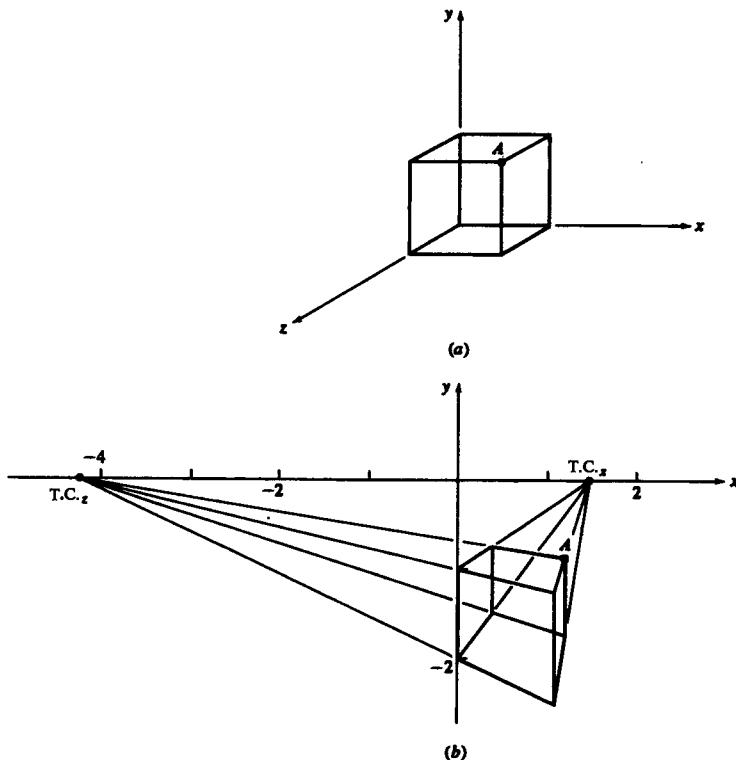


Рис. 3-35 Двухточечная перспективная проекция с поворотом вокруг одной оси.

Аналогичным образом трехточечное перспективное преобразование выполняется с помощью вращения вокруг двух или более главных осей и последующего одноточечного перспективного преобразования. Например, поворот вокруг оси y , потом поворот вокруг оси x и перспективное проецирование на плоскость $z = 0$ с центром проекции в точке $z = z_c$ имеет следующую матрицу преобразования

$$\begin{aligned}
 [T] &= [R_y][R_x][P_{rz}] = \\
 &= \begin{bmatrix} \cos \phi & 0 & -\sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1/z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} \cos \phi & \sin \phi \sin \theta & 0 & \frac{\sin \phi \cos \theta}{z_c} \\ 0 & \cos \theta & 0 & -\frac{\sin \theta}{z_c} \\ \sin \phi & -\cos \phi \sin \theta & 0 & -\frac{\cos \phi \cos \theta}{z_c} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3-62}
 \end{aligned}$$

Отметим три иенуловых перспективных элемента. Объект можно также переместить, если перемещение происходит после вращения, тогда результирующая матрица преобразования равна

$$\begin{aligned}
 [T] &= [R_y][R_x][Tr][P_{rz}] = \\
 &= \begin{bmatrix} \cos\phi & \sin\phi \sin\theta & 0 & \frac{\sin\phi \cos\theta}{z_c} \\ 0 & \cos\theta & 0 & -\frac{z_c}{\sin\theta} \\ \sin\phi & -\cos\phi \sin\theta & 0 & -\frac{z_c \cos\theta}{\cos\phi} \\ l & m & 0 & 1 - \frac{z_n}{z_c} \end{bmatrix}. \tag{3-63}
 \end{aligned}$$

Отметим здесь очевидный масштабирующий эффект перемещения вдоль z . Результаты преобразования будут различными, если поменять порядок выполнения поворотов или перенос выполнять до вращения.

Пример 3-24 Трехточечная перспективная проекция с поворотом вокруг двух осей

Рассмотрим куб на рис. 3-35 а, повернутый вокруг оси y на угол $\phi = -30^\circ$, вокруг оси x на $\theta = 45^\circ$ и спроектированный на плоскость $z = 0$ с центром в точке $z = z_c = 2.5$.

Из уравнения (3-62)

$$[T] = \begin{bmatrix} 0.866 & -0.354 & 0 & -0.141 \\ 0 & 0.707 & 0 & -0.283 \\ -0.5 & -0.612 & 0 & -0.245 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Преобразованные координаты равны

$$\begin{aligned}
 [X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & -0.354 & 0 & -0.141 \\ 0 & 0.707 & 0 & -0.283 \\ -0.5 & -0.612 & 0 & -0.245 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} -0.5 & -0.612 & 0 & 0.755 \\ 0.366 & -0.966 & 0 & 0.614 \\ 0.366 & -0.259 & 0 & 0.331 \\ -0.5 & 0.095 & 0 & 0.472 \\ 0 & 0 & 0 & 1 \\ 0.866 & -0.354 & 0 & 0.859 \\ 0.866 & 0.354 & 0 & 0.576 \\ 0 & 0.707 & 0 & 0.717 \end{bmatrix} = \begin{bmatrix} -0.662 & -0.811 & 0 & 1 \\ 0.596 & -1.574 & 0 & 1 \\ 1.107 & -0.782 & 0 & 1 \\ -1.059 & 0.201 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1.009 & -0.412 & 0 & 1 \\ 1.504 & 0.614 & 0 & 1 \\ 0 & 0.986 & 0 & 1 \end{bmatrix}.
 \end{aligned}$$

Результат изображен на рис. 3-36.

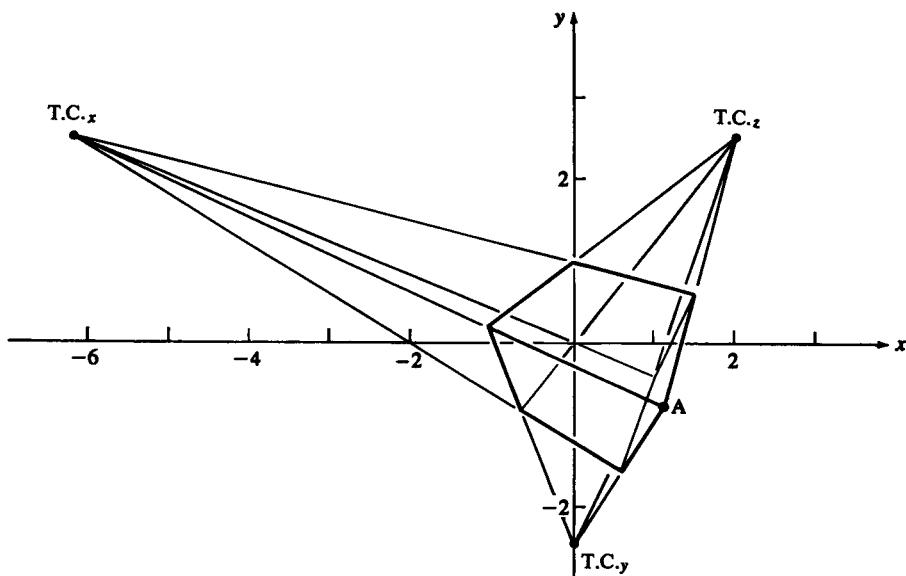


Рис. 3-36 Трехточечная перспективная проекция с поворотом вокруг двух осей.

Из этих результатов становится ясно, что одно-, дву- или трехточечное перспективное преобразование можно сконструировать с помощью поворотов и переносов вокруг и вдоль главных осей с последующим одноточечным перспективным преобразованием с центром проекции, расположенным на одной из главных осей. Эти результаты также справедливы для поворота вокруг произвольной оси в пространстве. Следовательно, при использовании в графической системе парадигмы с фиксированным центром проекции и манипулируемым объектом, необходимо обеспечить только построение одноточечной перспективной проекции на плоскость $z = 0$ с центром проекции на оси z .

3-17 ТОЧКИ СХОДА

Как показано на рис. 3-37 а при построении перспективного вида объекта используется линия горизонта, расположенная обычно на уровне глаз. Главные точки схода — это те точки на линии горизонта, в которые сходятся прямые, параллельные в исходном пространстве главным осям. В общем случае различные множества параллельных прямых имеют различные главные точки схода, как это показано на рис. 3-37 б. Для плоскостей объекта, расположенных наклонно относительно исходных главных осей, точки схода лежат выше или ниже линии горизонта. Как показано на рис. 3-37 с, такие точки часто называются следом точек.

Представляют интерес два метода определения точек схода. В первом просто вычисляется точка пересечения пары преобразованных спроектированных параллельных прямых. Второй метод сложен, но зато дает более точные результаты. В

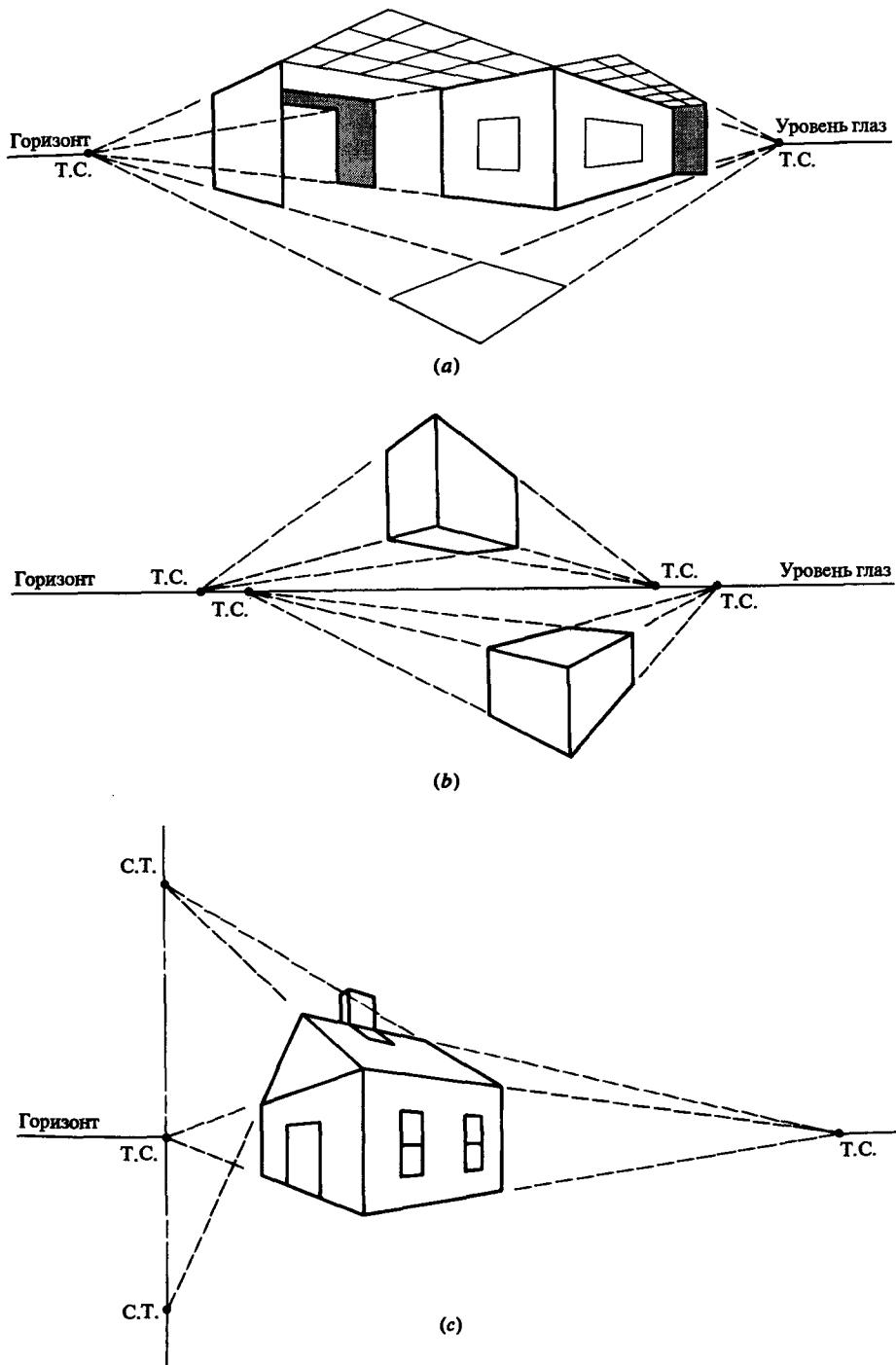


Рис. 3-37 Следы точек и точки схода.

в этом методе в нужную позицию и ориентацию преобразуется объект, стороны которого параллельны исходным главным осям. Потом применяется одноточечное перспективное проецирование. Затем результирующая матрица преобразования (см. уравнение 3-63) применяется к точкам, расположенным в бесконечности на главных осях. Полученные в результате этого обычные координаты есть главные точки схода для этого объекта. Для нахождения следов точек, возникающих для наклонных плоскостей, сначала находят расположенные в бесконечности точки на наклонной плоскости и затем подвергают их преобразованию.

Эти методы проиллюстрированы на нескольких примерах. В первом из них для определения точек схода используется пересечение преобразованных прямых.

Пример 3-25 Главные точки схода, определяемые по пересечению прямых

Из примера 3-23 преобразованные координатные векторы для пары отрезков, один из которых проходит через точку A (см. рис. 3-35 a), и первоначально параллельных соответственно осям x и z , равны

$$\begin{array}{ll} \textcircled{4} & \begin{bmatrix} 1.083 & -1.25 & 0 & 1 \end{bmatrix} \\ \textcircled{3} & \begin{bmatrix} 1.192 & -0.872 & 0 & 1 \end{bmatrix} \\ \textcircled{8} & \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix} \\ \textcircled{7} & \begin{bmatrix} 0.371 & 0.743 & 0 & 1 \end{bmatrix} \end{array} \quad \text{и} \quad \begin{array}{ll} \textcircled{3} & \begin{bmatrix} 1.192 & -0.872 & 0 & 1 \end{bmatrix} \\ \textcircled{7} & \begin{bmatrix} 0.371 & -0.743 & 0 & 1 \end{bmatrix} \\ \textcircled{4} & \begin{bmatrix} 1.083 & -1.25 & 0 & 1 \end{bmatrix} \\ \textcircled{8} & \begin{bmatrix} 0 & -1 & 0 & 1 \end{bmatrix} \end{array}$$

Здесь цифры в кружках относятся к строкам исходной и преобразованной матриц из примера 3-23. Уравнения пары прямых, параллельных в исходном пространстве оси x таковы:

$$\begin{aligned} y &= 3.468x - 5.006, \\ y &= 0.693x - 1. \end{aligned}$$

Решение этой системы дает точку схода $[VP_x] = [1.444 \ 0]$.

Уравнения пары прямых, параллельных в исходном пространстве оси z , имеют вид:

$$\begin{aligned} y &= -0.157x - 0.685 \\ y &= -0.231x - 1. \end{aligned}$$

Решение дает точку схода $[VP_z] = [-4.333 \ 0]$.

Эти точки показаны на рис. 3-35 b.

Во втором примере для нахождения точек схода используется преобразование расположенных в бесконечности точек на главных осях.

Пример 3-26 Главные точки схода, найденные с помощью преобразования

Общая матрица преобразования из примера 3-24

$$[T] = \begin{bmatrix} 0.866 & -0.354 & 0 & -0.141 \\ 0 & 0.707 & 0 & -0.283 \\ -0.5 & -0.612 & 0 & -0.245 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Преобразование расположенных в бесконечности на осях x , y и z точек дает

$$\begin{aligned} [VP][T] &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.866 & -0.354 & 0 & -0.141 \\ 0 & 0.707 & 0 & -0.283 \\ -0.5 & -0.612 & 0 & -0.245 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} -6.142 & 2.5 & 0 & 1 \\ 0 & -2.5 & 0 & 1 \\ 2.04 & 2.5 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Эти точки схода показаны на рис. 3-36.

В третьем примере для нахождения следов точек используется преобразование точек, расположенных в бесконечности на наклонных плоскостях.

Пример 3-27 Следы точек, полученные с помощью преобразования

Рассмотрим простую треугольную призму на рис. 3-38 а. Координатные векторы призмы

$$[X] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0.5 & 0.5 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0.5 & 0.5 & 0 & 1 \end{bmatrix}.$$

Применяя общее преобразование из примера 3-24, получим преобразованные координаты

$$\begin{aligned} [X^*] &= [X][T] = \\ &= \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0.5 & 0.5 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0.5 & 0.5 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & -0.354 & 0 & -0.141 \\ 0 & 0.707 & 0 & -0.283 \\ -0.5 & -0.612 & 0 & -0.245 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} -0.5 & -0.612 & 0 & 0.755 \\ 0.366 & -0.966 & 0 & 0.614 \\ -0.067 & -0.436 & 0 & 0.543 \\ 0 & 0 & 0 & 1 \\ 0.866 & -0.354 & 0 & 0.859 \\ 0.433 & 0.177 & 0 & 0.788 \end{bmatrix} = \\ &= \begin{bmatrix} -0.662 & -0.811 & 0 & 1 \\ 0.596 & -1.574 & 0 & 1 \\ -0.123 & -0.802 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1.009 & -0.412 & 0 & 1 \\ 0.55 & 0.224 & 0 & 1 \end{bmatrix}. \end{aligned}$$

На рис. 3-38 б изображена преобразованная призма.

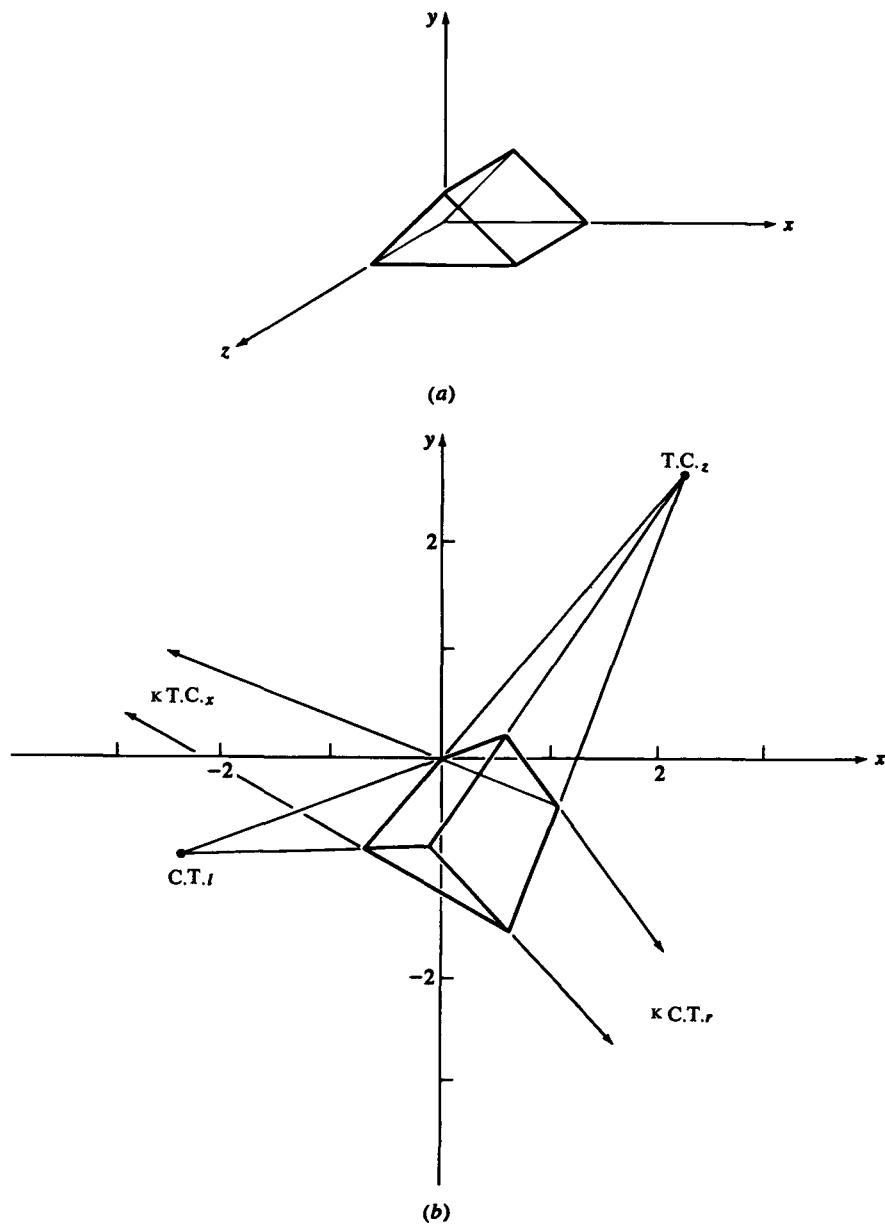


Рис. 3-38 Следы точек.

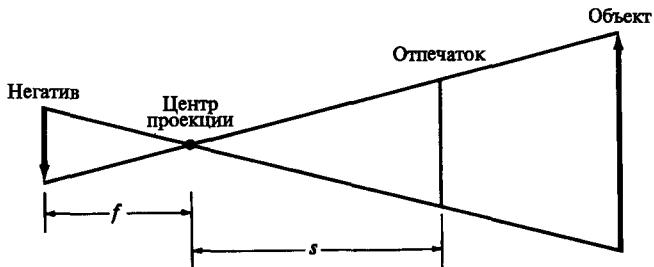


Рис. 3-39 Фотография как перспективная проекция.

Направляющие косинусы наклонных ребер левой верхней плоскости призмы до преобразования равны [0.5 0.5 0]. Таким образом, точка, лежащая в бесконечности в этом направлении, имеет однородные координаты [1 1 0 0].

Аналогично [-0.5 0.5 0] — направляющие косинусы наклонных ребер правой верхней плоскости призмы до преобразования. Таким образом, точка, лежащая в бесконечности в этом направлении, имеет координаты [-1 1 0 0].

Применяя преобразование к только что полученным точкам и к точкам, лежащим в бесконечности на главных осях, получим

$$\begin{aligned}
 [VP][T] &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.866 & -0.354 & 0 & -0.141 \\ 0 & 0.707 & 0 & -0.283 \\ -0.5 & -0.612 & 0 & -0.245 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 0.866 & -0.354 & 0 & -0.141 \\ 0 & 0.707 & 0 & -0.283 \\ -0.5 & -0.612 & 0 & -0.245 \\ 0.866 & 0.354 & 0 & -0.424 \\ -0.866 & 1.061 & 0 & -0.141 \end{bmatrix} = \\
 &= \begin{bmatrix} -6.142 & 2.5 & 0 & 1 \\ 0 & -2.5 & 0 & 1 \\ 2.041 & 2.5 & 0 & 1 \\ -2.041 & -0.833 & 0 & 1 \\ 6.142 & -7.5 & 0 & 1 \end{bmatrix} \begin{bmatrix} VP_x \\ VP_y \\ VP_z \\ TP_r \end{bmatrix}.
 \end{aligned}$$

Точки схода и следы точек изображены на рисунке 3-38 б. Как мы и ожидали, точки VP_x , VP_y и VP_z совпадают с полученными в примере 3-26.

3-18 ФОТОГРАФИЯ И ПЕРСПЕКТИВНОЕ ПРЕОБРАЗОВАНИЕ

Фотография — это перспективная проекция. Схема работы камеры в общем виде изображена на рис. 3-39. Центр проекции — это точка фокуса линзы камеры. Удобно рассматривать создание негатива и отпечатка с него как два разных процесса.

На рис. 3-40 а показана схема создания исходного негатива. Для удобства поместим негатив в плоскость $z = 0$ так, чтобы центр проекции и сцена находились

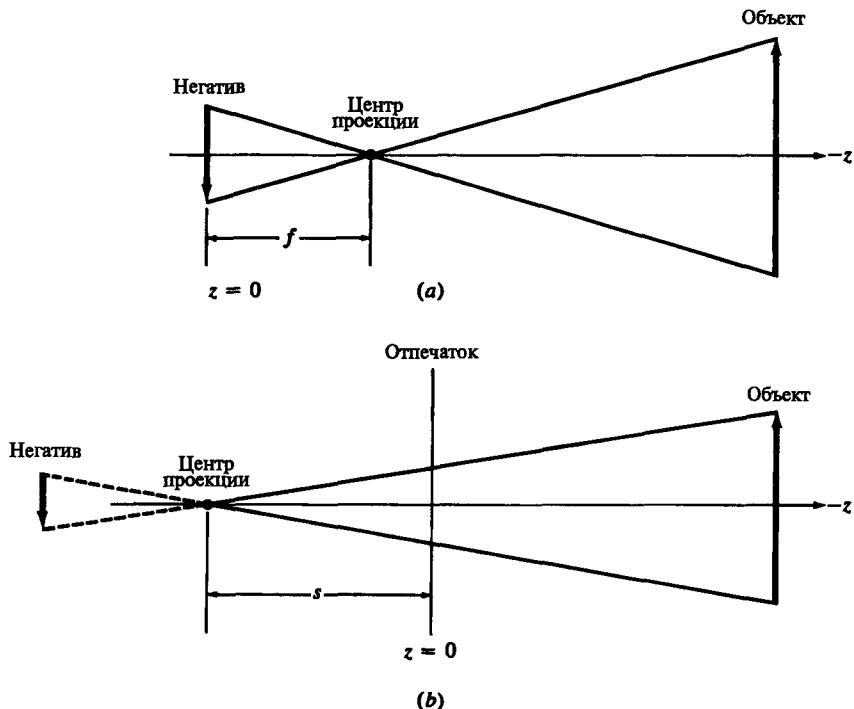


Рис. 3-40 Схема фотографической перспективы. (a) Создание исходного негатива; (b) создание оттиска.

в отрицательном полупространстве $z < 0$. Перспективное проецирование на плоскость $z = 0$ (негатив) порождает следующее преобразование:

$$[T_n] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/f \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3-64)$$

где f — это фокусное расстояние линзы. Заметим, что на негативе формируется перевернутое изображение объекта. Конкретнее,

$$[x \ y \ z \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/f \\ 0 & 0 & 0 & 1 \end{bmatrix} = [x \ y \ 0 \ 1 + z/f]$$

и

$$x^* = \frac{xf}{f+z}, \quad y^* = \frac{yf}{f+z}.$$

Здесь для $f + z < 0$ значения x^* и y^* имеют противоположный знак по сравнению с x и y , следовательно, на негативе образуется перевернутое изображение.

На рисунке 3-40 б изображена схема создания отпечатка с фотографического негатива. Здесь s обозначает расстояние от фокуса линзы увеличителя до бумаги. Предполагается, что бумага находится в $z = 0$. Тогда преобразование перспективного проецирования имеет вид:

$$[T_p] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/s \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3-65)$$

Заметим, что на отпечатке формируется прямое изображение. А именно,

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/s \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 - \frac{z}{s} \end{bmatrix}$$

и

$$x^* = \frac{xs}{s - z}, \quad y^* = \frac{ys}{s - z}.$$

Для объекта значение $z < 0$, тогда $s - z > 0$ и координаты x^* , y^* имеют тот же знак, что и x , y , т. е. формируется прямое изображение.

3-19 СТЕРЕОГРАФИЧЕСКАЯ ПРОЕКЦИЯ

Во многих приложениях важно увеличение восприятия трехмерной глубины сцены. Для восприятия глубины у системы глаз–мозг существует два основных типа подсказок: монокулярные и бинокулярные, в зависимости от того, сколько глаз используется — один или два. Основными монокулярными подсказками являются:

- перспективы — схождение в одну точку параллельных прямых;
- параллакс движения — при горизонтальном движении головы кажется, что ближние объекты сдвигаются на плоскости проекции больше, чем дальние;
- относительный размер знакомых наблюдателю объектов;
- перекрытие — более близкий объект загораживает остальные и располагается перед более отдаленным;
- свет и тени;
- ослабление атмосферой и неспособность глаза из-за этого различить мелкие детали удаленных объектов;
- фокусирующая аккомодация — для объектов, расположенных на разных расстояниях, требуются разные усилия фокусирующих мускулов глаза;

Основными бинокулярными подсказками являются:

- углы сближения оптических осей глаз;

- несогласованность на сетчатке — разница в местоположении спроектированных на стечатку глаза объектов интерпретируется как различие в расстоянии до глаз.

Монокулярное зрение дает лишь слабое восприятие трехмерной глубины. Однако бинокулярное зрение дает очень сильное восприятие глубины, потому что система глаз–мозг смешивает в одно два различных изображения, созданных для каждого глаза. Стереография пытается создать изображение с характеристиками, аналогичными настоящему бинокулярному зрению. Для генерации стереоизображений существуют несколько методов (см. [3-4] и [3-5]). Все они основываются на том, что для левого и правого глаза готовятся разные изображения.

В двух методах, называемых цветовым анаглифом и поляризованным анаглифом методами, для получения левым и правым глазами корректных и различных изображений используются фильтры. Если рассказать в нескольких словах, то в первом случае двумя разными цветами создаются два изображения — одно для левого и другое для правого глаза. При просмотре через соответствующие фильтры левый глаз видит левое изображение, а правый — только правое. Система глаз–мозг объединяет оба двумерных изображения в одно трехмерное с правильными цветами. Во втором случае вместо цветовых фильтров используются поляризованные.

В третьем методе используется попеременное проецирование видов для левого и правого глаза. Соответствующее устройство просмотра синхронизировано так, чтобы свет не попадал в другой глаз.

В четвертом методе, автостереоскопии, никакого специального оборудования для просмотра не требуется. Метод основан на использовании плоских или линзообразных экранов. Изображения называются параллаксными стереограммами, параллаксными панорамаграммами и панорамными параллаксными стереограммами. Более полную информацию по этому вопросу можно найти в [3-5].

Во всех этих методах требуется проецирование объекта на плоскость из двух различных центров проекции — одно для правого глаза и одно для левого. На рис. 3-41 показано проецирование на плоскость $z = 0$ точки P из центров проекции в $E_L(-e, 0, d_e)$ и $E_R(e, 0, d_e)$, соответствующих левому и правому глазу.

Как показано на рис. 3-41 *b*, для удобства центр проекции для левого глаза перенесен таким образом, чтобы он лежал на оси z . Используя подобные треугольники, получим

$$\frac{x_L^{*'}}{d_e} = \frac{x'}{d_e - z}$$

и

$$x_L^{*'} = \frac{x'}{1 - z/d_e} = \frac{x'}{1 + rz},$$

где

$$r = -1/d_e.$$

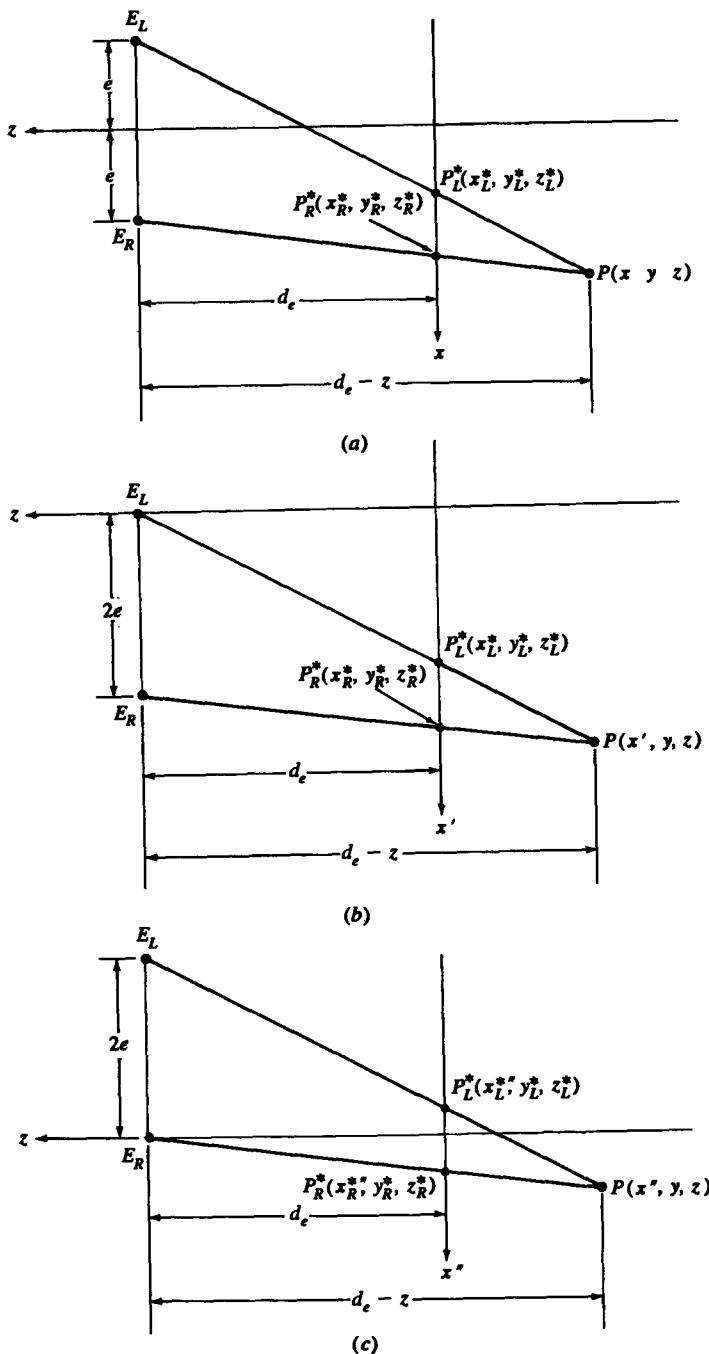


Рис. 3-41 Стереографическая проекция на плоскость $z = 0$.

Аналогичным образом перенеся центр проекции для правого глаза на ось z (как показано на рис. 3-41 *c*) и снова используя подобные треугольники, получим

$$\frac{x_R^{*''}}{d_e} = \frac{x''}{d_e - z}$$

и

$$x_R^{*''} = \frac{x''}{1 - z/d_e} = \frac{x''}{1 + rz}.$$

Так как для каждого глаза значение $y = 0$, то оба спроектированных значения y имеют вид:

$$y^* = \frac{y}{1 - z/d_e} = \frac{y}{1 + rz}.$$

Эквивалентные матрицы преобразования 4×4 построения видов для правого и левого глаза следующие:

$$\begin{aligned} [S_L] &= [Tr_x][P_{rz}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ e & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/d_e \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/d_e \\ e & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (3-66)$$

и

$$[S_R] = [Tr_x][P_{rz}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/d_e \\ e & 0 & 0 & 1 \end{bmatrix}. \quad (3-67)$$

Следовательно, стереографическая проекция получается при преобразовании сцены с помощью уравнений (3-66), (3-67) и вывода обоих изображений.

Существует ряд методов для вывода изображений стереографических проекций. Многие люди смогут создать стереоизображение без помощи каких-либо технических средств. Один из методов, требующий некоторой тренировки, состоит в следующем: сначала сфокусировать глаза в бесконечности, затем, не изменяя фокуса, постепенно передвигая стереопары и держа их примерно на расстоянии локтя, добиться стереоэффекта.

Бинокулярное смешение стереопар улучшается при использовании небольшой непрозрачной маски, например полоски черного картона шириной приблизительно в дюйм. Как показано на рис. 3-42, маску помещают между глазами и стереопарой и передвигают вперед и назад до получения нужного положения. В этом

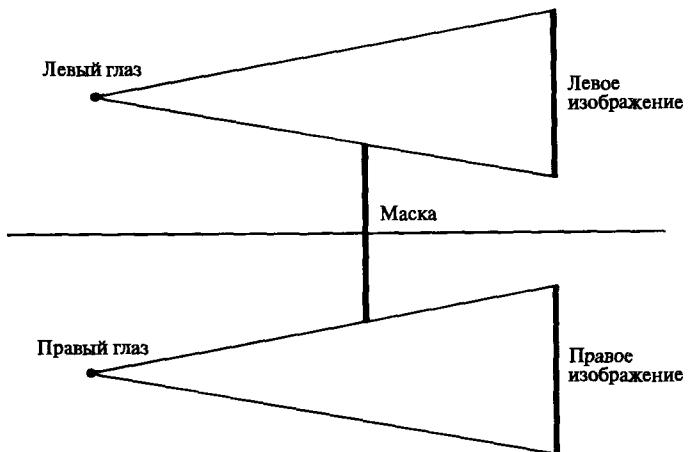


Рис. 3-42 Простой метод просмотра стереопар.

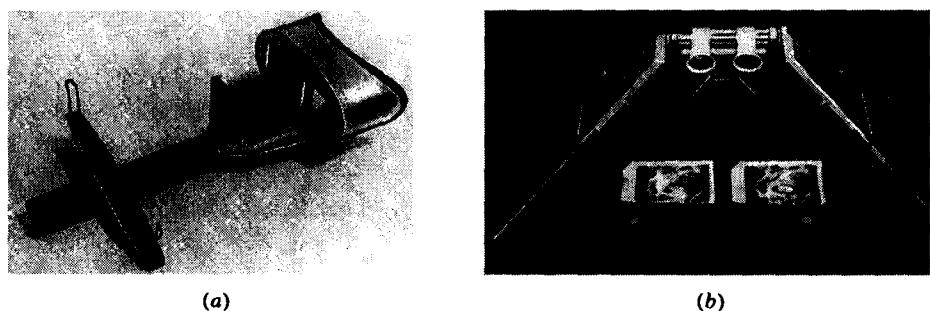


Рис. 3-43 Стереоскопы. (a) Стереоптикон Брустера; (b) типичный лабораторный инструмент.

положении левый глаз видит только левое изображение стереопары, а правый глаз — правое.

Два более специализированных устройства для просмотра стереопар изображены на рис. 3-43 *a* и 3-43 *b*. На рис. 3-43 *a* показан популярный в начале нашего века стереоптикон Брустера, а на рис. 3-43 *b* — типичный современный лабораторный стереоскоп. Оба устройства относятся к простым стереоскопам с фокусирующими линзами.

Хотя уравнения (3-66) и (3-67) задают основное преобразование для генерации стереопар, их надо модифицировать для приспособления к различным схемам стереопросмотровых устройств. Схема простого фокального стереоскопа приведена на рис. 3-44. Стереопары в нем располагаются на фокусном расстоянии f от линз. Стереоизображение реконструируется на расстоянии I от линз. Как было ранее упомянуто, стереоугол γ , очень сильно влияющий на бинокулярное смешение, связан со сближением оптических осей глаз. У обычного человека расстояние

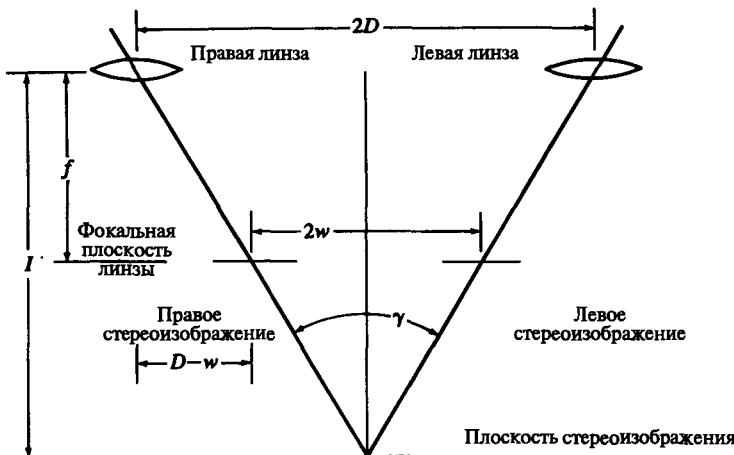


Рис. 3-44 Схема фокусного плоского стереоскопа.

между зрачками глаз приблизительно равно 60 мм. Эксперименты показали ([3-1]), что наилучший стереоэффект наблюдается на расстоянии приблизительно 300 мм. Следовательно, наилучший стереоэффект достигается для $\operatorname{tg}(\gamma/2) = 1/5$. Как показано на рис. 3-44, для достижения бинокулярного смещения стереопары должны быть разнесены на расстояние $2w$ при условии, что они размещены на фокусном расстоянии f .

Обозначив через $2D$ на рис. 3-44 расстояние между линзами стереоскопа, из подобия треугольников получим

$$\frac{D - w}{f} = \frac{D}{I} = \frac{1}{5}.$$

Таким образом, разнесение на расстояние

$$w = D - \frac{f}{5} \quad (3-68)$$

дает наилучший стереоэффект. Для стереопары, просматриваемой через простой линзовый стереоскоп, уравнения (3-66) и (3-67) записываются в виде

$$[S_L] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/f \\ w & 0 & 0 & 1 \end{bmatrix} \quad (3-69a)$$

и

$$[S_R] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1/f \\ -w & 0 & 0 & 1 \end{bmatrix}. \quad (3-69b)$$

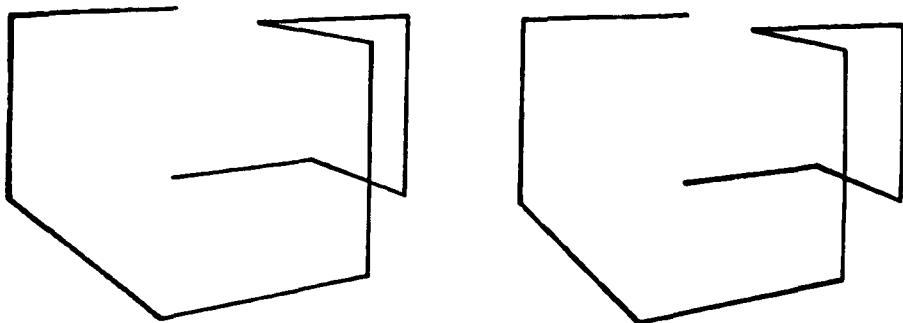


Рис. 3-45 Стереопара.

Типичная стереопара для простого линзового стереоскопа показана на рис. 3-45.

Расстояние между стереопарой, полученное из уравнения (3-68) для типичных значений D и f , довольно мало. Следовательно, можно просматривать только маленькие стереоизображения. Чтобы увеличить размеры изображений для увеличения расстояния между ними используются зеркала или призмы. Типичный зеркальный стереоскоп изображен на рис. 3-43 б. Соответствующая схема приведена на рис. 3-46. Снова используя подобие треугольников, получим расстояние между стереопарой

$$\frac{w_0 - w}{f} = \frac{D}{I} = \frac{1}{5}$$

и

$$w = \frac{w_0 - f}{5}. \quad (3-70)$$

Создание типичной стереопары проиллюстрировано на примере.

Пример 3-28 Создание стереопары

Рассмотрим изображенный на рис. 3-45 простой трехмерный проволочный объект с координатами

$$[X] = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 2 & 1 \\ -1 & 0 & 2 & 1 \\ -1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \end{bmatrix}.$$

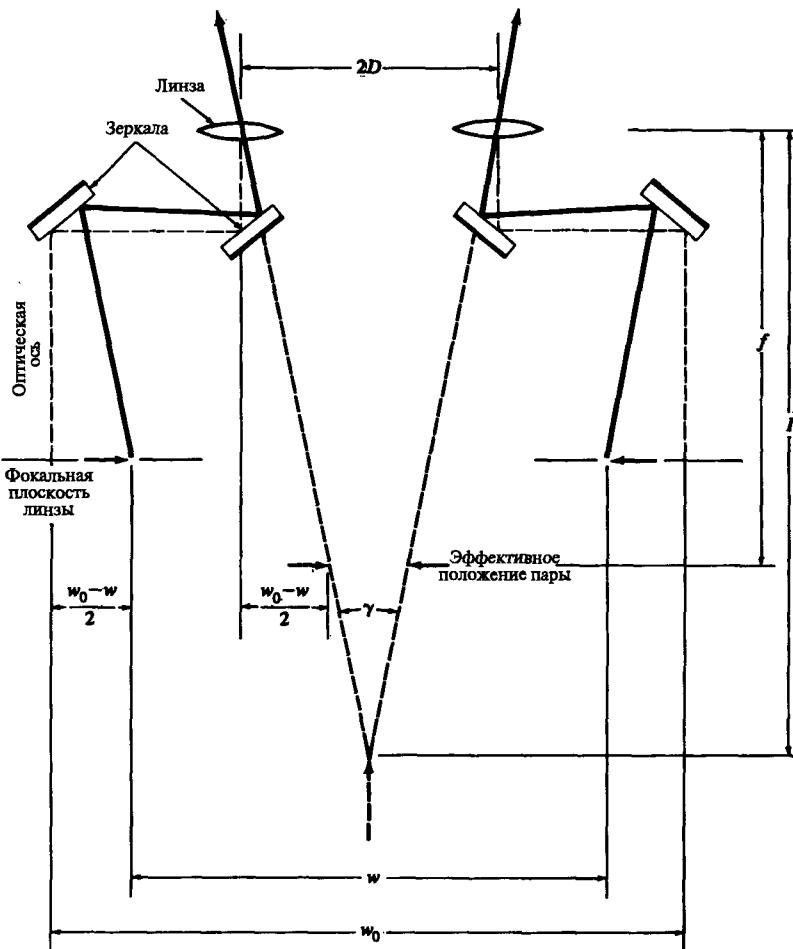


Рис. 3-46 Схема зеркального стереоскопа.

Для визуализации фигуру сначала повернем на 20° вокруг оси y , а затем перенесем на -1.5 единицы в направлении z . Результатирующее преобразование будет

$$\begin{aligned}
 [V] &= [R_y] [T_z] = \begin{bmatrix} 0.94 & 0 & -0.342 & 0 \\ 0 & 1 & 0 & 0 \\ 0.342 & 0 & 0.94 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1.5 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 0.94 & 0 & -0.342 & 0 \\ 0 & 1 & 0 & 0 \\ 0.342 & 0 & 0.94 & 0 \\ 0 & 0 & -1.5 & 1 \end{bmatrix}.
 \end{aligned}$$

Стереопара создается для просматривания с помощью простого фокусирующего плоского стереоскопа. Расстояние между линзами $2D = 4$ дюймам

и фокусное расстояние тоже равно 4 дюймам. Из уравнения (3-68) следует, что расстояние между стереопарой

$$w = 2 - 4/5 = 1.2''.$$

Используя уравнение (3-69), получим преобразования для левого и правого стереоизображений

$$[S_L] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.25 \\ 1.2 & 0 & 0 & 1 \end{bmatrix} \quad \text{и} \quad [S_R] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -0.25 \\ -1.2 & 0 & 0 & 1 \end{bmatrix}.$$

Объединение этих преобразований с видовым дает

$$[C_L] = [V][S_L] \quad \text{и} \quad [C_R] = [V][S_R],$$

$$[C_L] = \begin{bmatrix} 0.94 & 0 & 0 & 0.086 \\ 0 & 1 & 0 & 0 \\ 0.342 & 0 & 0 & -0.235 \\ 1.2 & 0 & 0 & 1.38 \end{bmatrix}$$

и

$$[C_R] = \begin{bmatrix} 0.94 & 0 & 0 & 0.086 \\ 0 & 1 & 0 & 0 \\ 0.342 & 0 & 0 & -0.235 \\ -1.2 & 0 & 0 & 1.375 \end{bmatrix}.$$

Преобразованные координаты для левого и правого изображений вычисляются следующим образом:

$$[X_L^*] = [X][C_L] \quad \text{и} \quad [X_R^*] = [X][C_R],$$

$$[X_L^*] = \begin{bmatrix} 0.873 & 0 & 0 & 1 \\ 1.465 & 0 & 0 & 1 \\ 2.025 & 0 & 0 & 1 \\ 2.025 & 0.816 & 0 & 1 \\ 1.353 & 0.877 & 0 & 1 \\ 2.081 & 1.105 & 0 & 1 \\ 2.081 & 0 & 0 & 1 \\ 1.152 & 0 & 0 & 1 \\ 0.202 & 0 & 0 & 1 \\ 0.202 & 0 & 0 & 1 \\ 0.202 & 0.775 & 0 & 1 \end{bmatrix} \quad \text{и} \quad [X_R^*] = \begin{bmatrix} -0.873 & 0 & 0 & 1 \\ -0.178 & 0 & 0 & 1 \\ 0.067 & 0 & 0 & 1 \\ 0.067 & 0.816 & 0 & 1 \\ -0.753 & 0.877 & 0 & 1 \\ -0.57 & 1.105 & 0 & 1 \\ -0.57 & 0 & 0 & 1 \\ -1.776 & 0 & 0 & 1 \\ -1.659 & 0 & 0 & 1 \\ -1.659 & 0 & 0 & 1 \\ -1.659 & 0.775 & 0 & 1 \end{bmatrix}.$$

Результат изображен на рис. 3-45. Заметим, что при создании конечного изображения эти данные должны быть промасштабированы для каждого конкретного устройства вывода, с тем чтобы обеспечить корректное физическое значение для w .

3-20 СРАВНЕНИЕ ДВУХ МЕТОДОВ ПОСТРОЕНИЯ ПРОЕКЦИЙ – С ФИКСИРОВАННЫМ ОБЪЕКТОМ И С ФИКСИРОВАННЫМ ЦЕНТРОМ ПРОЕКЦИИ

Метод проецирования, в котором объект зафиксирован, а перемещается центр проекции, может быть легко преобразован в обсуждавшийся ранее метод с перемещающимся объектом и фиксированным центром. Представляет интерес два случая. В первом, более простом, предполагается, что плоскость проекции перпендикулярна направлению взгляда, направленного из центра проекции к сцене. Во втором случае перпендикулярность не предполагается. Здесь будет рассмотрен только первый случай.

Когда плоскость проекции перпендикулярна направлению взгляда, с помощью описанной ниже процедуры построение проекции можно свести к случаю с движущимся объектом и фиксированным центром проекции:

- найти пересечение линии взгляда и плоскости проекции;
- перенести точку пересечения в начало координат;
- повернуть вектор взгляда таким образом, чтобы он совместился с осью $+z$ и был направлен к началу системы координат (см. разд. 3-9);
- применить к сцене полученное преобразование;
- выполнить одноточечное перспективное проецирование на плоскость $z = 0$ из преобразованного центра проекции, расположенного теперь на оси z .

В качестве иллюстрации этого метода приведем относительно простой пример.

Пример 3-29 Перспективное проецирование на перпендикулярную плоскость проекции для фиксированного объекта

Рассмотрим обсуждавшийся ранее в примере 3-10 куб с отсеченным углом. Спроектируем куб из центра в точке $[10\ 10\ 10]$ на плоскость, проходящую через точку $[-1\ -1\ -1]$ и перпендикулярную направлению взгляда, как это показано на рис. 3-47.

Уравнение плоскости проекции можно получить из ее нормали (другие методы можно найти в [3-1]). В данном случае нормаль направлена в сторону, противоположную вектору взгляда.

Направление вектора взгляда

$$[s] = [-1 \quad -1 \quad -1].$$

Тогда нормаль к плоскости проекции, проходящей через точку $[-1\ -1\ -1]$ перпендикулярно направлению взгляда, будет

$$[n] = [1 \quad 1 \quad 1].$$

Общее выражение уравнения плоскости имеет вид:

$$ax + by + cz + d = 0.$$

Нормаль к такой плоскости задается вектором

$$[n] = [a \quad b \quad c].$$

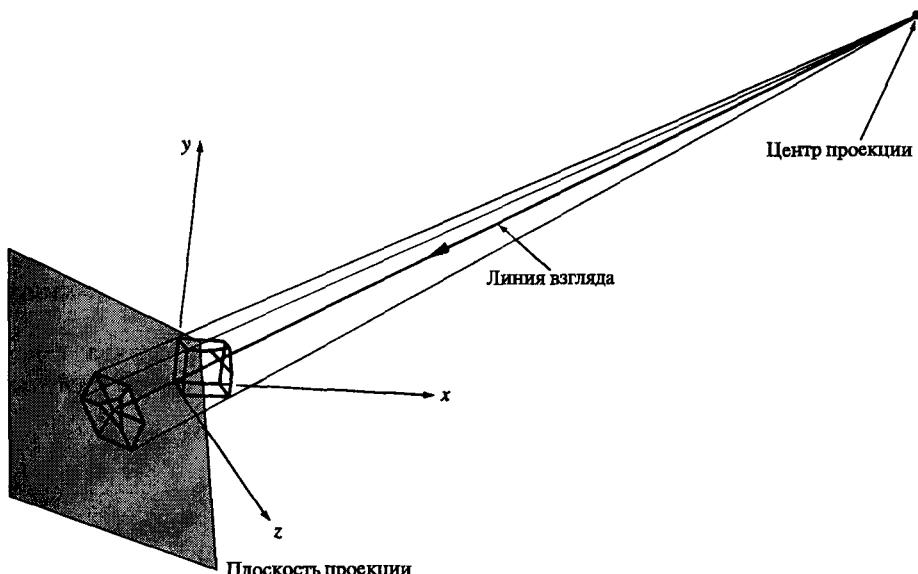


Рис. 3-47 Перспективное проецирование с перемещающимся центром проекции.

Значение d в уравнении плоскости вычисляется по любой точке, принадлежащей этой плоскости. Тогда уравнение плоскости проекции, проходящей через точку $[-1 \ -1 \ -1]$, примет вид:

$$x + y + z + d = 0$$

и

$$d = -x - y - z = 1 + 1 + 1 = 3.$$

Следовательно,

$$x + y + z + 3 = 0$$

есть уравнение плоскости проекции.

Пересечение направления взгляда и плоскости проекции определяется с помощью записи уравнения прямой в параметрическом виде, подстановки его в уравнение плоскости и решения полученного уравнения.

Параметрическое уравнение вектора взгляда таково:

$$\begin{aligned} [S(t)] &= [x(t) \quad y(t) \quad z(t)] = \\ &= [10 \quad 10 \quad 10] + [-11 \quad -11 \quad -11]t, \quad 0 \leq t \leq 1. \end{aligned}$$

Его подстановка в уравнение плоскости приводит к

$$x(t) + y(t) + z(t) + 3 = (10 - 11t) + (10 - 11t) + (10 - 11t) + 3 = 0.$$

Решение уравнения дает значение параметра для точки пересечения, т. е.

$$-33t + 33 = 0 \rightarrow t = 1.0.$$

Саму точку пересечения получаем подстановкой t в $[S(t)]$. А именно,

$$\begin{aligned}[I] &= [S(1)] = [10 \ 10 \ 10] + [-11 \ -11 \ -11](1.0) = \\ &= [-1 \ -1 \ -1].\end{aligned}$$

Как и ожидалось из простых геометрических рассуждений, точка пересечения находится в $x = y = z = -1$.

Требуемая матрица переноса (для переноса точки пересечения в начало координат) следующая:

$$[Tr] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

После переноса центр проекции находится в точке $[11 \ 11 \ 11]$, а направление взгляда проходит через начало координат.

Используя результаты разд. 3-9, совместить вектор взгляда с осью z можно с помощью Поворота вокруг оси x на угол $\alpha = 45^\circ$ и последующего поворота вокруг оси b на угол $\beta = 35.26^\circ$. Матрицы поворотов записываются в виде:

$$[R_x] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{и} \quad [R_y] = \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} & 0 \\ 0 & 1 & 0 & 0 \\ -1/\sqrt{3} & 0 & 2/\sqrt{6} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

а суммарная матрица преобразования имеет вид

$$[M] = [Tr][R_x][R_y] = \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} & 0 \\ 0 & 0 & 3/\sqrt{3} & 1 \end{bmatrix}.$$

Преобразование центра проекции дает

$$\begin{aligned}[C_p][M] &= [10 \ 10 \ 10 \ 1] \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} & 0 \\ 0 & 0 & 3/\sqrt{3} & 1 \end{bmatrix} = \\ &= [0 \ 0 \ 33/\sqrt{3} \ 1].\end{aligned}$$

Преобразование одноточечного центрального проецирования с центром в точке $z = 33/\sqrt{3}$ на плоскость $z = 0$ задается матрицей

$$[P_{rz}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\sqrt{3}/33 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Объединив с $[M]$, получим, что

$$\begin{aligned}[T] &= [M][P_{rz}] = \begin{bmatrix} 2/\sqrt{6} & 0 & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} & 0 \\ -1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} & 0 \\ 0 & 0 & 3/\sqrt{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\sqrt{3}/33 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 2/\sqrt{6} & 0 & 0 & -1/33 \\ -1/\sqrt{6} & 1/\sqrt{2} & 0 & -1/33 \\ -1/\sqrt{6} & -1/\sqrt{2} & 0 & -1/33 \\ 0 & 0 & 0 & 30/33 \end{bmatrix}. \end{aligned}$$

Преобразованные обычные координаты спроектированного объекта будут

$$\begin{aligned}[X^*] &= [X][T] = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0.5 & 1 & 1 \\ 0.5 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} 0.816 & 0 & 0 & -0.030 \\ -0.408 & 0.707 & 0 & -0.030 \\ -0.408 & -0.707 & 0 & -0.030 \\ 0 & 0 & 0 & 0.909 \end{bmatrix} = \\ &= \begin{bmatrix} -0.408 & -0.707 & 0 & 0.879 \\ 0.408 & -0.707 & 0 & 0.848 \\ 0.204 & -0.354 & 0 & 0.833 \\ -0.408 & 0 & 0 & 0.833 \\ -0.816 & 0 & 0 & 0.848 \\ 0 & 0 & 0 & 0.909 \\ 0.816 & 0 & 0 & 0.879 \\ 0.408 & 0.707 & 0 & 0.848 \\ -0.408 & 0.707 & 0 & 0.879 \\ 0.204 & 0.354 & 0 & 0.833 \end{bmatrix} = \\ &= \begin{bmatrix} -0.465 & -0.805 & 0 & 1 \\ 0.481 & -0.833 & 0 & 1 \\ 0.245 & -0.424 & 0 & 1 \\ -0.490 & 0 & 0 & 1 \\ -0.962 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0.929 & 0 & 0 & 1 \\ 0.481 & 0.833 & 0 & 1 \\ -0.465 & 0.805 & 0 & 1 \\ 0.245 & 0.424 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Результат изображен на рис. 3-48.

3-21 ВОССТАНОВЛЕНИЕ ТРЕХМЕРНЫХ ОБЪЕКТОВ ПО ПРОЕКЦИЯМ

Задача восстановления трехмерной формы объекта или координат в пространстве широко распространена. Например, она постоянно возникает при использовании

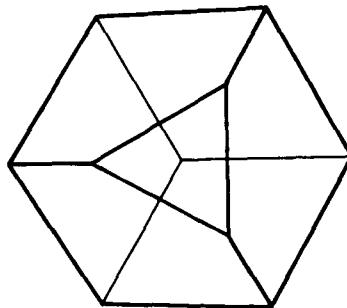


Рис. 3-48 Результат проецирования в примере 3-29.

чертежей, являющихся ортографическими проекциями. Хорошо известен метод восстановления трехмерного объекта или координаты по двум или более представленным на чертеже видам (ортографическим проекциям). Метод же восстановления трехмерного координатного вектора по двум перспективным проекциям, например по двум фотографиям, не столь хорошо известен. Конечно, если метод пригоден для перспективных проекций, то он подходит также и для более простых ортографических проекций и фактически для всех рассмотренных в предыдущих разделах проекций. Как будет показано ниже, при наличии определенной дополнительной информации не надо иметь точную информацию о самом преобразовании.

Прежде чем рассматривать задачу в общем виде, обратимся к специальному случаю восстановления трехмерных координат точки по двум или более ортографическим проекциям. На рис. 3-49 представлены ортографические виды (проекции) спереди, справа и сверху. При определении трехмерных координат точки *A* вид спереди дает значения координат *x* и *y*, вид справа — *y* и *z* и вид сверху — *x* и *z*, т. е.

$$\begin{array}{ll} \text{спереди (front):} & x_f \quad y_f \\ \text{справа (right side):} & y_r \quad z_r \\ \text{сверху (top):} & x_t \quad z_t \end{array}$$

Заметим, что для каждой координаты мы получаем два значения. В математике в подобном случае говорят, что на задачу наложены избыточные условия. Ведь надо найти только три независимые величины, тогда как известны шесть определяющих их условий (уравнений). Для любой измерительной системы в общем случае $x_f \neq x_t$, $y_f \neq y_r$, $z_r \neq z_t$ ¹. Так как неизвестно, какое из значений верно, то наиболее разумным решением будет усреднить их.

Вернувшись теперь к восстановлению трехмерных координат по перспективным проекциям, напомним, что обобщенное перспективное преобразование представляется в виде матриц размера 4×4 . Таким образом,

$$[x \quad y \quad z \quad 1] [T'] = [x' \quad y' \quad z' \quad h],$$

¹ По этой причине в чертежах размеры указываются явным образом.

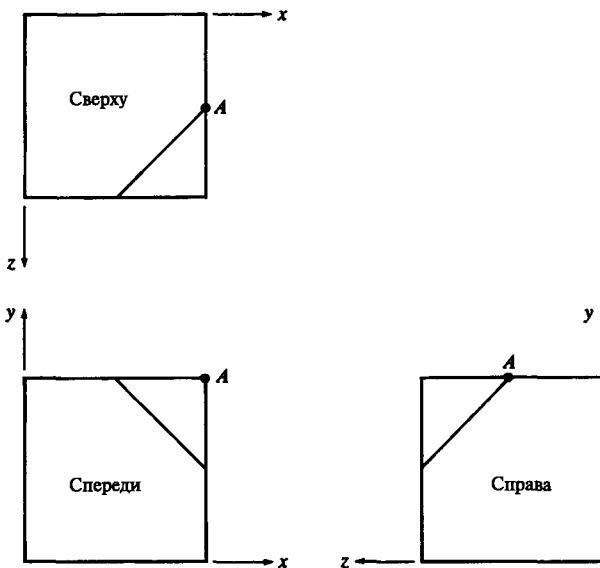


Рис. 3-49 Восстановление трехмерной формы по ортографическим проекциям.

где

$$[T'] = \begin{bmatrix} T'_{11} & T'_{12} & T'_{13} & T'_{14} \\ T'_{21} & T'_{22} & T'_{23} & T'_{24} \\ T'_{31} & T'_{32} & T'_{33} & T'_{34} \\ T'_{41} & T'_{42} & T'_{43} & T'_{44} \end{bmatrix}.$$

Результаты могут быть спроектированы на двумерную плоскость, скажем на плоскость $z = 0$, с помощью

$$[T''] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Объединение двух матриц дает

$$[T] = [T''] [T'] = \begin{bmatrix} T_{11} & T_{12} & 0 & T_{14} \\ T_{21} & T_{22} & 0 & T_{24} \\ T_{31} & T_{32} & 0 & T_{34} \\ T_{41} & T_{42} & 0 & T_{44} \end{bmatrix}.$$

Полезно записать это преобразование в виде

$$[x \ y \ z \ 1] \begin{bmatrix} T_{11} & T_{12} & 0 & T_{14} \\ T_{21} & T_{22} & 0 & T_{24} \\ T_{31} & T_{32} & 0 & T_{34} \\ T_{41} & T_{42} & 0 & T_{44} \end{bmatrix} = [x' \ y' \ 0 \ h] = h[x^* \ y^* \ 0 \ 1]. \quad (3-71)$$

Заметим, что x^* и y^* — это координаты перспективной проекции на плоскость $z = 0$. Могли бы также использоваться проекции на плоскости $x = 0$ или $y = 0$.

Расписав в явном виде матричное уравнение (3-71), получим

$$T_{11}x + T_{21}y + T_{31}z + T_{41} = hx^*, \quad (3-72\text{a})$$

$$T_{12}x + T_{22}y + T_{32}z + T_{42} = hy^*, \quad (3-72\text{b})$$

$$T_{14}x + T_{24}y + T_{34}z + T_{44} = h. \quad (3-72\text{c})$$

Подставим значение h из (3-72 c) в уравнения (3-72 a) и (3-72 b):

$$(T_{11} - T_{14}x^*)x + (T_{21} - T_{24}x^*)y + (T_{31} - T_{34}x^*)z + (T_{41} - T_{44}x^*) = 0, \quad (3-73\text{a})$$

$$(T_{12} - T_{14}y^*)x + (T_{22} - T_{24}y^*)y + (T_{32} - T_{34}y^*)z + (T_{42} - T_{44}y^*) = 0. \quad (3-73\text{b})$$

По предложению Сазерленда [3–4] эту пару уравнений можно рассматривать тремя разными способами. В первом предполагаются известными T , x , y , z . Тогда у нас есть два уравнения с двумя неизвестными x^* и y^* . Таким образом, эти уравнения можно использовать для непосредственного вычисления координат перспективной проекции. Именно этот подход применялся во всех предыдущих рассуждениях в данной главе.

При другом подходе полагаются известными T , x^* , y^* . В этом случае получается два уравнения от трех неизвестных пространственных координат x , y , z . Эту систему уравнений невозможно решить. Однако, если известны две перспективные проекции, скажем две фотографии, тогда уравнения (3-73) можно записать для обеих проекций. В результате получим

$$(T_{11}^1 - T_{14}^1 x^{*1})x + (T_{21}^1 - T_{24}^1 x^{*1})y + (T_{31}^1 - T_{34}^1 x^{*1})z + (T_{41}^1 - T_{44}^1 x^{*1}) = 0,$$

$$(T_{12}^1 - T_{14}^1 y^{*1})x + (T_{22}^1 - T_{24}^1 y^{*1})y + (T_{32}^1 - T_{34}^1 y^{*1})z + (T_{42}^1 - T_{44}^1 y^{*1}) = 0,$$

$$(T_{11}^2 - T_{14}^2 x^{*2})x + (T_{21}^2 - T_{24}^2 x^{*2})y + (T_{31}^2 - T_{34}^2 x^{*2})z + (T_{41}^2 - T_{44}^2 x^{*2}) = 0,$$

$$(T_{12}^2 - T_{14}^2 y^{*2})x + (T_{22}^2 - T_{24}^2 y^{*2})y + (T_{32}^2 - T_{34}^2 y^{*2})z + (T_{42}^2 - T_{44}^2 y^{*2}) = 0,$$

где верхние индексы 1 и 2 обозначают первую и вторую перспективные проекции. Заметим, что преобразования $[T^1]$ и $[T^2]$ не обязательно должны быть одинаковыми. Эти уравнения можно переписать в матричной форме

$$[A][X] = [B], \quad (3-74)$$

где

$$A = \begin{bmatrix} T_{11}^1 - T_{14}^1 x^{*1} & T_{21}^1 - T_{24}^1 x^{*1} & T_{31}^1 - T_{34}^1 x^{*1} \\ T_{12}^1 - T_{14}^1 y^{*1} & T_{22}^1 - T_{24}^1 y^{*1} & T_{32}^1 - T_{34}^1 y^{*1} \\ T_{11}^2 - T_{14}^2 x^{*2} & T_{21}^2 - T_{24}^2 x^{*2} & T_{31}^2 - T_{34}^2 x^{*2} \\ T_{12}^2 - T_{14}^2 y^{*2} & T_{22}^2 - T_{24}^2 y^{*2} & T_{32}^2 - T_{34}^2 y^{*2} \end{bmatrix},$$

$$[X]^T = [x \quad y \quad z]$$

$$[B]^T = [T_{44}^1 x^{*1} - T_{41}^1 \quad T_{44}^1 y^{*1} - T_{42}^1 \quad T_{44}^2 x^{*2} - T_{41}^2 \quad T_{44}^2 y^{*2} - T_{42}^2].$$

Уравнения (3-73) представляют четыре уравнения от трех неизвестных пространственных координат x , y , z . Для нахождения решения $[X]$ матрица $[A]$ не может быть инвертирована, так как она не квадратная. Снова, как и в случае восстановления трехмерных координат по ортографическим проекциям, условия избыточны и, таким образом, задача может быть решена только в некотором усредненном или наиболее подходящем смысле.

Усредненное решение можно вычислить, вспомнив, что матрица, умноженная на свою транспонированную матрицу, является квадратной. Таким образом, умножив обе части уравнения (3-74) на $[A]^T$, получим

$$[A]^T [A] [X] = [A]^T [B].$$

Вычислив обратную матрицу $[[A]^T [A]]$, получим усредненное решение для $[X]$, т. е.

$$[X] = [[A]^T [A]]^{-1} [A]^T [B]. \quad (3-75)$$

Если решения для $[X]$ нет, то наложенные условия избыточны и не существуют единственного решения, обеспечивающего наименьшую ошибку. Этот метод иллюстрируется на примере.

Пример 3-30 Трехмерное восстановление

Предположим, что обмеренные координаты точки на одной перспективной проекции равны $[0.836 \ 1.836 \ 0 \ 1]$ и $[0.6548 \ 0 \ 0.2886 \ 1]$ на другой. Первое преобразование Перспективной проекции является результатом поворота на 60° вокруг оси y и последующего переноса на 2 единицы в отрицательном направлении оси y . Точка проекции находится в $z = -1$, а результат проецируется на плоскость $z = 0$. Это действительно является двуточечной перспективной проекцией. Второе преобразование является результатом поворотов на 30° вокруг каждой из осей x , y . Точка проекции находится в $y = -1$, а результат проецируется на плоскость $y = 0$, т. е. является трехточечной перспективной проекцией. Таким образом преобразования $[T^1]$ и $[T^2]$

$$[T^1] = \begin{bmatrix} 0.5 & 0 & 0 & -0.87 \\ 0 & 1 & 0 & 0 \\ 0.87 & 0 & 0 & 0.5 \\ 0 & -2 & 0 & 1 \end{bmatrix} \text{ и } [T^2] = \begin{bmatrix} 0.87 & 0 & -0.5 & 0 \\ 0.25 & 0 & 0.43 & 0.87 \\ 0.43 & 0 & 0.75 & -0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Отметим, что последние две строки $[A]$ и $[B]$ должны быть переписаны, чтобы учсть тот факт, что $[T^2]$ является проецированием на плоскость $y = 0$. Тогда матрица A

$$[A] = \begin{bmatrix} 1.22 & 0 & 0.45 \\ -1.59 & 1 & 0.92 \\ 0.87 & -0.32 & 0.76 \\ -0.5 & 0.18 & 0.89 \end{bmatrix}$$

и

$$[B]^T = [0.84 \ 0.16 \ 0.65 \ 0.29].$$

Решив уравнение, получим $[X] = [0.5 \ 0.5 \ 0.5]$, т. е. центр единичного куба.

В качестве третьего подхода к рассмотрению уравнений (3-73) отметим, что если известны координаты нескольких точек в объектном пространстве и на перспективной проекции, то можно найти элементы преобразования T_{ij} . Эти элементы впоследствии можно использовать для нахождения местоположения неизвестных точек, используя описанный выше второй метод. Чтобы показать это, перепишем (3-73) в виде

$$T_{11}x + T_{21}y + T_{31}z + T_{41} - T_{14}xx^* - T_{24}yx^* - T_{34}zx^* - T_{44}x^* = 0, \quad (3-76\text{ a})$$

$$T_{12}x + T_{22}y + T_{32}z + T_{42} - T_{14}xy^* - T_{24}yy^* - T_{34}zy^* - T_{44}y^* = 0. \quad (3-76\text{ b})$$

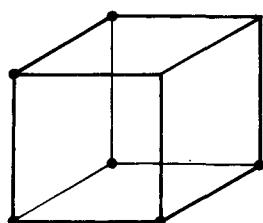
В предположении, что известны x^*, y^* , x, y, z , уравнения (3-76 a) и (3-76 b) представляют два уравнения с 12 неизвестными элементами T_{ij} . Применяя эти уравнения к 6 некомпланарным точкам в объектном пространстве и на перспективной проекции, получим систему из 12 уравнений с 12 неизвестными. Эти уравнения могут быть решены относительно T_{ij} . Таким образом, находится преобразование, породившее перспективную проекцию, например фотографию. Заметим, что в этом случае не требовалось никакой предварительной информации о преобразовании. Если, например, это были фотографии, то не требовалось знать ни о местоположении, ни об ориентации фотокамеры. В матричном виде система уравнений записывается в таком виде:

$$\left[\begin{array}{ccccccccccccc} x_1 & 0 & -x_1x_1^* & y_1 & 0 & -y_1x_1^* & z_1 & 0 & -z_1x_1^* & 1 & 0 & -x_1^* \\ 0 & x_1 & -x_1y_1^* & 0 & y_1 & -y_1y_1^* & 0 & z_1 & -z_1y_1^* & 0 & 1 & -y_1^* \\ x_2 & 0 & -x_2x_2^* & y_2 & 0 & -y_2x_2^* & z_2 & 0 & -z_2x_2^* & 1 & 0 & -x_2^* \\ 0 & x_2 & -x_2y_2^* & 0 & y_2 & -y_2y_2^* & 0 & z_2 & -z_2y_2^* & 0 & 1 & -y_2^* \\ x_3 & 0 & -x_3x_3^* & y_3 & 0 & -y_3x_3^* & z_3 & 0 & -z_3x_3^* & 1 & 0 & -x_3^* \\ 0 & x_3 & -x_3y_3^* & 0 & y_3 & -y_3y_3^* & 0 & z_3 & -z_3y_3^* & 0 & 1 & -y_3^* \\ x_4 & 0 & -x_4x_4^* & y_4 & 0 & -y_4x_4^* & z_4 & 0 & -z_4x_4^* & 1 & 0 & -x_4^* \\ 0 & x_4 & -x_4y_4^* & 0 & y_4 & -y_4y_4^* & 0 & z_4 & -z_4y_4^* & 0 & 1 & -y_4^* \\ x_5 & 0 & -x_5x_5^* & y_5 & 0 & -y_5x_5^* & z_5 & 0 & -z_5x_5^* & 1 & 0 & -x_5^* \\ 0 & x_5 & -x_5y_5^* & 0 & y_5 & -y_5y_5^* & 0 & z_5 & -z_5y_5^* & 0 & 1 & -y_5^* \\ x_6 & 0 & -x_6x_6^* & y_6 & 0 & -y_6x_6^* & z_6 & 0 & -z_6x_6^* & 1 & 0 & -x_6^* \\ 0 & x_6 & -x_6y_6^* & 0 & y_6 & -y_6y_6^* & 0 & z_6 & -z_6y_6^* & 0 & 1 & -y_6^* \end{array} \right] = \begin{bmatrix} T_{11} \\ T_{12} \\ T_{14} \\ T_{21} \\ T_{22} \\ T_{24} \\ T_{31} \\ T_{32} \\ T_{34} \\ T_{41} \\ T_{42} \\ T_{44} \end{bmatrix} = 0, \quad (3-77)$$

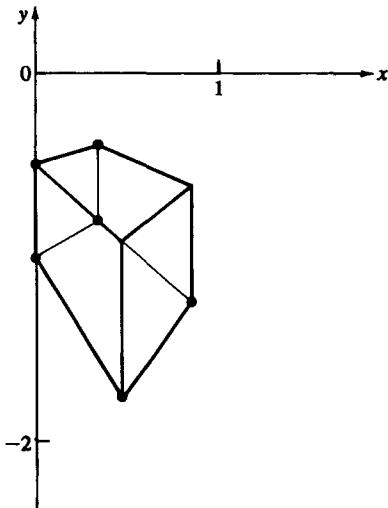
где нижние индексы соответствуют точкам с известным местоположением. Уравнения (3-77) записываются в более компактном виде:

$$[A'] [T] = 0.$$

Так как уравнения (3-77) являются однородными уравнениями, то они содержат произвольный масштабный коэффициент. Следовательно, T_{44} можно приравнять единице и нормализовать результирующее преобразование. Система сводится к 11 уравнениям или 5 1/2 точки. Если преобразование нормализовано, то



(a)



(b)

Рис. 3-50 Восстановление преобразования по перспективной проекции.

последний столбец в $[A']$ переносится в правую часть и решается неоднородное матричное уравнение. Ниже приводится пример.

Пример 3-31 Основы восстановления

В качестве конкретного примера рассмотрим единичный куб с шестью известными угловыми точками

$$[P] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Как показано на рис. 3-50, соответствующие им точки на проекции отмечены жирными кружками. Преобразованные координаты этих точек таковы:

$$\begin{bmatrix} 0 & -1 \\ 0.34 & -0.8 \\ 0.34 & -0.4 \\ 0 & -0.5 \\ 0.44 & -1.75 \\ 0.83 & -1.22 \end{bmatrix}.$$

Тогда (3-77) записывается в виде

$$\left[\begin{array}{ccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -0.34 & 1 & 0 & -0.34 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 & 0 & 1 & 0.8 \\ 0 & 0 & 0 & 1 & 0 & -0.34 & 1 & 0 & -0.34 & 1 & 0 & -0.34 \\ 0 & 0 & 0 & 0 & 1 & 0.4 & 0 & 1 & 0.4 & 0 & 1 & 0.4 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.5 & 0 & 0 & 0 & 0 & 1 & 0.5 \\ 1 & 0 & -0.44 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -0.44 \\ 0 & 1 & 1.75 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1.75 \\ 1 & 0 & -0.83 & 0 & 0 & 0 & 1 & 0 & -0.83 & 1 & 0 & -0.83 \\ 0 & 1 & 1.22 & 0 & 0 & 0 & 0 & 1 & 1.22 & 0 & 1 & 1.22 \end{array} \right] \left[\begin{array}{c} T_{11} \\ T_{12} \\ T_{14} \\ T_{21} \\ T_{22} \\ T_{24} \\ T_{31} \\ T_{32} \\ T_{34} \\ T_{41} \\ T_{42} \\ T_{44} \end{array} \right] = 0.$$

Решение относительно 12 неизвестных элементов T_{ij} такое:

$$\left[\begin{array}{c} T_{11} \\ T_{12} \\ T_{14} \\ T_{21} \\ T_{22} \\ T_{24} \\ T_{31} \\ T_{32} \\ T_{34} \\ T_{41} \\ T_{42} \\ T_{44} \end{array} \right] = \left[\begin{array}{c} 0.25 \\ 0 \\ -0.43 \\ 0 \\ 0.5 \\ 0 \\ 0.43 \\ 0 \\ 0.25 \\ 0 \\ -1 \\ 1 \end{array} \right].$$

Подстановка этих результатов в матрицу $[T]$ 4×4 дает

$$[T] = \left[\begin{array}{cccc} 0.25 & 0 & 0 & -0.43 \\ 0 & 0.5 & 0 & 0 \\ 0.43 & 0 & 0 & 0.25 \\ 0 & -1 & 0 & 1 \end{array} \right].$$

3-22 ЛИТЕРАТУРА

- 3-1 Rogers, D. F., *Procedural Elements for Computer Graphics*, McGraw-Hill Book Co., New York, 1985. (См. [1-1] на с. 74.)
- 3-2 Carlbom, I., and Paciorek, J., «Planar Geometric Projections and Viewing Transformations», *ACM Comp. Surv.*, Vol. 10, No. 4, pp. 465–502, 1978.
- 3-3 Sutherland, I. E., «Three Dimensional Data Input by Tablet», *Proc. IEEE*, Vol. 62, No. 2, pp. 453–461, 1974.
- 3-4 Slama, C. (ed.), *Manual of Photogrammetry*, American Society of Photogrammetry, 1980.
- 3-5 Kingslake, R. (ed.), *Applied Optics and Optical Engineering*, Vol. 2, Academic Press, New York, 1965.

Глава 4

ПЛОСКИЕ КРИВЫЕ

4-1 ВВЕДЕНИЕ

Существует множество способов построения кривых вручную с помощью карандаша, ручки, кисточки, ножа и разнообразных инструментов: линейки, лекала, циркуля, плаза, шаблона и т. д. Каждый инструмент служит определенной цели, причем нет ни одного абсолютного универсального. Точно так же в машинной графике кривые строятся с помощью разных методов и инструментов. В этой главе рассматриваются методы построения двумерных кривых, т. е. полностью лежащих в одной плоскости. Здесь мы ограничимся коническими сечениями.

4-2 ПРЕДСТАВЛЕНИЕ КРИВЫХ

В предыдущих двух главах изучались преобразования точек. Кривая может быть представлена совокупностью точек. Если точки расположены близко друг от друга, то, соединяя их отрезками прямой, мы получим изображение кривой. На рис. 4.1 показаны два разных точечных представления одной кривой. Точки на кривой 4.1a расположены равномерно по ее длине. В результате получается довольно плохое представление кривой, особенно если мал радиус кривизны. Улучшить вид можно, увеличивая плотность точек в таких участках, рис. 4.1b.

Хотя, как показано выше, кривые могут быть достаточно хорошо представлены совокупностью точек, аналитическое представление во многих отношениях предпочтительнее. Его преимущества — это точность, компактность записи и простота вычисления промежуточных точек. Аналитическое представление позволяет без труда определить наклон и радиус кривизны, а при точечном представле-

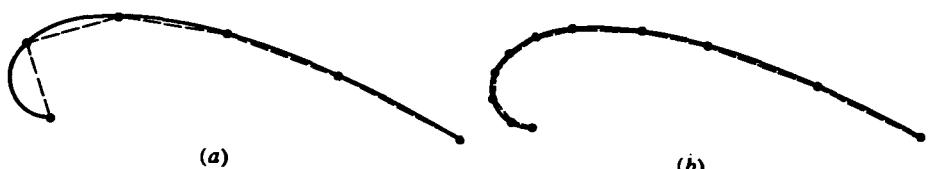


Рис. 4-1 Точечное представление кривых. (a) Равномерная плотность точек вдоль кривой; (b) плотность точек возрастает с уменьшением радиуса кривизны.

нии для этого требуется численное дифференцирование — чрезвычайно неточная процедура. Сравните объем памяти для точечного представления окружности с 32 точками на ней и аналитическое представление — координаты центра и радиус. При аналитическом представлении кривой можно точно определить положение любой точки, а при точечном нужна интерполяция, причем в общем случае результат интерполяции не принадлежит кривой.

Опыт показывает, что аналитически представленные кривые легче изобразить на рисунке. Аналитическое представление выгоднее, когда для удовлетворения заданному критерию необходимо непрерывно изменять форму кривой. Ниже рассматриваются различные методы аналитического представления двумерных кривых.

Часто требуется аналитическое представление кривой, первоначально заданной точками. С математической точки зрения это проблема интерполяции. Для того чтобы провести кривую через все заданные точки, применяется метод кусочной полиномиальной аппроксимации. Для этого требуется определить коэффициенты полинома некоторой степени. Вид кривой в промежуточных точках зависит от степени полинома и граничных условий. Метод рассматривается в гл. 5.

Если же точки — только приблизительные значения величин, например данные экспериментальных измерений или наблюдений, то нужна кривая, задающая верное направление изменения. В общем случае кривая может не проходить ни через одну точку данных. Здесь применяется метод наименьших квадратов [4-1]. Метод наименьших квадратов дает кривую в виде $y = f(x)$, которая минимизирует сумму квадратов отклонений по оси y между данными и полученной кривой. Вид $y = f(x)$ выбирают, исходя из характера изучаемого процесса.

Обычно для метода наименьших квадратов используются полиномы, степенные и экспоненциальные функции: $y = ax^b$, $y = ae^{bx}$ или $y = c_1 + c_2x + c_3x^2 + \dots + c_{n+1}x^n$, где a , b и c_i — константы. Независимо от выбора вида кривой метод требует решения системы линейных алгебраических уравнений для определения неизвестных констант. Более подробное описание приведено в работе [4-1] и книгах по численному анализу.

4-3 НЕПАРАМЕТРИЧЕСКИЕ КРИВЫЕ

Математически кривая может быть представлена в параметрической или непараметрической форме. Непараметрическая кривая задается в виде явной или неявной функции. Для плоской кривой явное непараметрическое представление имеет вид:

$$y = f(x).$$

Пример — уравнение прямой, $y = mx + b$. При этом одному значению x соответствует только одно значение y , поэтому замкнутые или многозначные кривые, например окружность, явно представить нельзя. Неявное представление

$$f(x, y) = 0$$

позволяет обойти это ограничение.

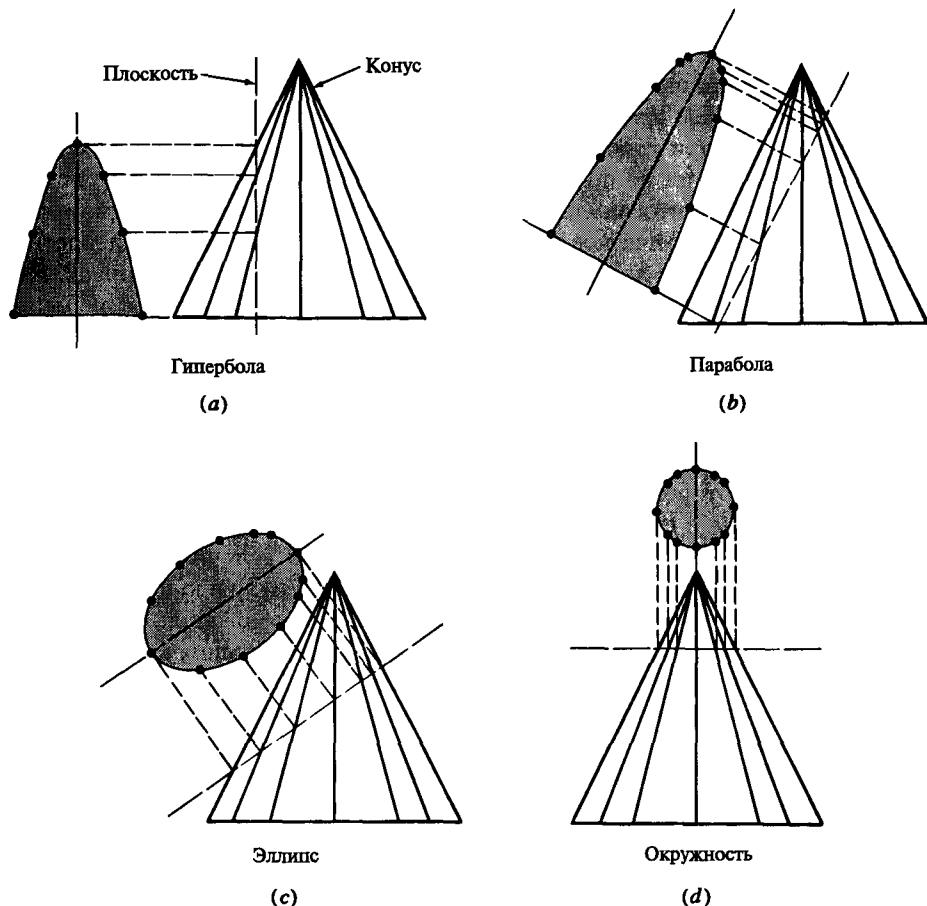


Рис. 4-2 Конические сечения.

Общий вид неявного уравнения второй степени

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$$

порождает различные двумерные кривые, называемые коническими сечениями. На рис. 4.2 изображены три вида конических сечений — парабола, гипербола и эллипс. Окружность — это частный случай эллипса. Определяя коэффициенты a , b , c , d , e и f , можно получить разные конические сечения. Если сечение задано относительно локальной системы координат и проходит через ее начало, то $f = 0$. Для того чтобы провести кривую через данные точки, используются граничные условия.

Пусть $c = 1.0$, тогда сегмент кривой между двумя точками определяется пятью независимыми условиями, из которых вычисляются оставшиеся коэффициенты a , b , d , e и f . Например, можно указать положение крайних точек, наклон кривой в них и промежуточную точку на кривой.

Если $b = 0$ и $c = 1.0$, то аналитическое представление кривой получается с помощью только четырех дополнительных условий, например положения концевых точек и наклона кривой в них. Кривая при $a = 1.0$, $b = 0$ и $c = 1.0$ еще проще:

$$x^2 + y^2 + 2dx + 2ey + f = 0.$$

Тремя условиями для вычисления d , e и f могут быть две концевые точки и наклон кривой в одной из них или же две концевые точки и третья точка на кривой.

При $a = b = c = 0$ получается прямая линия. Ее уравнение

$$dx + ey = f = 0$$

или

$$y = -\left(\frac{d}{e}\right)x - \frac{f}{e} = mx + b',$$

где m — наклон линии, b' — пересечение с осью y .

Как явное, так и неявное непараметрическое представление осезависимо, т. е. сложность обработки зависит от выбора системы координат. Например, если в заданной системе координат граничным условием является вертикальный наклон, нужно либо изменить ее, либо аппроксимировать бесконечный коэффициент наклона наибольшей допустимой положительной или отрицательной величиной.

Кроме того, если точки на осезависимой непараметрической кривой вычисляются с равномерным приращением по x или y , они не будут равномерно распределены вдоль кривой. Это может повлиять на качество и точность графического изображения. Тем не менее непараметрическое представление бывает полезным. Теперь рассмотрим параметрическое представление, позволяющее обойти эти ограничения.

4-4 ПАРАМЕТРИЧЕСКИЕ КРИВЫЕ

В параметрическом виде каждая координата точки кривой представлена как функция одного параметра. Значение параметра задает координатный вектор точки на кривой. Для двумерной кривой с параметром t координаты точки равны:

$$\begin{aligned} x &= x(t), \\ y &= y(t). \end{aligned}$$

Тогда векторное представление точки на кривой:

$$P(t) = [x(t) \quad y(t)].$$

Чтобы получить непараметрическую форму, нужно исключить t из двух уравнений и вывести одно в терминах x и y .

Параметрическая форма позволяет представить замкнутые и многозначные кривые. Производная, т. е. касательный вектор, есть

$$P'(t) = [x'(t) \quad y'(t)],$$

где ' обозначает дифференцирование по параметру. Наклон кривой, dy/dx , равен

$$\frac{dy}{dx} = \frac{dy/dt}{dx/dt} = \frac{y'(t)}{x'(t)}.$$

Отметим, что при $x'(t) = 0$ наклон бесконечен. Параметрическое представление не вызывает в этом случае вычислительных трудностей, достаточно приравнять нулю одну компоненту касательного вектора.

Так как точка на параметрической кривой определяется только значением параметра, эта форма не зависит от выбора системы координат. Конечные точки и длина кривой определяются диапазоном изменения параметра. Часто бывает удобно нормализовать параметр на интересующем отрезке кривой к $0 \leq t \leq 1$. Осенезависимость параметрической кривой позволяет с легкостью проводить с ней аффинные преобразования, рассмотренные в гл. 2 и 3.

Самое простое параметрическое представление у прямой. Для двух векторов положения P_1 и P_2 параметрический вид отрезка прямой между ними такой:

$$P(t) = P_1 + (P_2 - P_1)t, \quad 0 \leq t \leq 1.$$

Так как $P(t)$ это вектор, у каждой его составляющей есть параметрическое представление $x(t)$ и $y(t)$ между P_1 и P_2 :

$$\begin{aligned} x(t) &= x_1 + (x_2 - x_1)t, & 0 \leq t \leq 1. \\ y(t) &= y_1 + (y_2 - y_1)t, \end{aligned}$$

Пример 4-1 Параметрическое представление прямой

Найти параметрическое представление отрезка между точками $P_1[1 \ 2]$ и $P_2[4 \ 3]$, касательный вектор и наклон.

Параметрическое представление:

$$\begin{aligned} P(t) &= P_1 + (P_2 - P_1)t = [1 \ 2] + ([4 \ 3] - [1 \ 2])t, & 0 \leq t \leq 1, \\ P(t) &= [1 \ 2] + [3 \ 1]t, & 0 \leq t \leq 1. \end{aligned}$$

Параметрическое представление составляющих x и y :

$$\begin{aligned} x(t) &= x_1 + (x_2 - x_1)t = 1 + 3t, & 0 \leq t \leq 1, \\ y(t) &= y_1 + (y_2 - y_1)t = 2 + t. \end{aligned}$$

Дифференцируя $P(t)$, получим касательный вектор:

$$P'(t) = [x'(t) \ y'(t)] = [3 \ 1]$$

или

$$\bar{V}_t = 3\mathbf{i} + \mathbf{j},$$

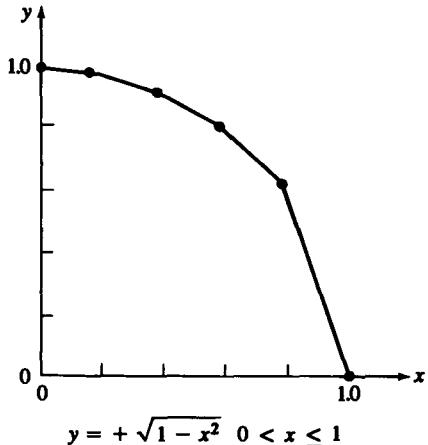
где \bar{V}_t — касательный вектор, а \mathbf{i}, \mathbf{j} — единичные векторы в направлениях x , y соответственно.

Наклон отрезка равен

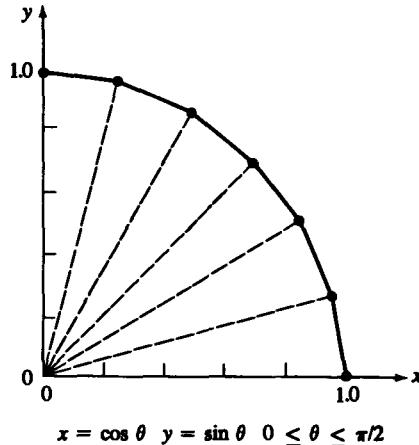
$$\frac{dy}{dx} = \frac{dy/dt}{dx/dt} = \frac{y'(t)}{x'(t)} = \frac{1}{3}.$$

На рис. 4.3 сравниваются непараметрическое и параметрическое представления окружности в первом квадранте. Непараметрический вид

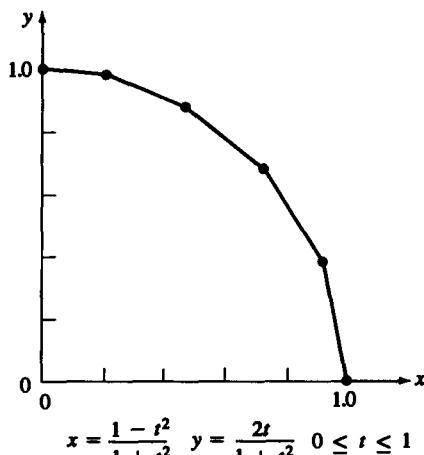
$$y = +\sqrt{1 - x^2}, \quad 0 \leq x \leq 1, \quad (4-1)$$



(a)



(b)



(c)

Рис. 4-3 Представление окружности для первого квадранта.

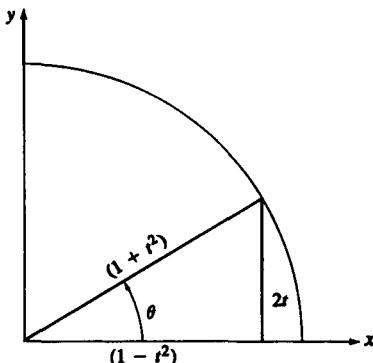


Рис. 4-4 Связь между параметрическими представлениями.

показан на рис. 4.3 а. Точки на дуге соответствуют равным приращениям x . При этом дуга состоит из отрезков разной длины, и получается весьма приблизительное графическое представление окружности. Кроме того, расчет квадратного корня — вычислительно дорогостоящая операция.

Стандартная параметрическая форма единичной окружности:

$$\begin{aligned}x &= \cos \theta, & 0 \leq \theta \leq 2\pi, \\y &= \sin \theta,\end{aligned}$$

или

$$P(\theta) = [x \quad y] = [\cos \theta \quad \sin \theta], \quad 0 \leq \theta \leq 2\pi, \quad (4-2)$$

где параметр θ — геометрический угол, отмеряемый против часовой стрелки от положительной полуоси x . На рис. 3.4 б изображена дуга, построенная по равным приращениям параметра в пределах $0 \leq \theta \leq \pi/2$. При этом точки располагаются на одинаковом расстоянии вдоль окружности, и окружность выглядит гораздо лучше. Недостаток такого представления — сложность вычисления тригонометрических функций. (Более простой метод рассматривается ниже в разд. 4.5.)

Параметрическое представление кривой не единственное, например,

$$P(t) = \left[\frac{(1-t^2)}{(1+t^2)} \quad \frac{2t}{(1+t^2)} \right], \quad 0 \leq t \leq 1 \quad (4-3)$$

также представляет дугу единичной окружности в первом квадранте (рис. 4.3 с). Связь между параметрическим представлением (3.4) и стандартным параметрическим представлением (4.2) показана на рис. 4.4. Из него видно, что для единичной окружности

$$\begin{aligned}x &= \cos \theta = \frac{1-t^2}{1+t^2}, & 0 \leq \theta \leq \pi/2, & 0 \leq t \leq 1, \\y &= \sin \theta = \frac{2t}{1+t^2}, & 0 \leq \theta \leq \pi/2, & 0 \leq t \leq 1.\end{aligned}$$

Факт, что уравнение 4.3 представляет дугу единичной окружности, подтверждается следующим:

$$r^2 = x^2 + y^2 = \left(\frac{1-t^2}{1+t^2}\right)^2 + \left(\frac{2t}{1+t^2}\right)^2 = \frac{1-2t^2+t^4+4t^2}{(1+t^2)^2} = \frac{(1+t^2)^2}{(1+t^2)^2} = 1,$$

где r — единичный радиус.

На рис. 4-3с показан результат для равных приращений t . Он лучше, чем у явного (4-1), но хуже, чем у стандартного параметрического представления (4-2). Однако уравнение (4-3) проще с вычислительной точки зрения, т. е. это компромиссное решение.

Пример 4-2 Определение точки на параметрической кривой

Пусть надо найти y для заданного значения x . Например, пусть $x = 0.5$, и требуется вычислить y для единичной окружности. Для явного представления (4-1) вычисляем непосредственно:

$$y = \sqrt{1-x^2} = \sqrt{1-(0.5)^2} = \sqrt{0.75} = 0.866.$$

Для параметрического представления сначала нужно выразить параметр t через x , а затем с помощью полученного значения найти y . Параметрическое представление уравнения (4-2)

$$x = \cos \theta,$$

$$y = \sin \theta.$$

Поэтому

$$\theta = \arccos(x) = \arccos(0.5) = 60^\circ,$$

$$y = \sin(60^\circ) = 0.866.$$

С другой стороны, из уравнения 4-3

$$x = \frac{1-t^2}{1+t^2},$$

$$y = \frac{2t}{1+t^2}.$$

Решаем первое из этих уравнений относительно

$$t = \left(\frac{1-x}{1+x}\right)^{1/2} = \frac{1}{\sqrt{3}} = 0.57735$$

и получаем

$$y = \frac{2/\sqrt{3}}{4/3} = \frac{\sqrt{3}}{2} = 0.866.$$

В случае более сложного параметрического представления бывает удобнее искать значение явной переменной итеративными методами.

Параметрическое представление конических сечений осенев зависимо и дает более качественное изображение, чем непараметрическое; однако оба имеют свои достоинства и недостатки и часто применяются в машинной графике.

4-5 ПАРАМЕТРИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ОКРУЖНОСТИ

Параметрическое представление окружности радиуса r с центром в начале координат:

$$\begin{aligned}x &= r \cos \theta, & 0 \leq \theta \leq 2\pi, \\y &= r \sin \theta,\end{aligned}\quad (4-4)$$

где θ — параметр. Ранее мы отмечали, что равномерное приращение дает отличное изображение, но алгоритм неэффективен из-за частого вызова тригонометрических функций. В работе [4-2] приводится более быстрый алгоритм.

Полной окружности соответствует диапазон изменения параметра θ от 0 до 2π . Если рассматривать некоторое фиксированное число равномерно распределенных точек по окружности, то приращение параметра между точками можно считать константой. Координаты любой точки на окружности с центром в начале координат

$$\begin{aligned}x_{i+1} &= r \cos(\theta_i + \delta\theta), \\y_{i+1} &= r \sin(\theta_i + \delta\theta),\end{aligned}$$

где θ_i — значение параметра для точки x_i, y_i .

По формуле суммы углов

$$\begin{aligned}x_{i+1} &= r(\cos \theta_i \cos \delta\theta - \sin \theta_i \sin \delta\theta), \\y_{i+1} &= r(\cos \theta_i \sin \delta\theta + \sin \theta_i \cos \delta\theta).\end{aligned}$$

Вспомним уравнение (4-4) для $\theta = \theta_i$

$$\begin{aligned}x_i &= r \cos \theta_i, \\y_i &= r \sin \theta_i\end{aligned}$$

и получим рекурсивные уравнения

$$\begin{aligned}x_{i+1} &= x_i \cos \delta\theta - y_i \sin \delta\theta, \\y_{i+1} &= x_i \sin \delta\theta + y_i \cos \delta\theta,\end{aligned}\quad (4-5)$$

что соответствует повороту точки x_i, y_i на $\delta\theta$.

Так как $\delta\theta$ постоянно и равно $2\pi/(n - 1)$, где n — количество равномерно распределенных по окружности точек, значения \sin и \cos нужно вычислить только один раз. Во внутреннем цикле используются только четыре умножения, вычитание и сложение, поэтому алгоритм работает очень быстро¹. Результат на рис. 4-5 соответствует расчетам по формуле (4-4).

Окружность с центром в произвольной точке получается переносом окружности соответствующего радиуса с центром в начале координат. В некоторых случаях можно упростить задачу: сначала строить единичную окружность с центром в начале координат, а затем, комбинируя перенос и масштабирование, получить окружности с любым радиусом и центром.

¹ Замечание. Как и для всех рекурсивных алгоритмов результаты при больших n могут сильно зависеть от конкретных данных.

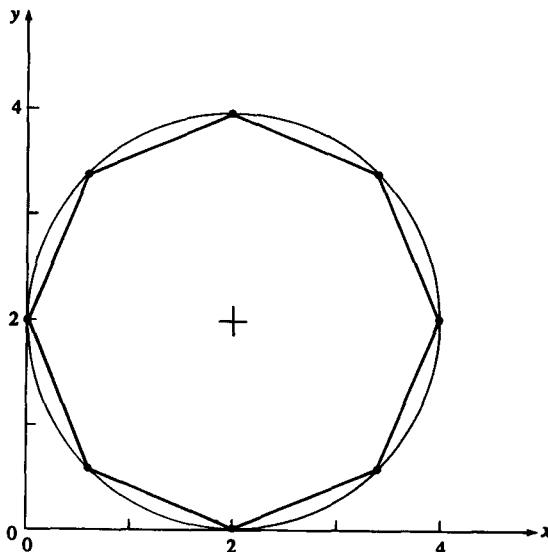


Рис. 4-5 Единичная параметрическая окружность с $n = 8$.

Пример 4-3 Генерация параметрической окружности

Построить окружность радиуса 2 с центром в точке (2, 2). Рассмотрим два метода. Первый: построить окружность радиуса 2 с центром в начале координат и перенести на 2 единицы по x и y . Второй: построить единичную окружность с центром в начале координат, увеличить в два раза и, наконец, перенести. Воспользуемся вторым методом. Пусть для простоты на окружности лежат восемь точек, хотя обычно требуется гораздо большее количество. На самом деле количество точек зависит от радиуса.

Окружность — это замкнутая кривая, поэтому первая ($\theta = 0$) и последняя ($\theta = 2\pi$) точки совпадают. Следовательно, чтобы получить n различных точек на окружности, нужно вычислить $n + 1$ точку. Для незамкнутых кривых это необязательно.

Найдем $\delta\theta$:

$$\delta\theta = \frac{2\pi}{(n+1-1)} = \frac{2\pi}{n} = \frac{2\pi}{8} = \frac{\pi}{4}.$$

Исходные значения x и y из уравнения (4-4) при $\theta = 0$ таковы:

$$x_1 = r \cos \theta_1 = (1) \cos(0) = 1,$$

$$y_1 = r \sin \theta_1 = (1) \sin(0) = 0.$$

Теперь по формуле (4-5) получим остальные семь точек. Для первой

$$\sin \delta\theta = \sin \frac{\pi}{4} = \frac{\sqrt{2}}{2},$$

$$\cos \delta\theta = \cos \frac{\pi}{4} = \frac{\sqrt{2}}{2}$$

Таблица 4-1 Результаты для единичной окружности

<i>i</i>	<i>x_i</i>	<i>y_i</i>
1	1	0
2	$\sqrt{2}/2$	$\sqrt{2}/2$
3	0	1
4	$-\sqrt{2}/2$	$\sqrt{2}/2$
5	-1	0
6	$-\sqrt{2}/2$	$-\sqrt{2}/2$
7	0	-1
8	$\sqrt{2}/2$	$-\sqrt{2}/2$
9	1	0

и

$$x_2 = x_1 \cos \delta\theta - y_1 \sin \delta\theta = (1)(\sqrt{2}/2) - 0(\sqrt{2}/2) = (\sqrt{2}/2),$$

$$y_2 = x_1 \sin \delta\theta + y_1 \cos \delta\theta = (1)(\sqrt{2}/2) + 0(\sqrt{2}/2) = (\sqrt{2}/2).$$

Результаты для остальных точек собраны в таблице 4-1.

Вспоминая двумерные преобразования из гл. 2, определим 3×3 преобразование, которое масштабирует полученные результаты с коэффициентом 2 и переносит центр окружности в точку $(2, 2)$:

$$[T] = [S][Tr_{xy}] = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 1 \end{bmatrix}.$$

Применяя это преобразование к точке (x, y) , получаем

$$\begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} [T] = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 1 \end{bmatrix},$$

как и ожидалось. Полный результат приведен в таблице 4-2 и на рис. 4-5.

Таблица 4-2 Результаты для окружности радиуса 2 с центром в $(2, 2)$

<i>i</i>	<i>x_i</i>	<i>y_i</i>
1	4	2
2	3.414	3.414
3	2	4
4	0.586	3.414
5	0	2
6	0.586	0.586
7	2	0
8	3.414	0.586
9	4	2

Ограничиваая диапазон параметра θ , можно получить дуги окружности. Например, $0 \leq \theta \leq \pi/2$ соответствует четверти круга в первом квадранте, а $\pi \leq \theta \leq 3\pi/2$ — четверти круга в третьем квадранте.

4-6 ПАРАМЕТРИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ЭЛЛИПСА

Достаточно хорошее представление окружности получается, если соединить отрезками некоторое количество равномерно распределенных на ней точек. Однако, если рассчитывать точки эллипса через равные приращения угла, изображение будет неверным, как показано штриховой линией на рис. 4.6. Особенно сильно неточности проявляются на концах, где кривизна слишком велика и требуется большее количество точек.

Другой метод основан на равных приращениях по периметру и дает хороший результат для достаточно большого количества точек. Недостатки его в том, что указывается слишком много точек на сторонах с малой кривизной, и вычисление равных частей периметра требует сложного расчета эллиптического интеграла. Что нам необходимо, это малые приращения параметра у концов, где кривизна велика, и большие приращения параметра вдоль сторон с малой кривизной.

Такое распределение точек получается из параметрического представления эллипса с центром в начале координат, большой полуосью a и малой полуосью b :

$$\begin{aligned} x &= a \cos \theta, \\ y &= b \sin \theta, \end{aligned} \tag{4-6}$$

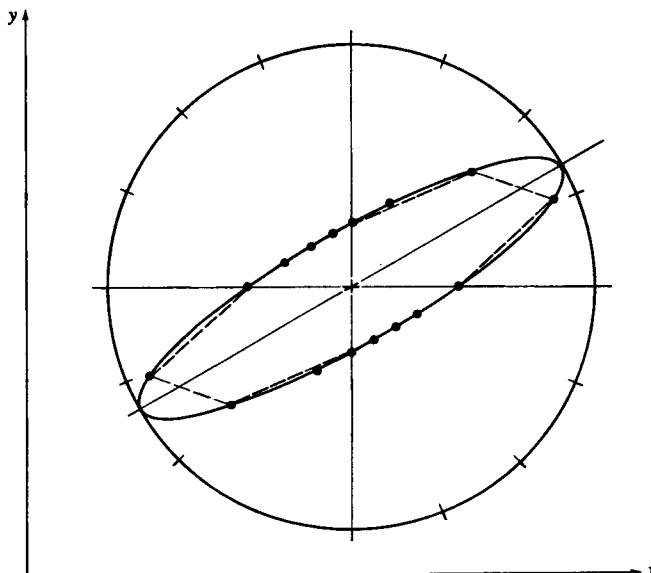


Рис. 4-6 Равноугольное представление сильно вытянутого эллипса.

где θ — параметр. Диапазон от 0 до 2π задает полный эллипс. Требуемое распределение точек порождается равномерными приращениями параметра θ .

Рассматривая производные x и y :

$$\begin{aligned} dx &= -a \sin \theta d\theta, \\ dy &= b \cos \theta d\theta, \end{aligned} \quad (4-7)$$

получаем, что при θ , близком к 0 или π , т. е. у концов, $|dx| \approx 0$ и $|dy| \approx bd\theta$. Если θ близко к $\pi/2$ или $3\pi/2$, т. е. вдоль сторон, то $|dx| \approx ad\theta$ и $|dy| \approx 0$. Таким образом, около концов, где кривизна более высокая, точки располагаются чаще, а вдоль сторон, где кривизна меньше, — реже. Отношение приращений периметра концов к приращениям вдоль сторон приблизительно равно b/a . Отметим, что для окружности ($b = a$) достигается оптимальное представление — равные приращения параметра или угла.

Если задано фиксированное количество точек на эллипсе, можно, пользуясь формулами суммы углов, получить эффективный алгоритм [4-2]. Координаты любой точки на эллипсе:

$$\begin{aligned} x_{i+1} &= a \cos(\theta_i + \delta\theta), \\ y_{i+1} &= b \sin(\theta_i + \delta\theta), \end{aligned}$$

где $\delta\theta = 2\pi/(n - 1)$ — фиксированное приращение θ , n — количество точек на периметре, θ_i — значение параметра для точки x_i , y_i .

По формуле суммы углов

$$\begin{aligned} x_{i+1} &= a(\cos \theta_i \cos \delta\theta - \sin \theta_i \sin \delta\theta), \\ y_{i+1} &= b(\cos \theta_i \sin \delta\theta + \sin \theta_i \cos \delta\theta). \end{aligned}$$

Применяя уравнение (4-4) с $\theta = \theta_i$, перепишем эти уравнения:

$$\begin{aligned} x_{i+1} &= x_i \cos \delta\theta - \left(\frac{a}{b}\right) y_i \sin \delta\theta, \\ y_{i+1} &= \left(\frac{b}{a}\right) x_i \sin \delta\theta + y_i \cos \delta\theta, \end{aligned} \quad (4-8)$$

Так как $\delta\theta$, a и b — константы, полученный алгоритм содержит во внутреннем цикле только четыре умножения, одно сложение и вычитание и достаточно эффективен. В работе [4-2] показано, что алгоритм дает многоугольник максимальной площади, вписанный в эллипс. Результат представлен на рис. 4-7.

Чтобы получить эллипс с центром не в начале координат и с главной осью, расположенной под углом к горизонтали, его поворачивают вокруг начала координат, а затем переносят.

Пример 4-4 Параметрическое построение эллипса

Построить эллипс с большой полуосью $a = 4$ и малой полуосью $b = 1$, под углом 30° к горизонтали, с центром в точке $(2, 2)$.

Сначала построим эллипс с центром в начале координат с помощью 32 точек ($n = 33$, так как первая и последняя точки совпадают). Мы рассмотрим только точки первого квадранта, т. е. диапазон параметра от 0 до $\pi/2$.

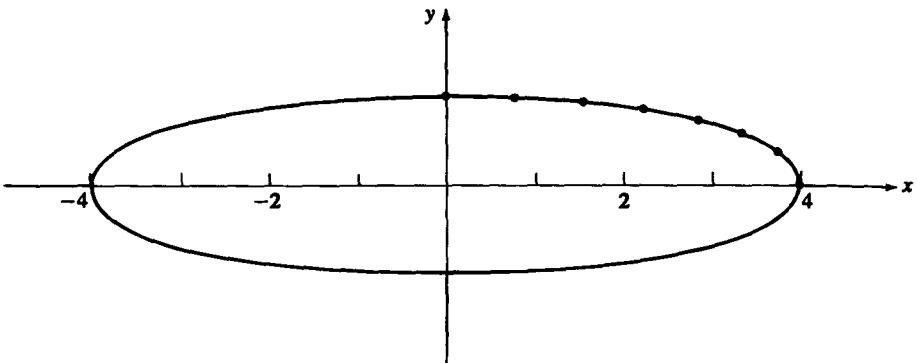


Рис. 4-7 Параметрическое представление эллипса.

Приращение параметра

$$\delta\theta = \frac{2\pi}{(n-1)} = \frac{2\pi}{32} = \frac{\pi}{16}.$$

Пользуясь уравнением (4-6) с начальным значением $\theta = 0$, найдем x и y :

$$x_1 = a \cos \theta_1 = (4) \cos(0) = 4,$$

$$y_1 = b \sin \theta_1 = (1) \sin(0) = 0.$$

Вычислим величины $a/b = 4$, $b/a = 1/4$ и

$$\sin \delta\theta = \sin \frac{\pi}{16} = 0.195,$$

$$\cos \delta\theta = \cos \frac{\pi}{16} = 0.981.$$

Теперь по формуле (4-8) найдем вторую точку

$$x_2 = x_1 \cos \delta\theta - \left(\frac{a}{b}\right) y_1 \sin \delta\theta = (4)(0.981) - (4)(0)(0.195) = 3.92,$$

$$y_2 = \left(\frac{b}{a}\right) x_1 \sin \delta\theta + y_1 \cos \delta\theta = (1/4)(4)(0.195) + (0)(0.981) = 0.195.$$

Результаты для оставшихся точек первого квадранта представлены в таблице 4-3 и на рис. 4-7.

Нам требуется комбинированное двумерное преобразование (гл. 2) поворота на 30° вокруг центра координат и переноса центра в точку $(2, 2)$.

Характерно, что

$$\begin{aligned}
 [T] &= [R][Tr_{xy}] = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 0.866 & 0.5 & 0 \\ -0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} 0.866 & 0.5 & 0 \\ -0.5 & 0.866 & 0 \\ 2 & 2 & 1 \end{bmatrix}.
 \end{aligned}$$

Таблица 4-3 Точки эллипса с центром в начале координат для первого квадранта

i	x_i	y_i
1	4.0	0
2	3.923	0.195
3	3.696	0.383
4	3.326	0.556
5	2.828	0.707
6	2.222	0.831
7	1.531	0.924
8	0.780	0.981
9	0	1.0

Применим это преобразование для точек (x_1, y_1) и (x_2, y_2) :

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix} [T] = \begin{bmatrix} 4 & 0 & 1 \\ 3.923 & 0.195 & 1 \end{bmatrix} \begin{bmatrix} 0.866 & 0.5 & 0 \\ -0.5 & 0.866 & 0 \\ 2 & 2 & 1 \end{bmatrix} = \\ = \begin{bmatrix} 5.464 & 4 & 1 \\ 5.3 & 4.131 & 1 \end{bmatrix}.$$

Результаты приведены в табл. 4-4 и на рис. 4-8.

Таблица 4-4 Повернутый и перенесенный эллипс

i	x_i	y_i
1	5.464	4.0
2	5.3	4.131
3	5.009	4.179
4	4.603	4.144
5	4.096	4.027
6	3.509	3.831
7	3.864	3.565
8	2.185	3.240
9	1.5	2.866

4-7 ПАРАМЕТРИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ПАРАБОЛЫ

Рассмотрим параболу с вершиной в центре координат, раскрыв вправо, т. е. с осью симметрии — положительной полуосью x . На рис. 4-9 изображена верхняя ветвь такой параболы. В прямоугольных координатах непараметрическое представление параболы:

$$y^2 = 4ax.$$

Параметрическое представление имеет вид

$$x = \operatorname{tg}^2 \phi,$$

$$y = \pm 2\sqrt{a \operatorname{tg} \phi},$$

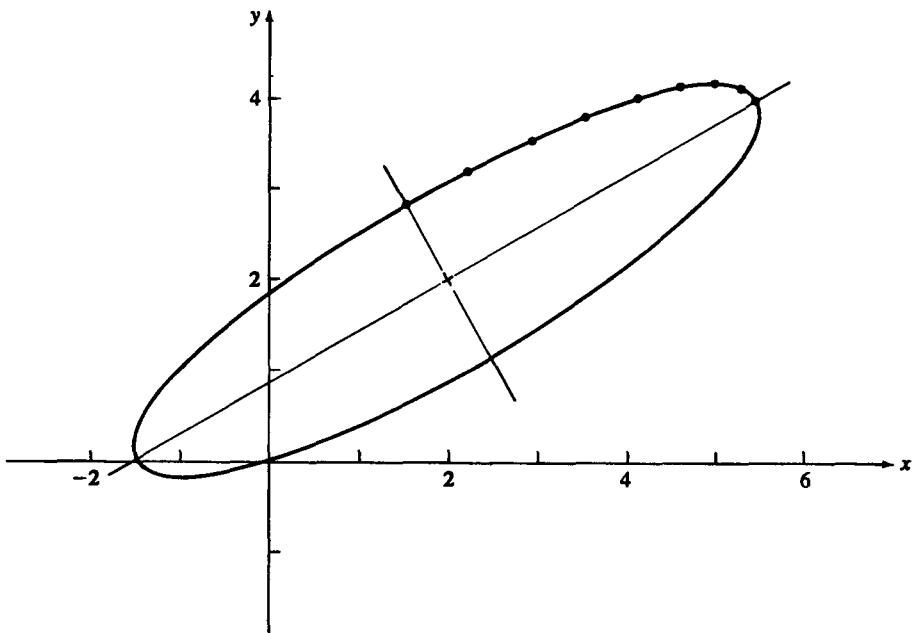


Рис. 4-8 Эллипс после поворота и переноса.

где $0 \leq \phi \leq \pi/2$. Хотя оно обеспечивает достаточно хорошее изображение, Смит [4-2] указал, что получаемая фигура не является фигурой с максимальной вписанной площадью и поэтому это не оптимальный вариант. Другое параметрическое представление действительно дает фигуру с наибольшей вписанной площадью:

$$\begin{aligned}x &= a\theta^2, \\y &= 2a\theta,\end{aligned}\tag{4-9}$$

где $0 \leq \theta \leq \infty$ соответствует всей верхней ветви параболы. В отличие от эллипса парабола не замкнутая кривая, поэтому изображаемая часть должна быть ограничена:

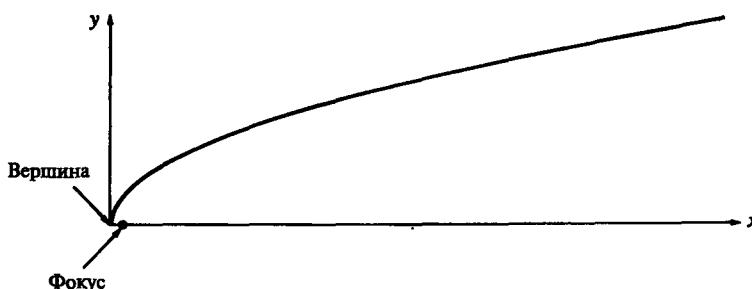


Рис. 4-9 Парабола.

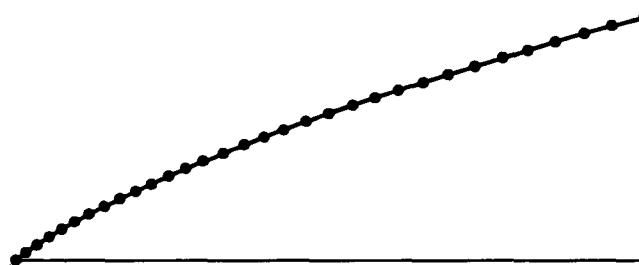


Рис. 4-10 Параметрически сгенерированная парабола.

ничена минимальным и максимальным значением параметра. Это можно сделать несколькими способами. Если диапазон изменения координаты x ограничен, то

$$\theta_{\min} = \sqrt{\frac{x_{\min}}{a}}, \quad \theta_{\max} = \sqrt{\frac{x_{\max}}{a}}. \quad (4-10)$$

Если ограничен диапазон изменения y , то

$$\theta_{\min} = \frac{y_{\min}}{2a}, \quad \theta_{\max} = \frac{y_{\max}}{2a}. \quad (4-11)$$

Установив θ_{\min} и/или θ_{\max} , можно построить параболу в первом квадранте. Параболы в других квадрантах, со смещенным центром и в других ориентациях строятся с помощью отражения, поворота и переноса.

Параболу можно построить также, пользуясь приращениями параметра. Пусть на параболе задано фиксированное количество точек, т. е. приращение параметра θ постоянно. Для $\theta_{i+1} = \theta_i + \delta\theta$ уравнение (4-9) принимает вид

$$\begin{aligned} x_{i+1} &= a\theta_i^2 + 2a\theta_i\delta\theta + a(\delta\theta)^2, \\ y_{i+1} &= 2a\theta_i + 2a\delta\theta. \end{aligned}$$

Используя уравнение (4-9) с $\theta = \theta_i$, перепишем формулы

$$\begin{aligned} x_{i+1} &= x_i + y_i\delta\theta + a(\delta\theta)^2, \\ y_{i+1} &= y_i + 2a\delta\theta. \end{aligned} \quad (4-12)$$

Расчет очередной точки требует трех сложений и одного умножения во внутреннем цикле алгоритма. На рис. 4-10 приведен пример параболы, сгенерированной по рекурсивным формулам (4-12).

Пример 4-5 Параметрическое представление параболы

Построить параболический сегмент в первом квадранте при $1 \leq x \leq 4$ для параболы

$$x = a\theta^2 = \theta^2, \quad y = 2a\theta = 2\theta$$

при $a = 1$.

Сначала найдем границы θ . В соответствии с уравнением (4-10) θ_{\min} и θ_{\max} определяются так:

$$\theta_{\min} = \sqrt{\frac{x_{\min}}{a}} = \sqrt{\frac{1}{1}} = 1,$$

$$\theta_{\max} = \sqrt{\frac{x_{\max}}{a}} = \sqrt{\frac{4}{1}} = 2.$$

Пусть на сегменте расположено 10 точек, тогда

$$\delta\theta = \frac{\theta_{\max} - \theta_{\min}}{n - 1} = \frac{2 - 1}{10 - 1} = \frac{1}{9}.$$

Начиная с $\theta_1 = \theta_{\min}$, $x_1 = 1$, из уравнения (4-9) получаем

$$y_1 = 2a\theta_1 = (2)(1) = 2.$$

Из уравнения (4-12)

$$x_2 = x_1 + y_1 \delta\theta + (\delta\theta)^2 = 1 + (2) \left(\frac{1}{9}\right) + \left(\frac{1}{9}\right)^2 = 1.235,$$

$$y_2 = y_1 + 2\delta\theta = 2 + (2) \left(\frac{1}{9}\right) = 2.222.$$

Окончательный результат приведен в табл. 4-5 и на рис. 4-11.

Таблица 4-5 Результаты для сегмента параболы

<i>i</i>	<i>x_i</i>	<i>y_i</i>
1	1.0	2.0
2	1.235	2.222
3	1.494	2.444
4	1.778	2.667
5	2.086	2.889
6	2.420	3.111
7	2.778	3.333
8	3.160	3.556
9	3.568	3.778
10	4.0	4.0

В некоторых случаях более удобны другие параметрические представления, вид которых зависит от прикладной задачи и имеющихся данных. Например, если надо нарисовать дугу параболы между двумя точками и учитывать наклон, то предлагается следующее представление:

$$\begin{aligned} x(t) &= (Q_x - 2R_x + R_x)t^2 + 2(R_x - P_x)t + P_x, \\ y(t) &= (Q_y - 2R_y + P_y)t^2 + 2(R_y - P_y)t + P_y, \quad 0 \leq t \leq 1.0, \end{aligned} \quad (4-13)$$

где t — параметр, а две конечные точки $P = [P_x \ P_y]$ и $Q = [Q_x \ Q_y]$. Точка $R = [R_x \ R_y]$ это точка пересечения касательных в конечных точках. Три вершины P , Q , R определяют параболу, как показано на рис. 4-12. Более общий метод построения кривых с помощью вершин незамкнутого многоугольника был разработан Безье и рассматривается в следующей главе.

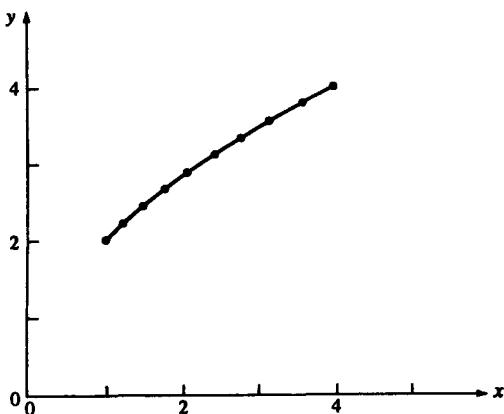


Рис. 4-11 Сегмент параболы.

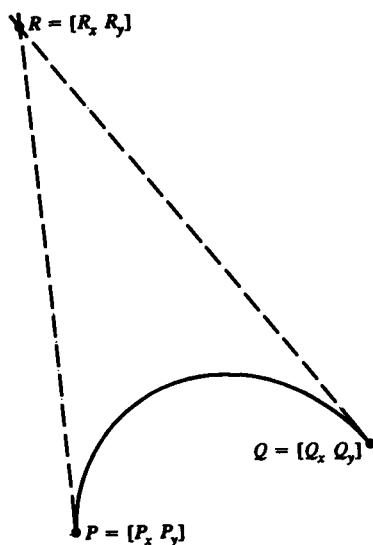


Рис. 4-12 Задание параметрической параболы вершинами.

4-8 ПАРАМЕТРИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ГИПЕРБОЛЫ

Построим гиперболу с центром в начале координат и осью симметрии, совпадающей с осью x . Ее непараметрическое представление в прямоугольных координатах:

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1.$$

При этом вершина находится в точке $(a, 0)$, оси асимптот $\pm b/a$. Вид параметрического представления:

$$\begin{aligned}x &= \pm a \sec \theta, \\y &= \pm b \operatorname{tg} \theta,\end{aligned}\quad (4-14)$$

где $0 \leq \theta \leq \pi/2$, дает искомую гиперболу. Смит [4-2] указывает, что для такого представления площадь вписанного многоугольника не максимальна. Однако она близка к максимальной, и с помощью формулы суммы углов можно получить эффективный алгоритм. Вспомним, что

$$\sec(\theta + \delta\theta) = \frac{1}{\cos(\theta + \delta\theta)} = \frac{1}{\cos \theta \cos \delta\theta - \sin \theta \sin \delta\theta}$$

и

$$\operatorname{tg}(\theta + \delta\theta) = \frac{(\operatorname{tg} \theta + \operatorname{tg} \delta\theta)}{1 - \operatorname{tg} \theta \operatorname{tg} \delta\theta}.$$

Подставим в уравнения (4-14)

$$\begin{aligned}x_{i+1} &= \pm a \sec(\theta + \delta\theta) = \pm \frac{ab / \cos \theta}{b \cos \delta\theta - b \operatorname{tg} \theta \sin \delta\theta}, \\y_{i+1} &= \pm b \operatorname{tg}(\theta + \delta\theta) = \pm \frac{b \operatorname{tg} \theta + b \operatorname{tg} \delta\theta}{1 - \operatorname{tg} \theta \operatorname{tg} \delta\theta}.\end{aligned}$$

Используя уравнения (4-14) с $\theta = \theta_i$, перепишем эти уравнения как

$$\begin{aligned}x_{i+1} &= \pm \frac{bx_i}{b \cos \delta\theta - y_i \sin \delta\theta}, \\y_{i+1} &= \pm \frac{b(y_i + b \operatorname{tg} \delta\theta)}{b - y_i \operatorname{tg} \delta\theta}.\end{aligned}\quad (4-15)$$

Другое параметрическое представление гиперболы, дающее максимальную вписанную площадь:

$$\begin{aligned}x &= a \operatorname{ch} \theta, \\y &= b \operatorname{sh} \theta.\end{aligned}\quad (4-16)$$

Гиперболические функции определяются как $\operatorname{ch} \theta = (e^\theta + e^{-\theta})/2$ и $\operatorname{sh} \theta = (e^\theta - e^{-\theta})/2$. При изменении от 0 до бесконечности проходится вся гипербола. Формула для суммы углов для и

$$\begin{aligned}\operatorname{ch}(\theta + \delta\theta) &= \operatorname{ch} \theta \operatorname{ch} \delta\theta + \operatorname{sh} \theta \operatorname{sh} \delta\theta, \\\operatorname{sh}(\theta + \delta\theta) &= \operatorname{sh} \theta \operatorname{ch} \delta\theta + \operatorname{ch} \theta \operatorname{sh} \delta\theta.\end{aligned}$$

Это позволяет записать уравнения (4-16) как

$$\begin{aligned}x_{i+1} &= a(\operatorname{ch} \theta \operatorname{ch} \delta\theta + \operatorname{sh} \theta \operatorname{sh} \delta\theta), \\y_{i+1} &= b(\operatorname{sh} \theta \operatorname{ch} \delta\theta + \operatorname{ch} \theta \operatorname{sh} \delta\theta)\end{aligned}$$

или

$$\begin{aligned}x_{i+1} &= x_i \operatorname{ch} \delta\theta + \left(\frac{a}{b}\right) y_i \operatorname{sh} \delta\theta, \\y_{i+1} &= \left(\frac{a}{b}\right) x_i \operatorname{sh} \delta\theta + y_i \operatorname{ch} \delta\theta.\end{aligned}\quad (4-17)$$

Чтобы ограничить область гиперболы, необходимо установить минимальное и максимальное значения. Пусть ветвь гиперболы лежит в первом и четвертом квадранте и рассматривается часть при $x_{\min} \leq x \leq x_{\max}$. Тогда

$$\begin{aligned}\theta_{\min} &= \operatorname{Arch} \left(\frac{x_{\min}}{a} \right), \\ \theta_{\max} &= \operatorname{Arch} \left(\frac{x_{\max}}{a} \right),\end{aligned}\quad (4-18)$$

где обратный гиперболический косинус получен как

$$\operatorname{Arch} x = \ln(x + \sqrt{x^2 - 1}). \quad (4-19)$$

Остальные границы определяются аналогично. Пример части гиперболы в первом квадранте, полученной этим методом, показан на рис. 4-13.

Пример 4-6 Параметрическая гипербола

С помощью параметрического представления (4-16) найти восемь точек на сегменте гиперболы в первом квадранте при $a = 2$, $b = 1$, для $4 \leq x \leq 8$.

Сначала определим границы параметра из уравнений (4-18) и (4-19)

$$\begin{aligned}\theta_{\max} &= \operatorname{Arch} \left(\frac{x_{\max}}{a} \right) = \ln \left(\frac{x_{\max}}{a} + \sqrt{\left(\frac{x_{\max}}{a} \right)^2 - 1} \right) = \\&= \ln(4 + \sqrt{16 - 1}) \\&= 2.063.\end{aligned}$$

Аналогично,

$$\theta_{\min} = 1.317.$$

Итак,

$$\delta\theta = \frac{\theta_{\max} - \theta_{\min}}{n - 1} = \frac{2.063 - 1.317}{7} = 0.107$$

и

$$\begin{aligned}\operatorname{sh}(\delta\theta) &= \operatorname{sh}(0.107) = \\&= \frac{e^{0.107} - e^{-0.107}}{2} = \\&= 0.107, \\ \operatorname{ch}(\delta\theta) &= \operatorname{ch}(0.107) = \\&= \frac{e^{0.107} + e^{-0.107}}{2} = \\&= 1.006.\end{aligned}$$

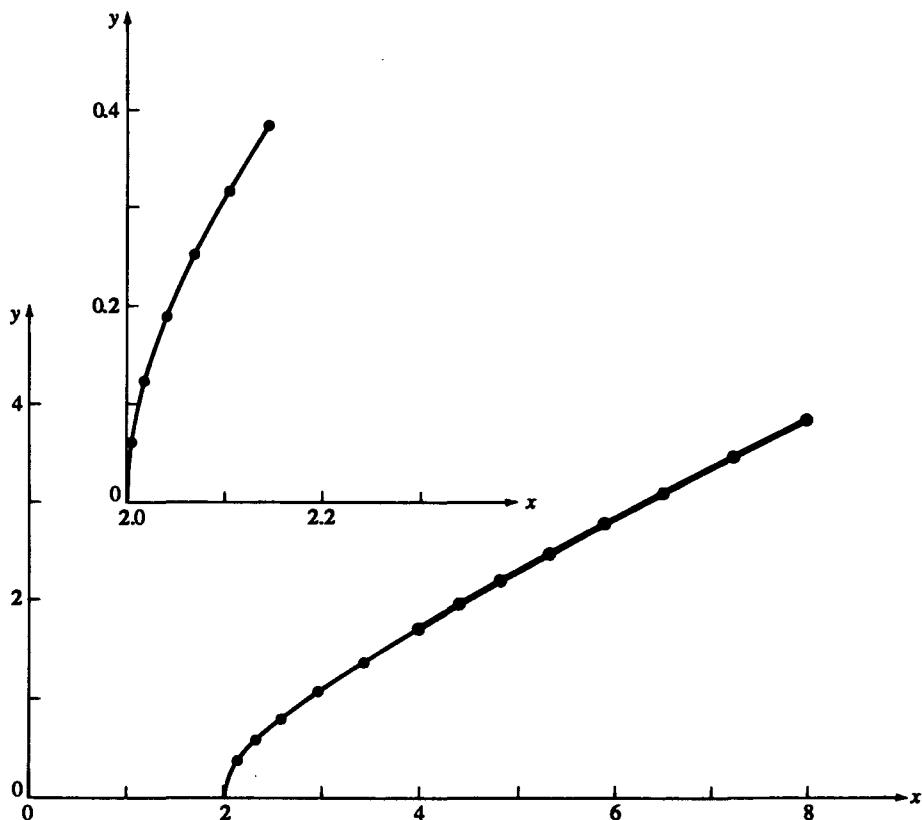


Рис. 4-13 Параметрическая гипербола.

Уравнения (4-16) с $\theta_1 = \theta_{\min}$ дают

$$\begin{aligned}x_1 &= a \operatorname{ch}(\theta_{\min}) = \\&= (2) \operatorname{ch}(1.317) = \\&= 4.000,\end{aligned}$$

$$\begin{aligned}y_1 &= b \operatorname{sh}(\theta_{\min}) = \\&= (1) \operatorname{sh}(1.317) = \\&= 1.732.\end{aligned}$$

Затем из уравнений (4-17)

$$\begin{aligned}x_2 &= x_1 \operatorname{ch}(\delta\theta) + (a/b)y_1 \operatorname{sh}(\delta\theta) \\&= (4)(1.006) + (2)(1.732)(0.107) \\&= 4.393.\end{aligned}$$

$$\begin{aligned}y_2 &= (b/a)x_1 \operatorname{sh}(\delta\theta) + y_1 \operatorname{ch}(\delta\theta) \\&= (1/2)(4)(0.107) + (1.732)(1.006) = 1.956.\end{aligned}$$

Результаты вычислений приведены в табл. 4-6 и выделены жирной линией на рис. 4-13.

Таблица 4-6 Результаты для сегмента гиперболы

<i>i</i>	<i>x_i</i>	<i>y_i</i>
1	4	1.732
2	4.393	1.956
3	4.836	2.201
4	5.334	2.472
5	5.892	2.771
6	6.518	3.102
7	7.218	3.468
8	8	3.873

4-9 ПРОЦЕДУРА ИСПОЛЬЗОВАНИЯ КОНИЧЕСКИХ СЕЧЕНИЙ

Плоские геометрические фигуры часто используются в прикладной машинной графике, например для разработки деталей или в чертежных системах. Особенно важны здесь проблемы определения положения конического сечения и его пересечения и/или точек касания с другим коническим сечением или прямой. На первый взгляд задача несложная, но для произвольно расположенных на плоскости элементов могут существовать различные решения. Выбор правильного решения не всегда очевиден, что приводит к нелинейным расчетам. Ниже приводятся методы, исключающие нелинейную математику и упрощающие линейные результаты. Кроме того, объясняется, как выбрать необходимое решение, если оно неоднозначно.

Основная идея заключается в том, чтобы методами машинной графики, в частности плоским поворотом и переносом, перевести расчеты в первый квадрант в стандартной конфигурации. Если фигура включает коническое сечение, его центр (для параболы и гиперболы — вершина) располагается в начале координат. В общем случае неизвестные центры, вершины, точки касания и пересечения вычисляются с помощью непараметрических уравнений, а параметрическое представление используется для изображения фигур. Разделение расчета и построения позволяет максимально использовать преимущества обоих представлений.

Рассмотрим подробный пример, чтобы лучше представить себе суть и возможности этого метода. Одна из простейших и часто используемых задач — построить окружность по трем точкам. Чертеж приведен на рис. 4-14, где $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$ заданные точки. Необходимо определить центр и радиус окружности. Для этого нужно решить систему из трех нелинейных уравнений с переменными h , k , координатами центра (x, y) и радиусом R :

$$(x_1 - h)^2 + (y_1 - k)^2 = R^2, \quad (4-20a)$$

$$(x_2 - h)^2 + (y_2 - k)^2 = R^2, \quad (4-20b)$$

$$(x_3 - h)^2 + (y_3 - k)^2 = R^2. \quad (4-20c)$$

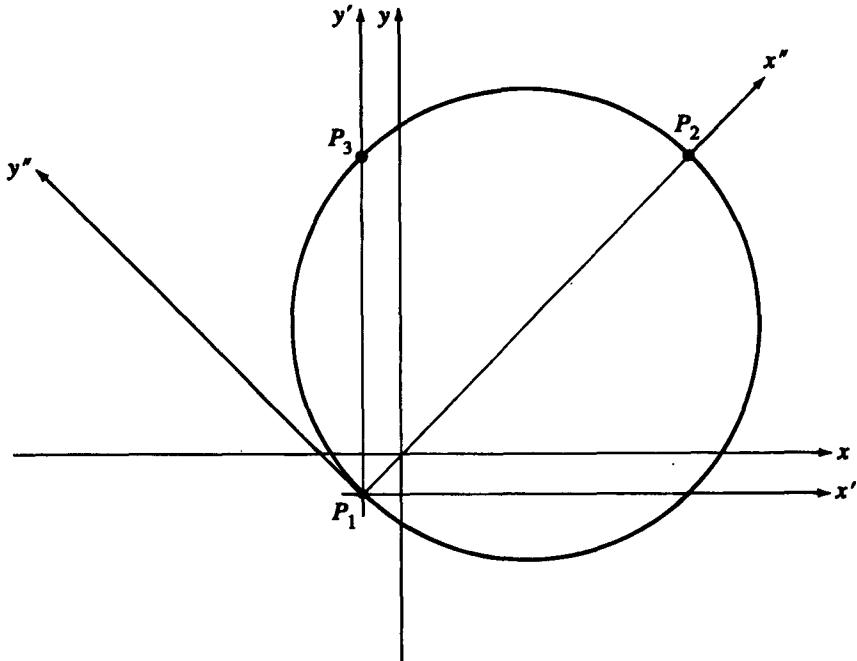


Рис. 4-14 Окружность, проходящая через три заданные точки.

Вычитание уравнения (4-20b) из уравнений (4-20a) и (4-20c) устраняет нелинейность. Собственно алгоритм решения — это $[(a)-(b)](x_3-x_2)-[(c)-(b)](x_1-x_2)$, где буквы в скобках взяты из уравнений (4-20). Уравнение, выраженное через неизвестную, таково:

$$k = \frac{[(x_1^2 - x_2^2) + (y_1^2 - y_2^2)](x_3 - x_2)}{2[(y_1 - y_2)(x_3 - x_2) - (y_3 - y_2)(x_1 - x_2)]} - \frac{[(x_3^2 - x_2^2) + (y_3^2 - y_2^2)](x_1 - x_2)}{2[(y_1 - y_2)(x_3 - x_2) - (y_3 - y_2)(x_1 - x_2)]}. \quad (4-21)$$

Аналогично из $[(a) - (b)]$ и уравнения (4-21) вычисляется

$$h = \frac{(x_1^2 - x_2^2) + (y_1^2 - y_2^2) - 2k(y_1 - y_2)}{2(x_1 - x_2)}, \quad (4-22)$$

R находится из любого из уравнений (4-20).

Уравнения (4-21) и (4-22) содержат ряд трудностей. В частности, если знаменатель дроби равен нулю, требуются другие методы решения. Также необходимы условия проверки на бесконечный радиус и коллинеарные точки.

Уравнения для h и k можно упростить переносом начала координат в точку (см. рис. 4-14). Тогда

$$h'^2 + k'^2 = R^2, \quad (4-23a)$$

$$(x'_2 - h')^2 + (y'_2 - k')^2 = R^2, \quad (4-23b)$$

$$(x'_3 - h')^2 + (y'_3 - k')^2 = R^2. \quad (4-23c)$$

Вычитая (a) из (b) и (c), получаем систему из двух линейных уравнений. Алгоритм решения: $[(b) - (a)]x'_3 - [(c) - (a)]x'_2$, что дает

$$k' = \frac{[x'_3 x'_2 (x'_2 - x'_3) + y'^2_2 x'_3 - y'^2_3 x'_2]}{2(y'_2 x'_3 - y'_3 x'_2)}, \quad (4-24)$$

$$h' = \frac{x'^2_2 + y'^2_2 - 2y'_2 k'}{2x'_2}, \quad (4-25)$$

$$R = (h'^2 + k'^2)^{\frac{1}{2}}. \quad (4-26)$$

Преобразование h' , k' обратно к первоначальной системе координат дает h , k . При этом знаменатель уравнений (4-24) или (4-25) опять может обратиться в нуль, и требуются условия проверки на бесконечный радиус и коллинеарные точки. Эти трудности преодолеваются с помощью следующей процедуры:

Перенести начало координат в одну из точек.

Повернуть оси координат, чтобы одна из оставшихся точек лежала на положительной полуоси x .

Проверить коллинеарность точек.

Найти центр и радиус окружности в перенесенной и повернутой системе координат.

Повернуть обратно к ориентации промежуточной системы координат.

Перенести начало координат в первоначальное положение и найти центр окружности в исходной системе координат.

На рис. 4-14 начало координат перенесено в точку P_1 , а положительная полуось x'' проходит через точку P_2 . Уравнения в координатах (x'', y'') :

$$h''^2 + k''^2 = R^2, \quad (4-27a)$$

$$(x''_2 - h'')^2 + k''^2 = R^2, \quad (4-27b)$$

$$(x''_3 - h'')^2 + (y''_3 - k'')^2 = R^2. \quad (4-27c)$$

Вычтем уравнение (4-27a) из (4-27b)

$$x''_2^2 - 2x''_2 h'' = 0,$$

или

$$h'' = x''_2 / 2. \quad (4-28)$$

Вычтем уравнение (4-27a) из (4-27c) и подставим уравнение (4-28):

$$k'' = \frac{x''_3}{2y''_3} (x''_3 - x''_2) + \frac{y''_3}{2}, \quad (4-29)$$

а из уравнения (4-27а) получим

$$R = (h''^2 + k''^2)^{1/2}. \quad (4-30)$$

Уравнение (4-29) показывает, что при $y_3'' = 0$ значение k бесконечно, однако это случается только тогда, когда все три точки лежат на одной прямой. Проверка этого условия очень проста.

Провести окружность через три точки достаточно легко. Здесь не требуется выбора из множества решений, как в большинстве геометрических конструкций, включающих конические сечения. Метод выбора предложен в работе Роджерса [4-3].

Роджерс рассматривает левое, правое, ближнее, дальнее, внутреннее и внешнее решение. Левое и правое существуют относительно направления от одной известной точки к другой; ближнее и дальнее — относительно линии пересечения; внутреннее и внешнее — относительно окружности.

Эти понятия справедливы как в ориентации пользователя, т. е. в любом месте пространства, так и в стандартной конфигурации в первом квадранте. Простой пример для иллюстрации понятий левое—правое и ближнее—далнее это определение центра и точек касания окружности известного радиуса R , касающейся двух пересекающихся прямых P и L . На практике задача возникает, например, при изготовлении закругленных углов. Для любого радиуса всегда существует четыре решения. Стратегия выбора такова:

Найти точку пересечения прямых. Если решения нет, прямые параллельны.

Перенести начало координат в точку пересечения.

Повернуть вокруг нового центра координат, чтобы одна из линий совпала с осью x .

Найти угол между другой линией и осью x .

Повернуть вокруг нового начала координат, чтобы положительная полуось x совпала с биссектрисой пересекающихся прямых.

Найти центр и точки касания.

Перевести в первоначальную систему координат.

На рис. 4-15 изображен этот случай в стандартной конфигурации. Две пересекающиеся линии P и L заданы точками P_1, P_2, L_1, L_2 соответственно. Для каждого возможного решения одна из координат центра (h, k) лежит на оси координат. Чтобы выбрать нужное решение, требуется указать, находится ли оно слева или справа от линии P , если смотреть от P_1 и P_2 , и ближнее оно или дальнее относительно линии L с той же точки зрения. На рис. 4-15 C_1 — правое и дальнее, а C_3 — левое и ближнее. Центр

$$h = R / \sin \theta,$$

$$k = 0,$$

где θ — острый угол между положительной осью x и P . Точки касания с линиями P и L

$$x_t = h - R \sin \theta,$$

$$y_t = \pm R \cos \theta.$$

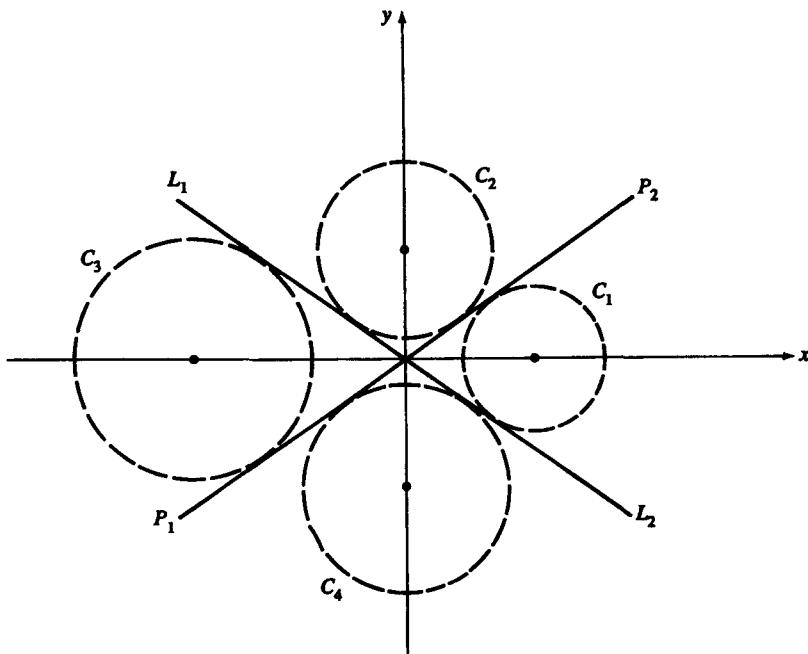


Рис. 4-15 Окружности, касающиеся двух пересекающихся прямых.

Таблица 4-7 Окружность, касающаяся двух пересекающихся линий

Ориентация относительно линии \$P\$	Ориентация относительно линии \$L\$	Центр	Точки касания
Левая	Дальняя	\$h = 0\$	\$x_t = \pm R \cos(90 - \theta) = \pm R \sin(\theta)\$ \$k = R / \sin(90 - \theta) = R / \cos(\theta)\$ \$y_t = k - R \cos(\theta)\$
Левая	Ближняя	\$h = -R / \sin(\theta)\$ \$k = 0\$	\$x_t = h + R \sin(\theta)\$ \$y_t = \pm R \cos(\theta)\$
Правая	Дальняя	\$h = -R / \sin(\theta)\$ \$k = 0\$	\$x_t = h - R \sin(\theta)\$ \$y_t = \pm R \cos(\theta)\$
Правая	Ближняя	\$h = 0\$ \$k = -R / \sin(90 - \theta) = -R / \cos(\theta)\$	\$x_t = \pm R \cos(90 - \theta) = \pm R \sin(\theta)\$ \$y_t = k + R \cos(\theta)\$

Полный перечень приведен в табл. 4-7. Более сложные примеры с использованием понятий внутреннее-внешнее содержатся в работе [4-3]. Возможно расширение для других конических сечений. В работе [4-4] рассматривается векторный подход к решению подобных проблем.

4-10 ОБЩИЕ УРАВНЕНИЯ КОНИЧЕСКИХ СЕЧЕНИЙ

Все конические сечения из предыдущих разделов являются частными случаями кривых, задаваемых общим уравнением второго порядка

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0, \quad (4-31)$$

где A, B, C, D, E, F — константы¹. Коническое сечение — это любая плоская кривая, удовлетворяющая уравнению (4-31). Уравнение интересно не только само по себе, но и для последующего обсуждения рациональных конических сечений и квадратичных поверхностей. Для простоты и полноты будем пользоваться методами линейной алгебры.

Конические сечения являются центральными — эллипс и гипербола (окружность это частный случай эллипса) или нецентральными — парабола. Кроме того, существует ряд вырожденных форм, которые все центральны. Необходимо определить для заданных значений констант, какое сечение задает уравнение (4-31) — центральное или нецентральное. Также нужно выделить все вырожденные случаи.

Заметим, что уравнение (4-31) можно записать в матричной форме

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} A & D/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$

или

$$[X][S][X]^T = 0. \quad (4-32)$$

Отметим также, что $[S]$ симметрична относительно главной диагонали.

Сначала приведем сечение к стандартному виду. Для центрального сечения (эллипс или гипербола) это значит, что центр находится в начале координат, а оси располагаются вдоль осей координат. Для нецентрального сечения (парабола) ось симметрии параболы совпадает с положительной осью x , вершина находится в начале координат, и парабола раскрывается направо. Сечение приводится к стандартному виду переносом и вращением.

Для центральных сечений линейные члены уравнений (4-31) или (4-32) уничтожаются переносом центра сечения в начало координат. После этого уравнение (4-32) принимает вид

$$[X][Tr][S][Tr]^T[X]^T = 0, \quad (4-33)$$

где матрица переноса $[Tr]$ такова:

$$[Tr] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{bmatrix}.$$

¹ Уравнение (4-31) — это модифицированная версия неявного уравнения из раздела 4-3 с $A = a$, $B = 2b$, $C = c$, $D = 2d$, $E = 2e$, $F = f$.

После конкатенации матриц переноса и коэффициентов уравнение принимает вид

$$[X][S'][X]^T = 0, \quad (4-34)$$

где

$$[S'] = [Tr][S][Tr]^T = \left(\frac{1}{2}\right) \begin{bmatrix} 2A' & B' & D' \\ D' & 2C' & E' \\ D' & E' & 2F' \end{bmatrix} \quad (4-35)$$

и

$$A' = A$$

$$B' = B$$

$$C' = C$$

$$D' = 2Am + Bn + D$$

$$E' = Bm + 2Cn + E$$

$$2F' = mD' + nE' + (Dm + En + 2F).$$

Заметим, что $[S']$ также симметрична.

Следовательно, уравнение (4-31) преобразуется к

$$A'x^2 + B'xy + C'y^2 + D'x + E'y + F' = 0.$$

Коэффициенты переноса m и n для уничтожения линейных членов вычисляются из условия $D' = E' = 0$. Отсюда

$$2Am + Bn + D = 0,$$

$$Bm + 2Cn + E = 0$$

или в матричном виде

$$\begin{bmatrix} m & n \end{bmatrix} \begin{bmatrix} 2A & B \\ B & 2C \end{bmatrix} = \begin{bmatrix} -D \\ -E \end{bmatrix}, \quad (4-36)$$

что можно записать как $[M][L] = [Q]$.

Если $[L]$ инвертируема, существует решение для $[M]$, и сечение центрально, т. е. это эллипс или гипербола. Если $[L]$ не инвертируема, т. е. сингулярна, то решения для $[M]$ не существует, и сечение не центрально (парабола). Детерминант сингулярной матрицы равен нулю:

$$\det[L] = |L| = \begin{vmatrix} 2A & B \\ B & 2C \end{vmatrix} = 0 \quad (4-37)$$

или

$$B^2 - 4AC = 0.$$

Итак, уравнение (4-31) представляет параболу при $B^2 - 4AC = 0$ и центральное сечение при $B^2 - 4AC \neq 0$. Если сечение центрально и $B^2 - 4AC < 0$, уравнение представляет эллипс, а если $B^2 - 4AC > 0$ — гиперболу.

Независимо от инвертируемости $[L]$ оси сечения можно поворотом сделать параллельными осям координат. Вернемся к уравнению (4-32). Используем матрицу плоского поворота $[R]$

$$[X][R][S][R]^T[X]^T = 0, \quad (4-38)$$

где для угла поворота θ она такая:

$$[R] = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Конкатенация матриц дает

$$[X][S''][X]^T = 0, \quad (4-39)$$

где

$$[S''] = [R][S][R]^T = \left(\frac{1}{2}\right) \begin{bmatrix} 2A'' & B'' & D'' \\ B'' & 2C'' & E'' \\ D'' & E'' & 2F'' \end{bmatrix}$$

и

$$\begin{aligned} A'' &= A \cos^2 \theta + B \cos \theta \sin \theta + C \sin^2 \theta, \\ B'' &= 2(C - A) \cos \theta \sin \theta + B(\cos^2 \theta - \sin^2 \theta), \\ C'' &= A \sin^2 \theta - B \cos \theta \sin \theta + C \cos^2 \theta, \\ D'' &= D \cos \theta + E \sin \theta, \\ E'' &= E \cos \theta - D \sin \theta, \\ F'' &= F. \end{aligned}$$

Опять заметим, что $[S'']$ симметрична. Если оси сечения параллельны осям координат, член Bxy в уравнении (4-31) отсутствует. Поэтому нулевой коэффициент дает угол поворота:

$$2(C - A) \cos \theta \sin \theta + B(\cos^2 \theta - \sin^2 \theta) = 0$$

или

$$(C - A) \sin 2\theta + B \cos 2\theta = 0.$$

Решим это уравнение относительно угла поворота θ :

$$\theta = \frac{1}{2} \operatorname{arctg} \left(\frac{B}{A - C} \right). \quad (4-40)$$

Для этого угла $[S'']$ принимает вид

$$[S''] = \left(\frac{1}{2}\right) \begin{bmatrix} 2A'' & 0 & D'' \\ 0 & 2C'' & E'' \\ D'' & E'' & 2F'' \end{bmatrix}_{\theta=\frac{1}{2}\arctg\left(\frac{B}{A-C}\right)}$$

Уравнение (4-37) позволяет узнать, центрально ли сечение. Центральное сечение можно привести к стандартному виду комбинацией переноса и поворота:

$$[X][Tr][R][S][R]^T[Tr]^T[X]^T = 0. \quad (4-41)$$

Конкатенация внутренних матриц дает

$$\begin{aligned} [S'''] &= [Tr][R][S][R]^T[Tr]^T = \\ &= \left(\frac{1}{2}\right) \begin{bmatrix} 2A''' & B''' & D''' \\ B''' & 2C''' & E''' \\ D''' & E''' & 2F''' \end{bmatrix} = \\ &= \begin{bmatrix} A''' & 0 \\ 0 & C''' \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \\ 0 & 0 \end{bmatrix}, \end{aligned} \quad (4-42)$$

где

$$A''' = A \cos^2 \theta + B \cos \theta \sin \theta + C \sin^2 \theta, \quad (4-43a)$$

$$B''' = 2(C - A) \cos \theta \sin \theta + B(\cos^2 \theta - \sin^2 \theta), \quad (4-43b)$$

$$C''' = A \sin^2 \theta - B \cos \theta \sin \theta + C \cos^2 \theta, \quad (4-43c)$$

$$D''' = \xi \cos \theta + \eta \sin \theta, \quad (4-43d)$$

$$E''' = \eta \cos \theta - \xi \sin \theta, \quad (4-43e)$$

$$F''' = (m\xi + \eta n + \zeta)/2 \quad (4-43f)$$

и

$$\xi = 2Am + Bn + D, \quad (4-43g)$$

$$\eta = Bm + 2Cn + E, \quad (4-43h)$$

$$\zeta = Dm + En + 2F. \quad (4-43i)$$

Заметим, что это диагональная матрица, т. е. все недиагональные элементы равны нулю. Поворот устраниет член ($B''' = 0$), а перенос — линейные члены ($D''' = E''' = 0$).

Угол вращения задается уравнением (4-40). Как и раньше, коэффициенты переноса можно получить, приравняв D''' и E''' к нулю, т. е. $\xi = \eta = 0$. Имеем

$$2Am + Bn + D = 0,$$

$$Bm + 2Cn + E = 0.$$

Отсюда получаем решение

$$m = \frac{2CD - BE}{B^2 - 4AC}, \quad (4-44a)$$

$$n = \frac{2AE - BD}{B^2 - 4AC}. \quad (4-44b)$$

Вспомним, что для центрального конического сечения $B^2 - 4AC \neq 0$. Запишем уравнение (4-41), используя (4-42):

$$\alpha x^2 + \beta y^2 = \kappa, \quad (4-45)$$

что является стандартным видом конического сечения. Остается систематически исследовать результат для различных значений α и β .

Если κ больше нуля, то α и β положительны, и сечение — эллипс. Если они имеют разные знаки и ни один из них не равен нулю, то сечение — гипербола.

Если α и β отрицательны, решения не существует.

Оба α и β одновременно не могут быть равны нулю, так как в этом случае уравнение (4-45) не содержит членов второго порядка. Однако один из коэффициентов α или β может быть нулевым. Пусть $\beta = 0$ (если $\alpha = 0$, замена x на y дает $\beta = 0$), тогда уравнение (4-45) принимает вид

$$\alpha x^2 = -\kappa.$$

Решением являются две параллельные линии $x = \pm\sqrt{-\kappa/\alpha}$ при $-\kappa/\alpha > 0$. Если $-\kappa/\alpha < 0$, то решения нет.

Если $\kappa = 0$, есть две возможности: α и β имеют одинаковые или разные знаки. В обоих случаях решение вырожденное. Если знаки α и β одинаковы, уравнению (4-45) удовлетворяет только начало координат $x = y = 0$. Можно считать, что это предельный случай эллипса.

Если знаки α и β различны, уравнение (4-45) принимает вид

$$y^2 = -\frac{\alpha}{\beta}x^2$$

или

$$y = \pm\sqrt{-\frac{\alpha}{\beta}}x,$$

т. е. это пара прямых, пересекающихся в начале координат, — предельный случай гиперболы.

Наконец, если $\beta = 0$ (если $\alpha = 0$, замена x на y дает $\beta = 0$), то решение — ось y для всех значений.

Для нецентрального сечения, параболы, оба линейных члена устраниТЬ нельзя, однако можно убрать один линейный и один квадратичный член для x или y . Применим оператор переноса к уравнению (4-39) — квадратному уравнению после поворота:

$$[S^+] = \frac{1}{2} \begin{bmatrix} 2A^+ & 0 & D^+ \\ 0 & 2C^+ & E^+ \\ D^+ & E^+ & 2F^+ \end{bmatrix}, \quad (4-46)$$

где

$$A^+ = A'', \quad (4-47a)$$

$$C^+ = C'', \quad (4-47b)$$

$$D^+ = 2A''m + D'', \quad (4-47c)$$

$$E^+ = 2C''n + E'', \quad (4-47d)$$

$$F^+ = (m/2)D^+ + (n/2)E^+ + (mD'' + nE'' + 2F''). \quad (4-47e)$$

Угол поворота задан уравнением (4-40). Здесь либо $A^+ = A''$, либо $C^+ = C''$ обратится в нуль. Один из линейных членов для x или y устраниется, если D^+ или E^+ приравнять к нулю. Пусть $D^+ = 0$, тогда

$$m = \frac{-D''}{2A''}. \quad (4-48a)$$

Если $E^+ = 0$, то

$$n = \frac{-E''}{2C''}. \quad (4-48b)$$

Заметим, что при $A'' = 0$ значение m не определено, следовательно, устраняются только линейные члены относительно y . Если $C'' = 0$, не определено n и устраняются только члены с x .

Предположим, что уничтожены линейный y -член ($E^+ = 0$) и квадратичный x -член ($A^+ = 0$) (если $\beta = 0$, замена x и y дает $\gamma = 0$). Тогда $[S^+]$ принимает вид

$$[S^+] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \beta & 0 \\ \gamma & 0 & -\kappa \end{bmatrix}.$$

Запишем уравнение конического сечения

$$\beta y^2 + 2\gamma x = \kappa. \quad (4-49)$$

Чтобы привести параболу к стандартной ориентации с вершиной κ в центре координат, перенесем ее по оси x на

$$y^2 = -\frac{2\gamma}{\beta}x' = ax'.$$

Все вырожденные формы сечений центральны, т. е. парабола — это единственное нецентральное сечение. Результаты собраны в табл. 4-8.

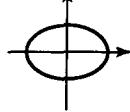
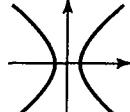
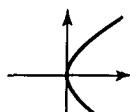
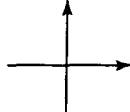
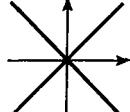
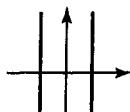
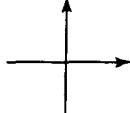
Пример 4-7 Сегмент гиперболы

Найти тип конического сечения, заданного формулой

$$2x^2 - 72xy + 23y^2 + 140x - 20y + 50 = 0,$$

изображенного непрерывной линией на рис. 4-16. Нарисовать его сегмент для $3 < x < 5$ для y_{\min} . Сегмент должен быть нарисован с помощью параметрического представления из предыдущих разделов. Для того чтобы определить

Таблица 4-8 Конические сечения*

Название	Уравнение	Условия	Тип	Чертеж
Эллипс	$\alpha x^2 + \beta y^2 = \kappa$	$\kappa, \alpha, \beta > 0$	Центральный	
Гипербола	$\alpha x^2 + \beta y^2 = \kappa$	$\beta < 0 < \kappa, \alpha$	Центральный	
Парабола	$\alpha y^2 + \beta x = 0$ $\beta x^2 + \alpha y = 0$		Нецентральный	
Пустое множество	$\alpha x^2 + \beta y^2 = \kappa$	$\alpha, \beta < 0 < \kappa$	(Центральный)	(Чертеж отсутствует)
Точка	$\alpha x^2 + \beta y^2 = 0$	$\alpha, \beta > 0$	Центральный	
Пара прямых	$\alpha x^2 + \beta y^2 = 0$	$\beta < 0 < \alpha$	Центральный	
Параллельные прямые	$\alpha x^2 = \kappa$	$\alpha, \kappa > 0$	Центральный	
Пустое множество	$\alpha x^2 = \kappa$	$\alpha < 0 < \kappa$	(Центральный)	(Чертеж отсутствует)
«Повторяющаяся» прямая	$\alpha x^2 = 0$		Центральный	

* Эллипс, гипербола и парабола — это невырожденные сечения, остальные — вырожденные.

значения параметрического представления, используются методы, рассмотренные в данном разделе.

Сначала узнаем тип сечения

$$B^2 - 4AC = (-72)^2 - (4)(2)(23) = 5000 > 0,$$

т. е. это гипербола.

Приведем гиперболу к стандартному виду с помощью уравнений (4-40)–(4-44). Угол поворота

$$\theta = \frac{1}{2} \operatorname{arctg} \left(\frac{B}{A - C} \right) = \frac{1}{2} \operatorname{arctg} \left(\frac{-72}{2 - 23} \right) = \frac{1}{2} \operatorname{arctg} \left(\frac{24}{7} \right) = 36.87^\circ.$$

Подставим значения $\cos \theta = 4/5$, $\sin \theta = 3/5$ и получим

$$\begin{aligned} A''' &= A \cos^2 \theta + B \cos \theta \sin \theta + C \sin^2 \theta = \\ &= (2)(4/5)^2 + (-72)(4/5)(3/5) + (23)(3/5)^2 = \\ &= -625/25 = -25, \end{aligned}$$

$$\begin{aligned} C''' &= A \sin^2 \theta - B \cos \theta \sin \theta + C \cos^2 \theta = \\ &= (2)(3/5)^2 - (-72)(4/5)(3/5) + (23)(4/5)^2 = \\ &= 1250/25 = 50. \end{aligned}$$

Коэффициенты переноса m и n таковы:

$$m = \frac{2CD - BE}{B^2 - 4AC} = \frac{(2)(23)(140) - (-72)(-20)}{5000} = \frac{5000}{5000} = 1,$$

$$n = \frac{2AE - BD}{B^2 - 4AC} = \frac{(2)(2)(-20) - (-72)(140)}{5000} = \frac{10000}{5000} = 2.$$

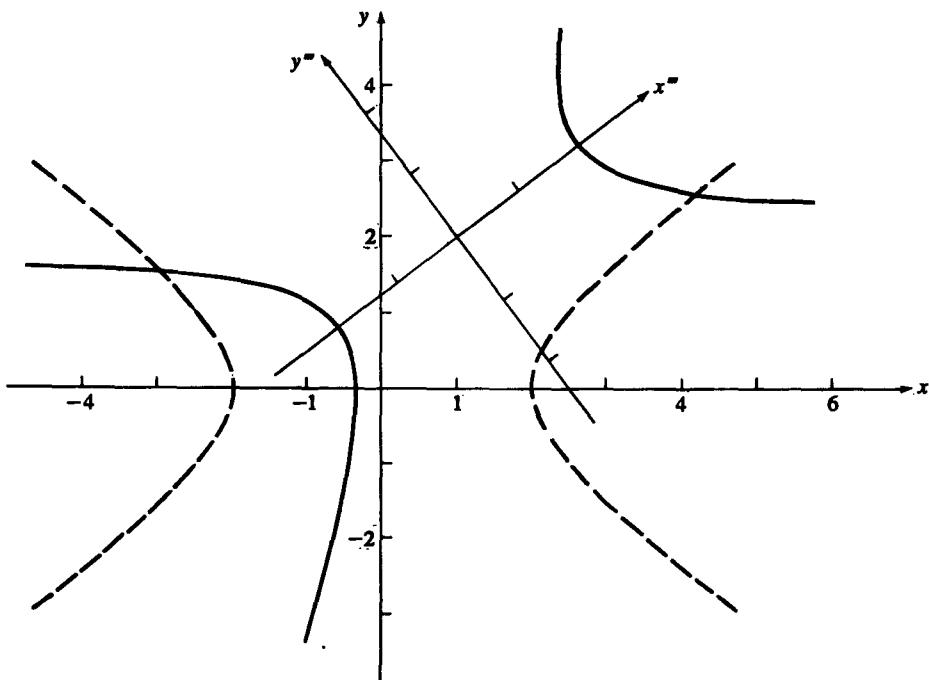


Рис. 4-16 Гипербола $2x^2 - 72xy + 23y^2 + 140x - 20y + 50 = 0$. Сплошная линия — заданная ориентация. Пунктир — стандартная ориентация.

Так как $\xi = \eta = 0$, константа равна

$$\begin{aligned} F''' &= \frac{\zeta}{2} = \frac{(Dm + En + 2F)}{2} \\ &= \frac{(140)(1) + (-20)(2) + (2)(50)}{2} = \frac{200}{2} = 100. \end{aligned}$$

Уравнение гиперболы

$$-25x^2 + 50y^2 + 100 = 0.$$

В стандартной форме $\alpha x^2 + \beta y^2 + \kappa = 0$ имеем

$$-x^2 + 2y^2 + 4 = 0 \Rightarrow x^2 - 2y^2 = 4.$$

В форме

$$\left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = 1$$

имеем

$$\frac{x^2}{4} - \frac{y^2}{2} = 1,$$

что дает $a = 2$, $b = \sqrt{2}$.

Параметрическое представление параболы (4-16):

$$x = a \operatorname{ch}(t), \quad y = b \operatorname{sh}(t),$$

где t — параметр. Для параметрического изображения гиперболы необходимо найти величину t для $x = 3,5$. Заметим, что преобразования к стандартному виду переносят оси координат, а не само сечение. Поэтому соответствующие преобразованные значения x получаются обратными преобразованиями. В результате

$$\begin{aligned} [TR] &= [T][R] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -m & -n & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin(-\theta) & 0 \\ -\sin(-\theta) & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ -(m \cos \theta + n \sin \theta) & m \sin \theta - n \cos \theta & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 4/5 & -3/5 & 0 \\ 3/5 & 4/5 & 0 \\ -2 & -1 & 1 \end{bmatrix}. \end{aligned}$$

Преобразованные координаты x

$$\begin{bmatrix} 3 & 2.871 & 1 \\ 5 & 2.476 & 1 \end{bmatrix} \begin{bmatrix} 4/5 & -3/5 & 0 \\ 3/5 & 4/5 & 0 \\ -2 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 2.123 & -0.503 & 1 \\ 3.486 & -2.019 & 1 \end{bmatrix},$$

где также включена y координата. Тогда параметрические значения

$$t_{\min} = t|_{x=3} = \operatorname{Arch}\left(\frac{x}{2}\right) = \operatorname{Arch}\left(\frac{2.123}{2}\right) = 0.3486,$$

$$t_{\max} = t|_{x=5} = \operatorname{Arch}\left(\frac{x}{2}\right) = \operatorname{Arch}\left(\frac{3.486}{2}\right) = 1.1539.$$

Пользуясь этими значениями для t_{\min} и t_{\max} , получаем результат на рис. 4-16 и в табл. 4-9.

Таблица 4-9 Сегмент гиперболы в стандартной ориентации

t	x	y
0.349	2.123	-0.503
0.438	2.195	-0.640
0.528	2.285	-0.781
0.617	2.393	-0.929
0.706	2.520	-1.084
0.796	2.668	-1.248
0.885	2.837	-1.422
0.975	3.028	-1.608
1.064	3.244	-1.806
1.153	3.486	-2.019

Затем эти результаты обратным преобразованием переводятся в исходное положение:

$$[TR]^{-1} = \begin{bmatrix} 4/5 & 3/5 & 0 \\ -3/5 & 4/5 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

Результат приведен в табл. 4-10 и на рис. 4-16.

Таблица 4-10 Сегмент гиперболы в исходной ориентации

t	x	y
0.349	3.000	2.871
0.438	3.140	2.805
0.528	3.297	2.746
0.617	3.472	2.693
0.706	3.667	2.645
0.796	3.883	2.602
0.885	4.123	2.564
0.975	4.387	2.531
1.064	4.679	2.502
1.153	5.000	2.476

4-11 ЛИТЕРАТУРА

- 4-1 Lancaster P. and Salkauskas K., *Curve and Surface Fitting, An Introduction*, Academic Press, 1987.
- 4-2 Smith L. B., «Drawing Ellipses, Hyperbolas, or Parabolas With a Fixed Number of Points and Maximum Inscribed Area», *Comp. J.*, Vol. 14, pp. 81–86, 1969.
- 4-3 Rogers D. F., «Interactive Graphics and Numerical Control», *Comp. Aid. Des.*, Vol. 12, pp. 253–261, 1980.
- 4-4 Middleditch A. E. and Stacey T. W., «Robust Computation of Tangent Lines and Circles», *Comp. Aid. Des.*, Vol. 19, pp. 503–507, 1987.

Глава 5

ПРОСТРАНСТВЕННЫЕ КРИВЫЕ

5-1 ВВЕДЕНИЕ

Трехмерные, или пространственные, кривые широко используются в проектировании и разработке самой различной продукции: автомобилей, кораблей, самолетов, обуви, бутылок, зданий и т. д. Также они имеют большое значение для описания и интерпретации физических явлений в геологии, физике и медицине.

До начала применения математических и компьютерных моделей в процессе производства, дизайна и изготовления использовалась начертательная геометрия. Многие ее методы были перенесены в машинную графику.

Поверхности часто изображаются как сеть кривых, лежащих в ортогональных секущих плоскостях, с трехмерными контурами деталей. Пример приведен на рис. 5-1. Здесь сечения были получены оцифровкой физической модели или

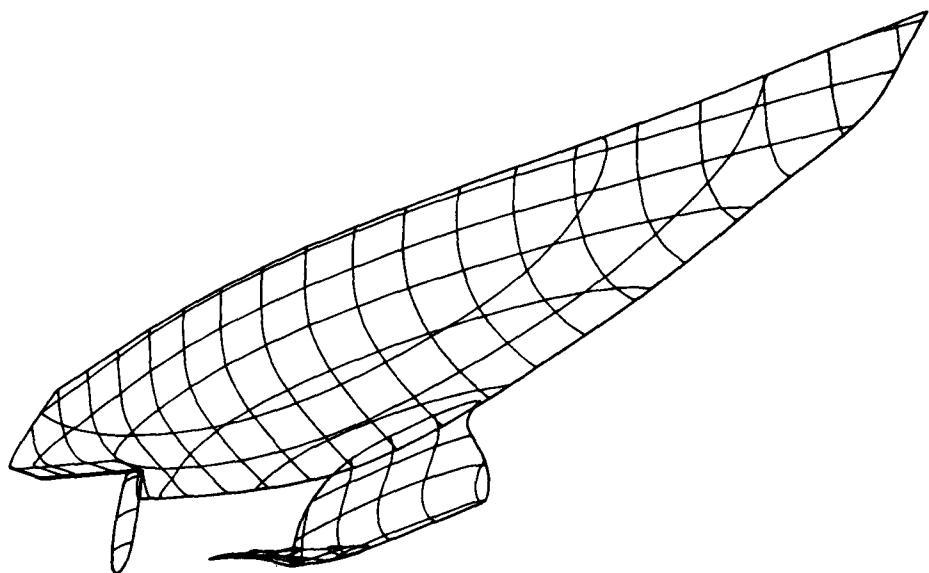


Рис. 5-1 Поверхность в виде сети ортогональных плоских кривых (Дж. Хейзен)

чертежа и математическим подбором кривой, проходящей через все заданные точки. В данной главе рассматриваются два таких метода: кубические сплайны и параболическая интерполяция. Существуют и другие реализации этого подхода, рассмотренные, например, в работе [5-1].

Другой подход состоит в том, что математическое описание кривых генерируется без изначального знания формы кривой. Его примеры — это кривые Безье и их обобщение до B-сплайнов. Эти методы отличаются тем, что кривая может не проходить ни через одну заданную точку. Контрольные точки определяют только направление изгиба.

Как будет показано далее, оба подхода можно применить при любом способе задания кривой.

5-2 ПРЕДСТАВЛЕНИЕ ПРОСТРАНСТВЕННЫХ КРИВЫХ

Трехмерные кривые можно представить параметрически или непараметрически. Явное непараметрическое представление имеет вид

$$\begin{aligned}x &= x, \\y &= f(x), \\z &= g(x).\end{aligned}$$

Неявное непараметрическое представление кривой как пересечения двух поверхностей задается уравнениями:

$$\begin{aligned}f(x, y, z) &= 0, \\g(x, y, z) &= 0.\end{aligned}$$

Пример 5-1 Пространственная кривая

Найти линию пересечения двух поверхностей второго порядка

$$\begin{aligned}f(x, y, z) &= y - z^2 = 0, \\g(x, y, z) &= zx - y^2 = 0.\end{aligned}$$

При условиях что

$$\det \begin{vmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{vmatrix} \neq 0,$$

$z \neq 0$, x и y можно выразить относительно z и получить явный вид линии пересечения

$$\begin{aligned}y &= z^2, \\x &= \frac{y^2}{z} = z^3.\end{aligned}$$

Заметим, что при пересечении двух поверхностей второго порядка получается кривая третьего порядка. Поверхности и кривая пересечения показаны на рис. 5-2.

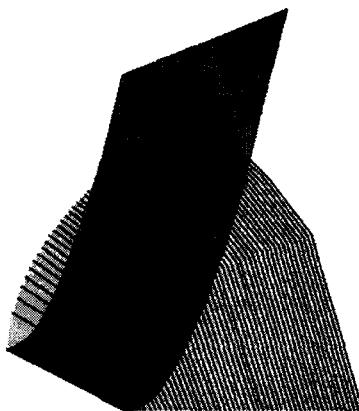


Рис. 5-2 Пересечение кривых второго порядка.

Общий параметрический вид пространственной кривой можно записать в виде:

$$\begin{aligned}x &= x(t), \\y &= y(t), \\z &= z(t),\end{aligned}$$

где параметр t изменяется в определенных пределах $t_1 \leq t \leq t_2$. В приведенном выше явном непараметрическом представлении x можно рассматривать как параметр, $x = t$. Тогда эта же кривая имеет параметрическую форму

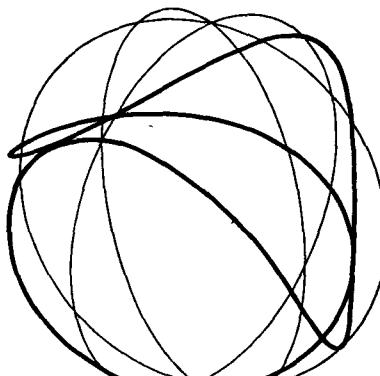
$$\begin{aligned}x &= t, \\y &= f(t), \\z &= g(t).\end{aligned}$$

Далее, пусть $z = t$ в неявном непараметрическом представлении из примера 5-1, тогда

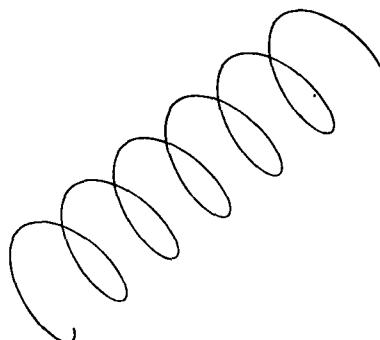
$$\begin{aligned}x &= t^3, \\y &= t^2, \\z &= t.\end{aligned}$$

Некоторые полезные параметрические трехмерные кривые имеют известное аналитическое решение. Например, кривая шва на теннисном или бейсбольном мяче имеет вид:

$$\begin{aligned}x &= \lambda[a \cos(\theta + \frac{\pi}{4}) - b \cos 3(\theta + \frac{\pi}{4})], \\y &= \mu[a \sin(\theta + \frac{\pi}{4}) - b \sin 3(\theta + \frac{\pi}{4})], \\z &= c \sin(2\theta),\end{aligned}$$



(a)



(b)

Рис. 5-3 Примеры параметрических пространственных кривых. (а) Шов бейсбольного мяча; (б) круговая спираль.

где

$$\lambda = 1 + d \sin(2\theta) = 1 + d \left(\frac{z}{c}\right),$$

$$\mu = 1 - d \sin(2\theta) = 1 - d \left(\frac{z}{c}\right),$$

параметр $\theta = 2\pi t$ и $0 \leq t \leq 1.0$. Если $d = 0$ и $c^2 = 4ab$, то кривая лежит на сфере радиуса $a + b$. На рис. 5-3а приведен пример для $a = 1$, $b = 1$, $c = 2$, $d = 0$, где кривая лежит на сфере радиуса 2.

Другой пример параметрической пространственной кривой — круговая спираль:

$$x = r \cos t,$$

$$y = r \sin t,$$

$$z = bt$$

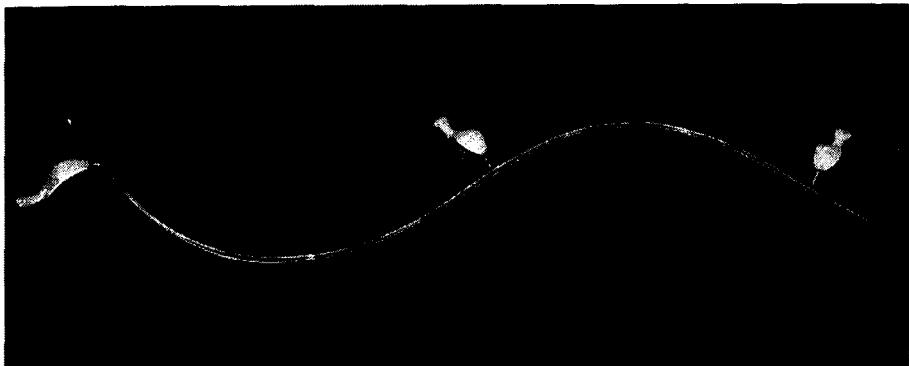


Рис. 5-4 Физические отвесы и сплайн.

для r , $b \neq 0$, $-\infty < t < \infty$. Эта кривая лежит на поверхности цилиндра радиуса $|r|$. Уравнение $z = bt$ отвечает за распространение спирали по оси z . После каждого изменения параметра t на 2π переменные x и y возвращаются к своим первоначальным значениям, а z увеличивается или уменьшается на $2\pi|b|$ в зависимости от знака b . Эта величина называется шагом спирали. Пример изображен на рис. 5-3б.

5-3 КУБИЧЕСКИЕ СПЛАЙНЫ

В промышленном производстве, например, судо-, автомобилье- и авиастроении, окончательная форма в реальном или близком к нему масштабе определяется в процессе доводки¹.

Автоматизация этого процесса представляла значительный интерес для машинной графики. Форма математического сплайна повторяет контур физического сплайна (рис. 5-4), т. е. гибкой деревянной или пластмассовой линейки, проходящей через определенные точки. Для изменения формы сплайна используются свинцовые грузики. Меняя их количество и расположение, получившуюся кривую стараются сделать более гладкой, красивой и «приятной для глаза»².

Если рассматривать физический сплайн как тонкую гибкую рейку, его форма (отклонение y) определяется уравнением Эйлера (5-2) для момента изгиба $M(x)$ вдоль рейки:

$$M(x) = \frac{EI}{R(x)},$$

где E — модуль Юнга, зависящий от свойств материала рейки, I — момент инерции, определяемый формой кривой, $R(x)$ — радиус кривизны.

¹ В судостроении, например, необходимо определить форму судна более точно, чем это возможно на небольшом чертеже. Поэтому общепринято изображать контур судна в натуральную величину на плазе.

² Не существует общепринятого термина для оценки внешнего вида кривой. Мы не пытаемся дать математического определения, потому что любая такая попытка вызовет волну протеста по крайней мере у половины заинтересованных специалистов.

Для малых отклонений ($y' \ll 1$) радиус приближенно равен

$$\frac{1}{R(x)} = \frac{y''}{(1 + y'^2)^{3/2}} \approx y'',$$

где штрих обозначает производную по x — расстоянию вдоль рейки, а y — отклонение рейки. Уравнение Эйлера принимает вид

$$y'' = \frac{M(x)}{EI}.$$

Пусть грузики действуют как простые подпорки, тогда момент изгиба между ними изменяется линейно. Подставляя $M(x) = Ax + B$ в уравнение Эйлера, получаем

$$y'' = \frac{Ax + B}{EI}$$

и после двойного интегрирования

$$y = A_1 x^3 + B_1 x^2 + C_1 x + D_1.$$

Таким образом, форма сплайна задается кубическим полиномом.

В общем случае математический сплайн это кусочный полином степени K с непрерывной производной степени $K - 1$ в точках соединения сегментов. Так, например, кубический сплайн имеет в точках соединения непрерывность второго порядка. Кусочные сплайны из многочленов невысокого порядка очень удобны для интерполяции кривых, так как они не требуют больших вычислительных затрат и не вызывают численных отклонений, свойственных многочленам высокого порядка. По аналогии с физическими сплайнами обычно используется серия кубических сегментов, причем каждый сегмент проходит через две точки. Кубический сплайн удобен еще и тем, что это кривая наименьшего порядка, допускающая точки перегиба и изгиб в пространстве.

Уравнение одного параметрического сегмента сплайна таково:

$$P(t) = \sum_{i=1}^4 B_i t^{i-1}, \quad t_1 \leq t \leq t_2, \tag{5-1}$$

где t_1 и t_2 — значения параметров в начале и конце сегмента. $P(t)$ — вектор к любой точке сегмента. $P(t) = [x(t) \ y(t) \ z(t)]$ — это векторно-значная функция, где три составляющие $P(t)$ — декартовы координаты вектора¹.

¹ $P(t) = [r(t) \ \theta(t) \ z(t)]$, где $r(t)$, $\theta(t)$, $z(t)$ — компоненты цилиндрической системы координат, также совершенно справедливо, как, впрочем, и представление в любой другой системе координат. (См. задачу 5-8.)



Рис. 5-5 Один сегмент кубического сплайна.

Каждая составляющая имеет вид, похожий на $P(t)$, т. е.

$$x(t) = \sum_{i=1}^4 B_{i_x} t^{i-1}, \quad t_1 \leq t \leq t_2,$$

$$y(t) = \sum_{i=1}^4 B_{i_y} t^{i-1}, \quad t_1 \leq t \leq t_2,$$

$$z(t) = \sum_{i=1}^4 B_{i_z} t^{i-1}, \quad t_1 \leq t \leq t_2.$$

Постоянные коэффициенты B_i вычисляются исходя из четырех граничных условий для сегмента сплайна. Запишем уравнение (5-1) в виде

$$P(t) = B_1 + B_2 t + B_3 t^2 + B_4 t^3, \quad t_1 \leq t \leq t_2. \quad (5-2)$$

Пусть P_1 и P_2 — векторы концов сегмента (см. рис. 5-5). Пусть также P'_1 и P'_2 , производные по t , будут касательными векторами в концах сегмента. Дифференцируя уравнение (5-1), получим

$$P'(t) = [x'(t) \quad y'(t) \quad z'(t)] = \sum_{i=1}^4 B_i(i-1)t^{i-2}, \quad t_1 \leq t \leq t_2. \quad (5-3)$$

Запишем результат

$$P'(t) = B_2 + 2B_3 t + 3B_4 t^2, \quad t_1 \leq t \leq t_2. \quad (5-4)$$

Предположим, без потери общности, что $t_1 = 0$, и применим граничные условия

$$P(0) = P_1, \quad (5-5a)$$

$$P(t_2) = P_2, \quad (5-5b)$$

$$P'(0) = P'_1, \quad (5-5c)$$

$$P'(t_2) = P'_2. \quad (5-5d)$$

Получим четыре уравнения для неизвестных B_i :

$$P(0) = B_1 = P_1, \quad (5-6a)$$

$$P'(0) = \sum_{i=1}^4 (i-1)t^{i-2}B_i \Big|_{t=0} = B_2 = P'_1, \quad (5-6b)$$

$$P(t_2) = \sum_{i=1}^4 B_i t^{i-1} \Big|_{t=t_2} = B_1 + B_2 t_2 + B_3 t_2^2 + B_4 t_2^3, \quad (5-6c)$$

$$P'(t_2) = \sum_{i=1}^4 (i-1)t^{i-2}B_i \Big|_{t=t_2} = B_2 + 2B_3 t_2 + 3B_4 t_2^2. \quad (5-6d)$$

Решения для B_3 и B_4 имеют вид:

$$B_3 = \frac{3(P_2 - P_1)}{t_2^2} - \frac{2P'_1}{t_2} - \frac{P'_2}{t_2} \quad (5-7a)$$

и

$$B_4 = \frac{2(P_1 - P_2)}{t_2^3} + \frac{P'_1}{t_2^2} + \frac{P'_2}{t_2^2}. \quad (5-7b)$$

Величины B_1, B_2, B_3 и B_4 задают сегмент кубического сплайна. Очевидно, что форма сегмента зависит от положения и касательных векторов в концах сегмента. Далее, заметим, что в результатах присутствует значение параметра $t = t_2$ в конце сегмента. Так как каждая конечная точка и вектор касания имеют три компоненты, параметрическое уравнение кубической пространственной кривой зависит от двенадцати векторных компонент и значения параметра t_2 в конце сегмента.

Подставив уравнения (5-6) и (5-7) в (5-1), получим уравнение для одного сегмента кубического сплайна:

$$\begin{aligned} P(t) &= P_1 + P'_1 t + \left[\frac{3(P_2 - P_1)}{t_2^2} - \frac{2P'_1}{t_2} - \frac{P'_2}{t_2} \right] t^2 \\ &\quad + \left[\frac{2(P_1 - P_2)}{t_2^3} + \frac{P'_1}{t_2^2} + \frac{P'_2}{t_2^2} \right] t^3. \end{aligned} \quad (5-8)$$

Это уравнение для одного сегмента. Чтобы получить кривую целиком, нужно соединить множество сегментов. На рис. 5-6 показаны два соседних сегмента. Если известны векторы P_1, P_2, P_3 , касательные векторы P'_1, P'_2, P'_3 и значения параметров t_2, t_3 , то форма каждого сегмента определяется из уравнения (5-8). Однако маловероятно, что известен касательный вектор P'_2 в точке соединения. К счастью, его можно вывести из условия непрерывности.

Вспомним, что кусочный сплайн степени K имеет непрерывность степени $K-1$ в точках соединения; непрерывность кубического сплайна равна двум. Для этого должна быть непрерывна вторая производная $P''_2(t)$ или кривизна линии. Дважды продифференцировав уравнение (5-1), получим

$$P''(t) = \sum_{i=1}^4 (i-1)(i-2)B_i t^{i-3}, \quad t_1 \leq t \leq t_2. \quad (5-9)$$

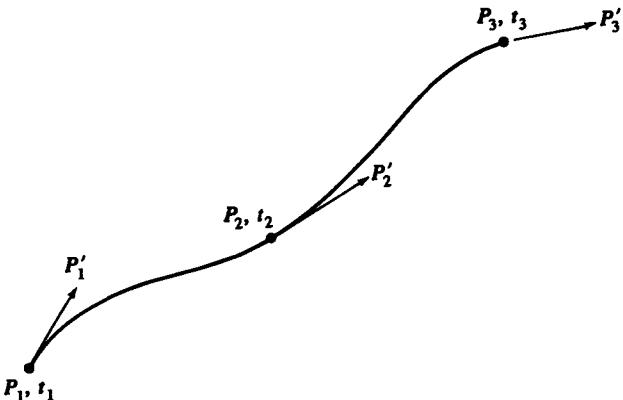


Рис. 5-6 Два кусочно кубических сегмента сплайна.

Для первого куска сплайна параметр изменяется в пределах $0 \leq t \leq t_2$. Подставим $t = t_2$ в уравнение (5-9):

$$P'' = 6B_4t_2 + 2B_3.$$

Для второго участка сплайна параметр изменяется в диапазоне $0 \leq t \leq t_3$. Подставим в уравнение (5-9) значение $t = 0$ в начале второго участка

$$P'' = 2B_3.$$

Приравнивая полученные результаты и пользуясь уравнениями (5-6a, b) и (5-7a), получим

$$\begin{aligned} 6t_2 \left[\frac{2(P_1 - P_2)}{t_2^3} + \frac{P'_1}{t_2^2} + \frac{P'_2}{t_2^2} \right] + 2 \left[\frac{3(P_2 - P_1)}{t_2^2} - \frac{2P'_1}{t_2} - \frac{P'_2}{t_2} \right] = \\ = 2 \left[\frac{3(P_3 - P_2)}{t_3^2} - \frac{2P'_2}{t_3} - \frac{P'_3}{t_3} \right]. \end{aligned}$$

Левая часть этого уравнения представляет кривизну в конце первого сегмента, а правая — в начале второго. Домножим на t_2t_3 и сгруппируем члены:

$$t_3P'_1 + 2(t_3 + t_2)P'_2 + t_2P'_3 = \frac{3}{t_2t_3}[t_2^2(P_3 - P_2) + t_3^2(P_2 - P_1)]. \quad (5-10)$$

Отсюда определяется P'_2 , неизвестный касательный вектор в точке соединения. Отметим, что в окончательном уравнении опять присутствуют значения параметра t в концах сегментов t_2 и t_3 .

Полученную формулу можно обобщить для n точек, и для $n - 1$ сегментов кубического сплайна получить непрерывность второго порядка в точках соедине-

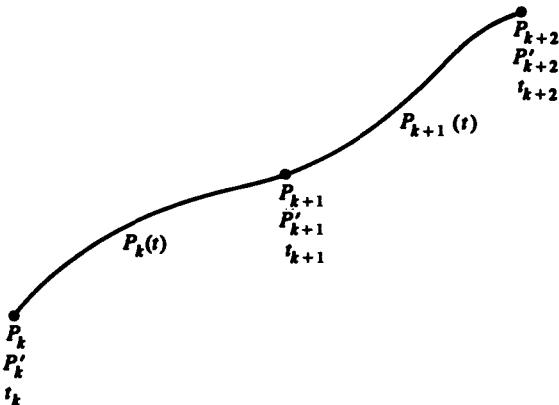


Рис. 5-7 Обозначения множества кусочно кубических сегментов сплайна.

ния. Обобщенное уравнение для двух любых соседних сегментов сплайна $P_k(t)$ и $P_{k+1}(t)$ ¹ в обозначениях рис. 5-7 имеет вид:

$$\begin{aligned} P_k(t) = P_k + P'_k t + \left[\frac{3(P_{k+1} - P_k)}{t_{k+1}^2} - \frac{2P'_k}{t_{k+1}} - \frac{P'_{k+1}}{t_{k+1}} \right] t^2 + \\ + \left[\frac{2(P_k - P_{k+1})}{t_{k+1}^3} + \frac{P'_k}{t_{k+1}^2} + \frac{P'_{k+1}}{t_{k+1}^2} \right] t^3 \end{aligned} \quad (5-11)$$

для первого сегмента и

$$\begin{aligned} P_{k+1}(t) = P_{k+1} + P'_{k+1} t + \left[\frac{3(P_{k+2} - P_{k+1})}{t_{k+2}^2} - \frac{2P'_{k+1}}{t_{k+2}} - \frac{P'_{k+2}}{t_{k+2}} \right] t^2 + \\ + \left[\frac{2(P_{k+1} - P_{k+2})}{t_{k+2}^3} + \frac{P'_{k+1}}{t_{k+2}^2} + \frac{P'_{k+2}}{t_{k+2}^2} \right] t^3 \end{aligned} \quad (5-12)$$

для второго, так как для каждого сегмента параметр начинает изменяться с нуля, для первого $0 \leq t \leq t_{k+1}$ и для второго $-0 \leq t \leq t_{k+2}$.

Приравнивание вторых производных в точках стыковки для любых соседних сегментов, $P''_k(t_k) = P''_{k+1}(0)$, дает общий результат, эквивалентный уравнению (5-10),

$$\begin{aligned} t_{k+2}P'_k + 2(t_{k+1} + t_{k+2})P'_{k+1} + t_{k+1}P'_{k+2} = \\ = \frac{3}{t_{k+1}t_{k+2}} \left[t_{k+1}^2(P_{k+2} - P_{k+1}) + t_{k+2}^2(P_{k+1} - P_k) \right], \quad 1 \leq k \leq n-2, \end{aligned} \quad (5-13)$$

откуда определяется касательный вектор в точках соединения любых двух сегментов P_k и P_{k+1} .

¹ Чтобы не вводить лишних определений, условимся обозначать участок параметрической кривой между точками P_k и P_{k-1} как $P_k(t)$.

Рекурсивное использование уравнения (5-13) для всех сегментов сплайна порождает $n - 2$ уравнений касательных векторов P'_k , $2 \leq k \leq n - 1$. В матричной форме:

$$\begin{bmatrix} t_3 & 2(t_2 + t_3) & t_2 & 0 & \cdot & \cdot & \cdot & \cdot \\ 0 & t_4 & 2(t_3 + t_4) & t_3 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & t_5 & 2(t_4 + t_5) & t_4 & 0 & \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 & t_n & 2(t_n + t_{n-1}) & t_{n-1} \end{bmatrix} = \begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ \cdot \\ \cdot \\ \cdot \\ P'_n \end{bmatrix}$$

$$= \begin{bmatrix} \frac{3}{t_2 t_3} \{t_2^2 (P_3 - P_2) + t_3^2 (P_2 - P_1)\} \\ \frac{3}{t_3 t_4} \{t_3^2 (P_4 - P_3) + t_4^2 (P_3 - P_2)\} \\ \cdot \\ \cdot \\ \cdot \\ \frac{3}{t_{n-1} t_n} \{t_{n-1}^2 (P_n - P_{n-1}) + t_n^2 (P_{n-1} - P_{n-2})\} \end{bmatrix} \quad (5-14)$$

или

$$[M^*][P'] = [R].$$

Матрица $[M^*]$ неквадратная, так как имеется только $n - 2$ уравнений для n векторов, и ее нельзя обратить и получить решение для $[P']$. Если предположить, что известны касательные векторы на концах кривой P'_1 и P'_n , проблема разрешается. Теперь матрица имеет вид

$$\begin{bmatrix} 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ t_3 & 2(t_2 + t_3) & t_2 & 0 & \cdot & \cdot & \cdot & \cdot \\ 0 & t_4 & 2(t_3 + t_4) & t_3 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & t_5 & 2(t_4 + t_5) & t_4 & 0 & \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 0 & t_n & 2(t_n + t_{n-1}) & t_{n-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 & 1 \end{bmatrix} = \begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ \cdot \\ \cdot \\ \cdot \\ P'_n \end{bmatrix}$$

$$\begin{aligned}
 & \left[\begin{array}{l} P'_1 \\ \frac{3}{t_2 t_3} \{t_2^2(P_3 - P_2) + t_3^2(P_2 - P_1)\} \\ \frac{3}{t_3 t_4} \{t_3^2(P_4 - P_3) + t_4^2(P_3 - P_2)\} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \frac{3}{t_{n-1} t_n} \{t_{n-1}^2(P_n - P_{n-1}) + t_n^2(P_{n-1} - P_{n-2})\} \\ P'_n \end{array} \right] \\
 = & \quad (5-15)
 \end{aligned}$$

или

$$[M][P'] = [R],$$

где матрица $[M]$ квадратная и обратимая. Заметим также, что $[M]$ трехдиагональная¹, что снижает вычислительные затраты на ее обращение. Далее, матрица $[M]$ диагонально доминантная². Отсюда следует, что у нее существует единственное решение:

$$[P'] = [M]^{-1}[R]. \quad (5-16)$$

Если нам известны P'_k , то легко определить коэффициенты B_i для каждого сегмента сплайна. Обобщая уравнения (5-6)–(5-11), получим

$$\begin{aligned}
 B_{1k} &= P_k, \\
 B_{2k} &= P'_k, \\
 B_{3k} &= \frac{3(P_{k+1} - P_k)}{t_{k+1}^2} - \frac{2P'_k}{t_{k+1}} - \frac{P'_{k+1}}{t_{k+1}}, \\
 B_{4k} &= \frac{2(P_k - P_{k+1})}{t_{k+1}^3} + \frac{P'_k}{t_{k+1}^2} + \frac{P'_{k+1}}{t_{k+1}^2}.
 \end{aligned}$$

Так как P_k и P'_k – это векторные величины, то и B_i тоже векторные; если P_k и P'_k имеют x, y, z составляющие, значит, и B_i также имеет эти составляющие.

¹ У трехдиагональной матрицы элементы расположены только на главной диагонали, первой верхней и первой нижней диагоналях.

² У диагонально доминантной матрицы порядок членов на главной диагонали больше, чем других членов в этом же ряду.

В матричной форме уравнение любого сегмента сплайна k таково:

$$\begin{aligned} [B] &= \begin{bmatrix} B_{1k} \\ B_{2k} \\ B_{3k} \\ B_{4k} \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{3}{t_{k+1}^2} & -\frac{2}{t_{k+1}} & \frac{3}{t_{k+1}^2} & -\frac{1}{t_{k+1}} \\ \frac{2}{t_{k+1}^3} & \frac{1}{t_{k+1}^2} & -\frac{t_{k+1}^3}{t_{k+1}^3} & \frac{1}{t_{k+1}^2} \end{bmatrix} \begin{bmatrix} P_k \\ P'_k \\ P_{k+1} \\ P'_{k+1} \end{bmatrix}. \end{aligned} \quad (5-17)$$

Пусть требуется задать кубический сплайн, проходящий через n точек P_k , $1 \leq k \leq n$ с касательными векторами на концах P'_1 и P'_n . Из уравнения (5-16) находим внутренние касательные векторы P'_k , $2 \leq k \leq n-1$. Затем из уравнения (5-17) с известными координатами концов каждого сегмента и касательными векторами определяются B_{ik} , $1 \leq i \leq 4$, для каждого сегмента. Окончательное обобщение уравнения (5-1)

$$P_k(t) = \sum_{i=1}^4 B_{ik} t^{i-1}, \quad 0 \leq t \leq t_{k+1}, \quad 1 \leq k \leq n-1, \quad (5-18)$$

используется для расчета сегмента сплайна.

В матричном виде уравнение (5-18) выглядит следующим образом:

$$P_k(t) = [1 \quad t \quad t^2 \quad t^3] \begin{bmatrix} B_{1k} \\ B_{2k} \\ B_{3k} \\ B_{4k} \end{bmatrix}, \quad 0 \leq t \leq t_{k+1}. \quad (5-19)$$

Подставляя уравнение (5-17) и перегруппируя члены, получим

$$P_k(\tau) = [F_1(\tau) \quad F_2(\tau) \quad F_3(\tau) \quad F_4(\tau)] \begin{bmatrix} P_k \\ P_{k+1} \\ P'_k \\ P'_{k+1} \end{bmatrix} \quad \begin{array}{l} 0 \leq \tau \leq 1 \\ 1 \leq k \leq n-1 \end{array}, \quad (5-20)$$

где

$$\tau = (t/t_{k+1}),$$

$$F_{1k}(\tau) = 2\tau^3 - 3\tau^2 + 1, \quad (5-21a)$$

$$F_{2k}(\tau) = -2\tau^3 + 3\tau^2, \quad (5-21b)$$

$$F_{3k}(\tau) = \tau(\tau^2 - 2\tau + 1)t_{k+1}, \quad (5-21c)$$

$$F_{4k}(\tau) = \tau(\tau^2 - \tau)t_{k+1} \quad (5-21d)$$

называются весовыми функциями.

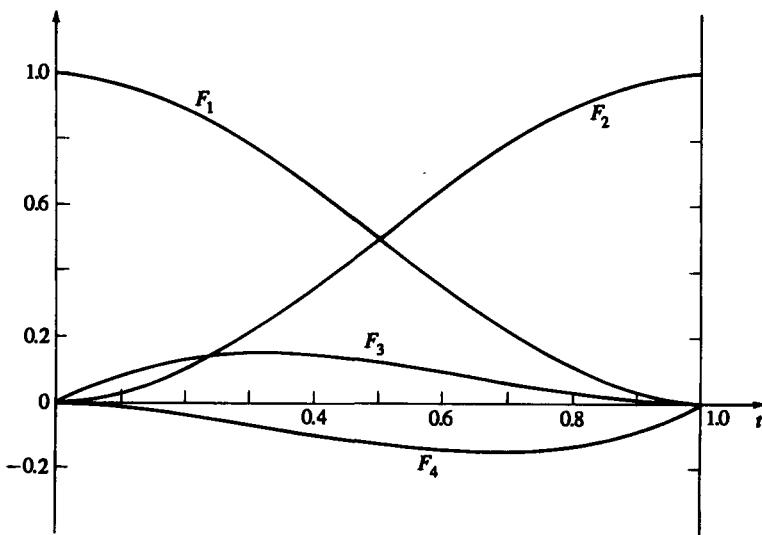


Рис. 5-8 Весовые функции кубического сплайна для $t_{k+1} = 1.0$

Пользуясь этими определениями, запишем уравнение (5-20) в матричном виде

$$P_k(\tau) = [F][G], \quad (5-22)$$

где $[F]$ — матрица весовой функции

$$[F] = [F_1(\tau) \quad F_2(\tau) \quad F_3(\tau) \quad F_4(\tau)] \quad (5-23)$$

и

$$[G]^T = [P_k \quad P_{k+1} \quad P'_k \quad P'_{k+1}] \quad (5-24)$$

содержит геометрическую информацию. Как будет видно из дальнейшего, уравнения типа (5-22), т. е. матрица весовой функции, умноженная на матрицу геометрических условий, часто применяются для описания кривых и поверхностей.

Из уравнения (5-21) видно, что каждая весовая функция имеет третий порядок. Любая точка на сегменте кубического сплайна это взвешенная сумма конечных точек и касательных векторов. Коэффициенты F_{ik} выступают в роли весовых функций. На рис. 5-8 изображены F_i для $t_{k+1} = 1.0$. Из рисунка видно, что $F_1(0) = 1$ и $F_2(0) = F_3(0) = F_4 = 0$, т. е. кривая проходит через вектор-точку P_1 . Аналогично $F_2(1) = 1$ и $F_1(1) = F_3(1) = F_4(1) = 0$, т. е. кривая также проходит через вектор-точку P_2 . Далее отметим симметрию F_1 и F_2 , и F_3 и F_4 . Фактически $F_2(\tau) = 1 - F_1(\tau)$. Наконец, обратим внимание на относительный порядок F_1 , F_2 , F_3 и F_4 . Значительная разница величин говорит о том, что в общем случае положение конечных точек имеет большее влияние, чем касательные векторы.

Вспомним, что кусочный кубический сплайн определяется точками, векторами касательных и величинами параметра, т. е. t_k в концах всех сегментов. Выбор t_k влияет на гладкость кривой.

Непрерывность второй производной в точках внутреннего соединения сама по себе не обеспечивает гладкости кривой в смысле минимальности кривизны вдоль нее. Подбирая соответствующие значения, можно минимизировать коэффициенты B_3 и B_4 для каждого сегмента и достичь большей гладкости кривой. Обычно эти дополнительные вычисления не требуются. Для практических целей применяются более простые методы, наподобие рассмотренных здесь.

Один метод вычисления t_k — установить величины параметров равными длиной хорд между соседними точками. При этом качество кривой удовлетворяет требованиям большинства прикладных задач. Другой метод состоит в том, что для нормализации вариации t_k полагается равным единице у каждого сегмента сплайна. Такой выбор упрощает вычисления (см. разд. 5-4). Как видно из приведенных выше уравнений, любой выбор t_k приводит к другим коэффициентам, и, следовательно, получаются различные кривые, проходящие через заданные точки.

Рассмотрим пример¹.

Пример 5-2 Кубический сплайн

Пусть даны четыре вектор-точки на плоскости: $P_1 [0 \ 0]$, $P_2 [1 \ 1]$, $P_3 [2 \ -1]$, $P_4 [3 \ 0]$ (см. рис. 5-9). Найти кусочный кубический сплайн, проходящий через них, используя хордовую аппроксимацию t_k . Касательные векторы в концах: $P'_1 [1 \ 1]$ и $P'_4 [1 \ -1]$. Найти промежуточные точки при $\tau = 1/3, 2/3$ для каждого сегмента.

Сначала найдем

$$t_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{(1)^2 + (1)^2} = \sqrt{2},$$

$$t_3 = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} = \sqrt{(1)^2 + (-2)^2} = \sqrt{5},$$

$$t_4 = \sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2} = \sqrt{(1)^2 + (1)^2} = \sqrt{2}.$$

Внутренние касательные векторы P'_2 и P'_3 вычисляются из уравнения (5-15):

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ t_3 & 2(t_2 + t_3) & t_2 & 0 \\ 0 & t_4 & 2(t_3 + t_4) & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ P'_4 \end{bmatrix} = \begin{bmatrix} P'_1 \\ \frac{3}{t_2 t_3} \{t_2^2 (P_3 - P_2) + t_3^2 (P_2 - P_1)\} \\ \frac{3}{t_3 t_4} \{t_3^2 (P_4 - P_3) + t_4^2 (P_3 - P_2)\} \\ P'_4 \end{bmatrix}.$$

¹ На протяжении всей этой главы рассматриваются двумерные примеры. Это сделано только для упрощения выкладок и представления результатов. Третье измерение вводится простым расширением (см. вопросы к гл. 5).

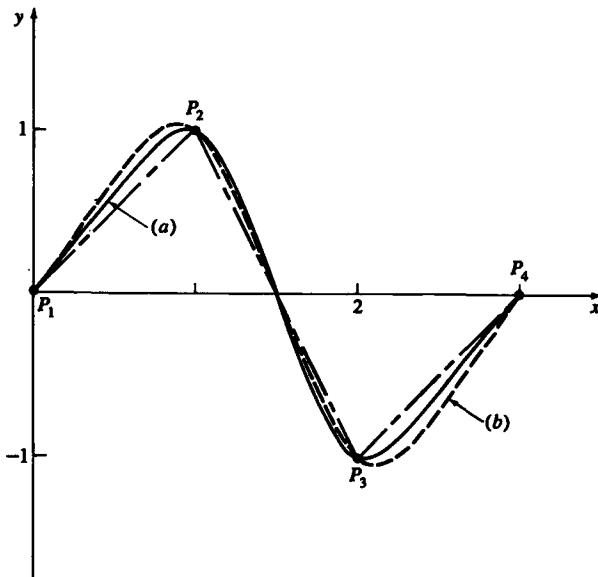


Рис. 5-9 Кусочный кубический сплайн. (а) t_k вычислены с помощью хордовой аппроксимации; (б) t_k нормализованы к 1.

Сделав подстановку, получим

$$\begin{aligned}
 & \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ \sqrt{5} & 2(\sqrt{2} + \sqrt{5}) & \sqrt{2} & 0 \\ 0 & \sqrt{2} & 2(\sqrt{5} + \sqrt{2}) & \sqrt{5} \\ 0 & 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} P'_1 \\ P'_2 \\ P'_3 \\ P'_4 \end{array} \right] \\
 = & \left[\begin{array}{c} [1 \ 1] \\ \frac{3}{\sqrt{2}\sqrt{5}} \{2([2 \ -1] - [1 \ 1]) + 5([1 \ 1] - [0 \ 0])\} \\ \frac{3}{\sqrt{5}\sqrt{2}} \{5([3 \ 0] - [2 \ -1]) + 2([2 \ -1] - [1 \ 1])\} \\ [1 \ 1] \end{array} \right] = \\
 = & \left[\begin{array}{c} [1 \ 1] \\ \frac{3}{\sqrt{2}\sqrt{5}} \{[2 \ -4] + [5 \ 5]\} \\ \frac{3}{\sqrt{5}\sqrt{2}} \{[5 \ 5] + [2 \ -4]\} \\ [1 \ 1] \end{array} \right] = \left[\begin{array}{cc} 0.949 & [1 \ 1] \\ 0.949 & [7 \ 1] \\ [1 \ 1] & [1 \ 1] \end{array} \right]
 \end{aligned}$$

или

$$\left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 2.236 & 7.300 & 1.414 & 0 \\ 0 & 1.414 & 7.300 & 2.236 \\ 0 & 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} P'_1 \\ P'_2 \\ P'_3 \\ P'_4 \end{array} \right] = \left[\begin{array}{cc} 1 & 1 \\ 6.641 & 0.949 \\ 6.641 & 0.949 \\ 1 & 1 \end{array} \right].$$

С помощью инверсии и умножения вычисляются касательные векторы

$$\begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ P'_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.318 & 0.142 & -0.028 & 0.062 \\ 0.062 & -0.028 & 0.142 & -0.318 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 6.641 & 0.949 \\ 6.641 & 0.949 \\ 1 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 1 \\ 0.505 & -0.148 \\ 0.505 & -0.148 \\ 1 & 1 \end{bmatrix}.$$

Из уравнения (5-21) находим весовые функции первого сегмента:

$$F_1(1/3) = 2(1/3)^3 - 3(1/3)^2 + 1 = \frac{20}{27} = 0.741,$$

$$F_2(1/3) = -2(1/3)^3 + 3(1/3)^2 = \frac{7}{27} = 0.259,$$

$$F_3(1/3) = (1/3)[(1/3)^2 - 2(1/3) + 1]\sqrt{2} = \frac{4\sqrt{2}}{27} = 0.210,$$

$$F_4(1/3) = (1/3)[(1/3)^2 - 1/3]\sqrt{2} = -\frac{2\sqrt{2}}{27} = -0.105$$

и второго:

$$F_1(2/3) = 2(2/3)^3 - 3(2/3)^2 + 1 = \frac{7}{27} = 0.259,$$

$$F_2(2/3) = -2(2/3)^3 + 3(2/3)^2 = \frac{20}{27} = 0.741,$$

$$F_3(2/3) = (2/3)[(2/3)^2 - 2(2/3) + 1]\sqrt{2} = \frac{2\sqrt{2}}{27} = 0.105,$$

$$F_4(2/3) = (2/3)[(2/3)^2 - 2/3]\sqrt{2} = -\frac{4\sqrt{2}}{27} = -0.210.$$

Точка на первом сегменте сплайна при $\tau = 1/3$ получается из уравнения (5-22), $P(\tau) = [F][G]$, т. е.

$$P(1/3) = [0.741 \quad 0.259 \quad 0.210 \quad -0.105] \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0.505 & -0.148 \end{bmatrix} =$$

$$= [0.416 \quad 0.484]$$

и при $\tau = 2/3$.

$$P(2/3) = [0.259 \quad 0.741 \quad 0.105 \quad -0.210] \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0.505 & -0.148 \end{bmatrix} =$$

$$= [0.740 \quad 0.876].$$

Полный результат приведен в табл. 5-1. Кубический сплайн показан на рис. 5-9.

Таблица 5-1 Результаты для кубического сплайна

Сегмент	τ	$P_x(\tau)$	$P_y(\tau)$
1	1/3	0.416	0.484
	2/3	0.740	0.876
2	1/3	1.343	0.457
	2/3	1.657	-0.457
3	1/3	2.260	-0.876
	2/3	2.584	-0.484

Хотя из весовых функций (см. (5-21)) видно, что касательные векторы в концах меньше влияют на форму сплайна, чем положение концевых точек, их эффект может оказаться значительным. На рис. 5-10 изображен один симметричный сегмент сплайна с одинаковым направлением касательных векторов, но разной величины. Направления касательных векторов обозначены углом α , а их относительная величина — длиной этих векторов. Если величина значительно меньше длины хорды l , то кривая выпукла на концах и лежит внутри треугольника из хорды и касательных. При возрастании величины кривая постепенно становится вогнутой и выходит за треугольник. В этом случае при величине вектора $3/\cos \alpha$ у кривой появляется вершина (см. рис. 5-10d). При еще больших величинах появляется петля, как видно из рис. 5-10e. Иногда для улучшения формы кривой величина вектора ограничивается длиной хорды.

5-4 НОРМАЛИЗОВАННЫЕ КУБИЧЕСКИЕ СПЛАЙНЫ

Рассмотрим другой подход к аппроксимации величин t_k -нормализацию, т. е. $0 \leq t \leq 1$ для всех сегментов.

В этом случае весовые функции (см. уравнение 5-21)¹ имеют вид:

$$F_1(t) = 2t^3 - 3t^2 + 1, \quad (5-25a)$$

$$F_2(t) = -2t^3 + 3t^2, \quad (5-25b)$$

$$F_3(t) = t^3 - 2t^2 + t, \quad (5-25c)$$

$$F_4(t) = t^3 - t^2. \quad (5-25d)$$

Матрица весовых функций для нормализованного кубического сплайна выглядит следующим образом:

$$[F] = [T][N] = [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \quad (5-26)$$

¹ Данные функции являются кубическими весовыми функциями Эрмита на интервале $0 \leq t \leq 1$.

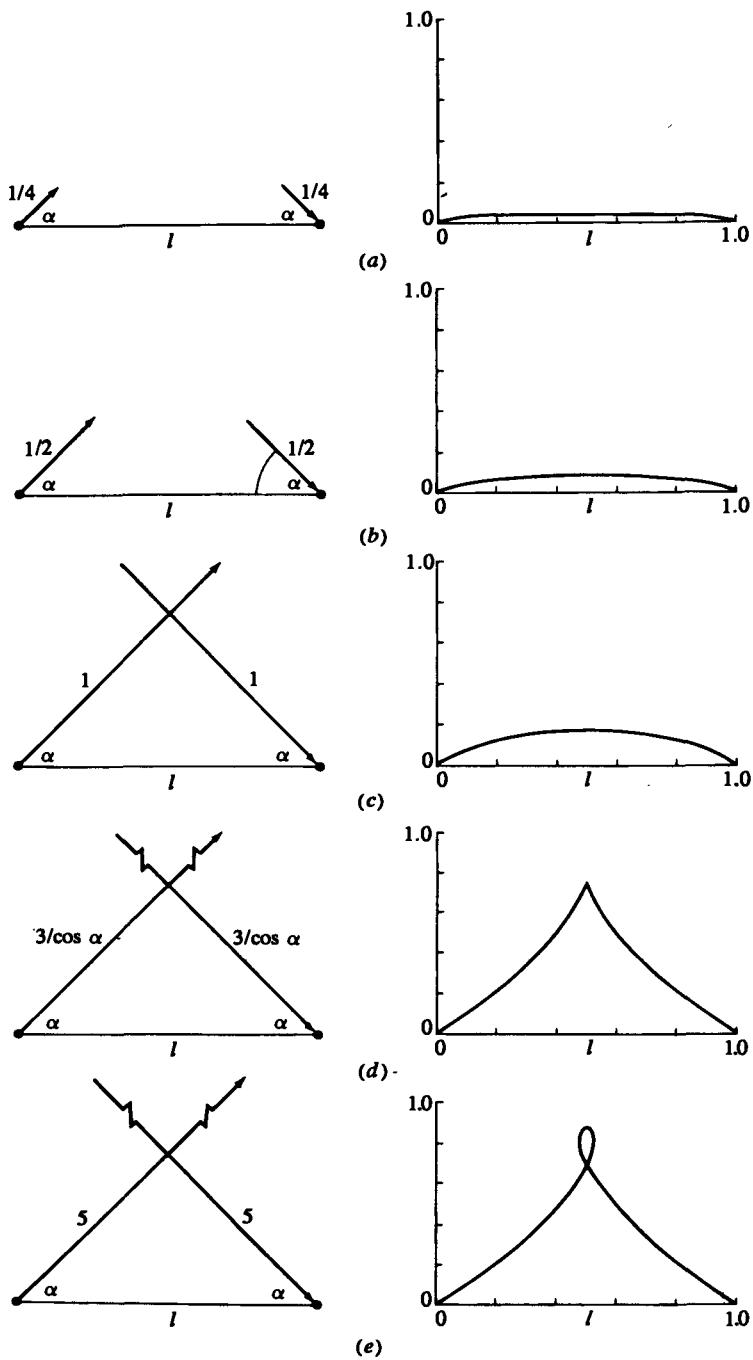


Рис. 5-10 Влияние величины касательного вектора на форму сегмента кубического сплайна, $\alpha = \pi/4$. (а) $1/4$; (б) $1/2$; (в) 1 ; (г) $3/\cos \alpha$; (е) 5 .

Матричное уравнение сегмента (уравнение (5-22)) записывается как

$$P(t) = [F][G] = [T][N][G]. \quad (5-27)$$

Заметим, что $[T]$ и $[N]$ — константы для всех сегментов сплайна. От сегмента к сегменту меняется только матрица геометрии $[G]$.

Уравнение (5-15) для определения внутренних касательных векторов, необходимых в $[G]$, приобретает вид:

$$\begin{bmatrix} 1 & 0 & \cdot & \cdot & \cdot & \cdot \\ 1 & 4 & 1 & 0 & \cdot & \cdot \\ 0 & 1 & 4 & 1 & 0 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 0 & 1 & 4 & 1 \\ \cdot & \cdot & \cdot & \cdot & 0 & 1 \end{bmatrix} \begin{bmatrix} P'_1 \\ P'_2 \\ \vdots \\ \vdots \\ P'_n \end{bmatrix} = \begin{bmatrix} 3\{(P_3 - P_2) + (P_2 - P_1)\} \\ 3\{(P_4 - P_3) + (P_3 - P_2)\} \\ \cdot \\ \cdot \\ 3\{(P_n - P_{n-1}) + (P_{n-1} - P_{n-2})\} \end{bmatrix}. \quad (5-28)$$

Решение опять дается уравнениями (5-16), но так как $[M]$ — константа, ее нужно обратить только один раз. При большом количестве точек это значительно сокращает вычисления.

Рассмотрим пример.

Пример 5-3 Нормализованный кубический сплайн

Опять рассмотрим четыре вектор-точки на плоскости из примера 5-2, т. е. $P_1[0 \ 0]$, $P_2[1 \ 1]$, $P_3[2 \ -1]$ и $P_4[3 \ 0]$ с касательными векторами $P'_1[1 \ 1]$ и $P'_4[1 \ -1]$. Найти нормализованный кусочный кубический сплайн, проходящий через эти вектор-точки.

Величины параметров $t_2 = t_3 = t_4 = 1.0$.

Касательные векторы в точках соединения рассчитываются из уравнения (5-28), т. е.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ P'_4 \end{bmatrix} = \begin{bmatrix} [1 \ 1] \\ 3\{[2 \ -1] - [1 \ 1] + [1 \ 1] - [0 \ 0]\} \\ 3\{[3 \ 0] - [2 \ -1] + [2 \ -1] - [1 \ 1]\} \\ [1 \ 1] \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 6 & -3 \\ 6 & -3 \\ 1 & 1 \end{bmatrix}.$$

Обращаем и умножаем:

$$\begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \\ P'_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.267 & 0.267 & -0.067 & 0.067 \\ 0.067 & -0.067 & 0.267 & -0.267 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 6 & -3 \\ 6 & -3 \\ 1 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & -0.8 \\ 1 & -0.8 \\ 1 & 1 \end{bmatrix}.$$

Отметим, что здесь внутренние касательные P'_2 , P'_3 существенно отличаются от результатов хордовой аппроксимации в примере 5-2.

Таблица 5-2 Результаты для нормализованного кубического сплайна

Сегмент	t	$P_x(t)$	$P_y(t)$
1	1/3	0.333	0.467
	2/3	0.667	0.933
2	1/3	1.33	0.422
	2/3	1.667	-0.422
3	1/3	2/333	-0.933
	2/3	2/667	-0.467

Из уравнения (5-26) матрица весовых коэффициентов для первого сегмента при $t = 1/3$ имеет вид:

$$[F] = [T][N] = [1/27 \quad 1/9 \quad 1/3 \quad 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \\ = [20/27 \quad 7/27 \quad 4/27 \quad -2/27]$$

и при $t = 2/3$:

$$[F] = [7/27 \quad 20/27 \quad 2/27 \quad -4/27].$$

Точка на первом сегменте сплайна при $t = 1/3$:

$$P(t) = [F][G] = [20/27 \quad 7/27 \quad 4/27 \quad -2/27] \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & -0.8 \end{bmatrix} = \\ = [1/3 \quad 63/135] = [0.333 \quad 0.467]$$

и при $t = 2/3$:

$$P(t) = [F][G] = [7/27 \quad 20/27 \quad 2/27 \quad -4/27] \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \\ 1 & -0.8 \end{bmatrix} = \\ = [2/3 \quad 126/135].$$

Результаты собраны в табл. 5-2.

На рис. 5-9 изображены два сплайна — из примера 5-1 и 5-3. Кривые получились разными, т. е. параметризация влияет на результат.

Если точки из набора данных распределены неравномерно, то опыт показывает, что хордовая аппроксимация t_k дает более гладкую кривую, чем нормализация. Это видно из рис. 5-11. На рис. 5-11a показано упорядоченное множество точек, представляющее контур петушки. На рис. 5-11b точки соединены отрезками прямых. Рисунки 5-11c и d представляют нормализованную и хордовую аппроксимацию сплайна, соответственно. Обратите внимание на разницу в изображении

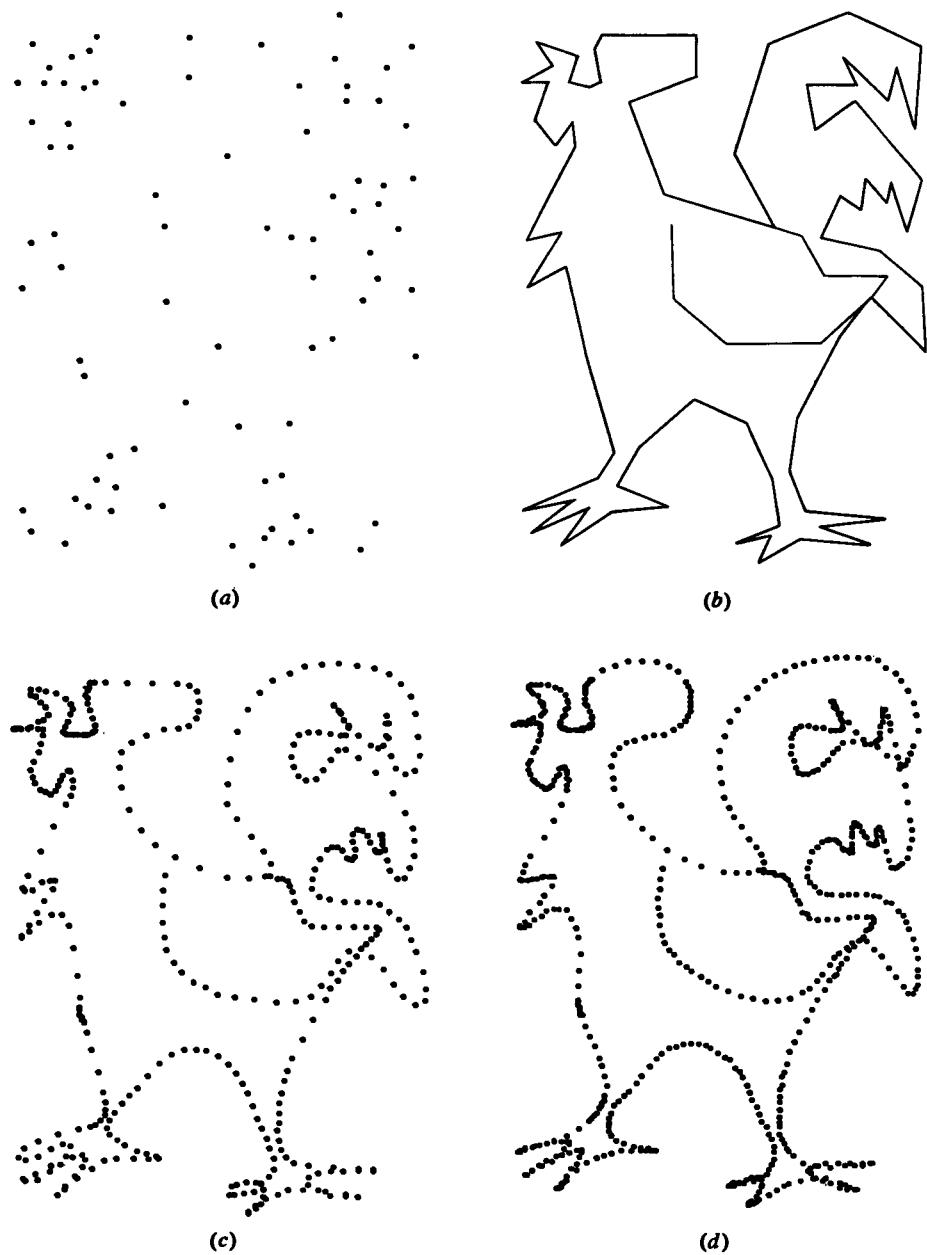


Рис. 5-11 Сравнение кубических сплайновых аппроксимаций. (а) Данные; (б) соединение прямыми; (с) нормализованная аппроксимация t_k ; (д) хордовая аппроксимация t_k .

лапок и гребешка. Хордовая аппроксимация устраниет петли, заметные на рис. 5-11с на лапках, и дает более гладкое изображение гребешка. Несмотря на эти недостатки, нормализованное представление применяется для создания кубических ограничивающих кривых для кусков трехмерной поверхности (см. гл. 6).

Еще одно интересное замечание: если для нормализованного сплайна известна матрица $[B]^T = [B_1 \ B_2 \ B_3 \ B_4]$, то геометрическая матрица сегмента сплайна $[G]$ равна

$$[G] = [N]^{-1} [B]^T,$$

где $[N]^{-1}$ — обратная матрица $[N]$:

$$[N]^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}. \quad (5-29)$$

Точки на сегменте рассчитываются из уравнения (5-27).

5-5 ДРУГИЕ ГРАНИЧНЫЕ УСЛОВИЯ

В предыдущих рассуждениях предполагалось, что известны касательные векторы P'_1 и P'_n в концевых точках кусочного кубического сплайна. Такое граничное условие называется *закрепленным*. Неизвестные касательные векторы в промежуточных точках вычисляются инверсией трехдиагональной матрицы $[M]$ из уравнения (5-15).

Если задано мало точек или физические ограничения требуют определенной формы кривой у концов, возможны другие граничные условия. Например, можно задать кривизну в концевых точках. Нулевая кривизна называется *слабым граничным условием*. Аппроксимируя кривизну второй производной и вспоминая уравнение (5-9), получаем

$$P''(t) = \sum_{i=1}^4 (i-1)(i-2)B_i t^{i-3}, \quad 0 \leq t \leq t_{k+1}.$$

В начале первого сегмента сплайна $t = 0$. Отметим, что на результат влияет только член $i = 3$. Из уравнения (5-17)

$$P''(0) = 2B_3 = 2 \left(\frac{3(P_2 - P_1)}{t_2^2} - \frac{2P'_1}{t_2} - \frac{P'_2}{t_2} \right) = 0.$$

После перегруппировки получаем

$$P'_1 + \frac{P'_2}{2} = \frac{3}{2t_2}(P_2 - P_1). \quad (5-30)$$

Теперь первая строка в матрицах $[M]$ и $[R]$ (см. (5-15)) такова:

$$[1 \ 1/2 \ 0 \ \cdots] [P'_1] = \left[\frac{3}{2t_2}(P_2 - P_1) \right].$$

В конце последнего сегмента $t = t_n$, $k = n - 1$. Здесь результат зависит от двух членов $i = 3$ и $i = 4$ в (5-9), а именно

$$P''(t_n) = 2B_3 + 6B_4t_n = 0$$

или, используя уравнение (5-17),

$$\frac{6}{t_n^2}(P_{n-1} - P_n) + \frac{2P'_{n-1}}{t_n} + \frac{4P'_n}{t_n} = 0.$$

После перегруппировки

$$2P'_{n-1} + 4P'_n = \frac{6}{t_n}(P_n - P_{n-1}). \quad (5-31)$$

Последняя строка матриц $[M]$ и $[R]$ (см. (5-15)) выглядит теперь так:

$$\begin{bmatrix} \cdots & 0 & 2 & 4 \end{bmatrix} [P'_n] = \left\lfloor \frac{6}{t_n}(P_n - P_{n-1}) \right\rfloor.$$

Заметим, что для сплайна слабые и закрепленные граничные условия могут меняться местами.

Пример 5-4 Кубический сплайн со слабым граничным условием

Пусть заданы три точки $P_1[0 \ 0]$, $P_2[1 \ 2]$ и $P_3[3 \ 2]$. Найти кубический сплайн, проходящий через них, со слабыми граничными условиями. Для t_k использовать хордовую аппроксимацию.

Коэффициенты t_k таковы:

$$\begin{aligned} t_2 &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{(1)^2 + (2)^2} = \sqrt{5}, \\ t_3 &= \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} = \sqrt{(3 - 1)^2 + (0)^2} = 2. \end{aligned}$$

Модифицируя матрицы $[M]$ и $[R]$ с помощью уравнений (5-30) и (5-31), получим уравнение внутреннего касательного вектора в P'_2 :

$$\begin{aligned} \begin{bmatrix} 1 & 0.5 & 0 \\ 2 & 8.472 & 2.236 \\ 0 & 2 & 4 \end{bmatrix} \begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \end{bmatrix} &= \begin{bmatrix} \frac{3}{2\sqrt{5}}([1 \ 2] - [0 \ 0]) \\ \frac{3}{2\sqrt{5}}\{5([3 \ 2] - [1 \ 2]) + 4([1 \ 2] - [0 \ 0])\} \\ 3([3 \ 2] - [1 \ 2]) \end{bmatrix} \\ &= \begin{bmatrix} 0.671 & 1.342 \\ 0.391 & 5.367 \\ 6 & 0 \end{bmatrix}. \end{aligned}$$

Инвертируя (3×3) -матрицу и умножая, найдем производные

$$\begin{bmatrix} P'_1 \\ P'_2 \\ P'_3 \end{bmatrix} = \begin{bmatrix} 1.157 & -0.079 & 0.044 \\ -0.315 & 0.157 & -0.088 \\ 0.157 & -0.079 & 0.294 \end{bmatrix} \begin{bmatrix} 0.671 & 1.342 \\ 9.391 & 5.367 \\ 6 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0.301 & 1.131 \\ 0.739 & 0.422 \\ 1.131 & -0.211 \end{bmatrix}.$$

Из уравнения (5-21) весовые функции при $\tau = 1/3$ и $2/3$ для первого сегмента имеют вид:

$$[F]_{\tau=1/3} = [0.741 \quad 0.259 \quad 0.331 \quad -0.166],$$

$$[F]_{\tau=2/3} = [0.259 \quad 0.741 \quad 0.166 \quad -0.331].$$

Затем уравнение (5-22) дает точки на первом сегменте, т. е.

$$P(1/3) = [0.741 \quad 0.259 \quad 0.331 \quad -0.166] \begin{bmatrix} 0 & 0 \\ 1 & 2 \\ 0.301 & 1.131 \\ 0.739 & 0.422 \end{bmatrix} =$$

$$= [0.237 \quad 0.823].$$

Аналогично

$$P(2/3) = [0.546 \quad 1.529].$$

Весовые функции при $\tau = 1/3, 2/3$ для второго сегмента таковы:

$$[F]_{\tau=1/3} = [0.741 \quad 0.259 \quad 0.296 \quad -0.148],$$

$$[F]_{\tau=2/3} = [0.259 \quad 0.741 \quad 0.148 \quad -0.296]$$

и уравнение (5-22) дает

$$P(1/3) = [1.570 \quad 2.156],$$

$$P(2/3) = [2.256 \quad 2.125].$$

Результаты показаны на рис. 5-12.

Интерес представляют еще два типа граничных условий: циклическое и ациклическое. Циклическое условие порождает замкнутую кривую или часть периодически повторяющейся кривой. Для этого необходимо, чтобы касательная вектора и кривизна на обоих концах были нулевыми:

$$P'_1(0) = P'_n(t_n), \tag{5-32}$$

$$P''_1(0) = P''_n(t_n). \tag{5-33}$$

Из уравнений (5-3) и (5-9), пользуясь уравнениями (5-17) и (5-32), получаем

$$P'_1 - P'_{n-1} = 2t_n \left[\frac{3(P_n - P_{n-1})}{t_n^2} - \frac{2P'_{n-1}}{t_n} - \frac{P'_n}{t_n} \right] +$$

$$+ 3t_n \left[\frac{2(P_{n-1} - P_n)}{t_n^3} + \frac{P'_{n-1}}{t_n^2} + \frac{P'_n}{t_n^2} \right]. \tag{5-34}$$

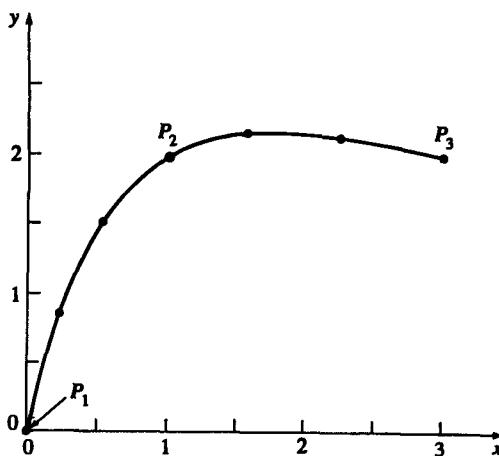


Рис. 5-12 Кубический сплайн для примера 5-4.

Аналогично, уравнение (5-33) приводится к виду

$$2 \left[\frac{3(P_2 - P_1)}{t_2^2} - \frac{2P'_1}{t_2} - \frac{P'_2}{t_2} \right] = 2 \left[\frac{3(P_n - P_{n-1})}{t_n^2} - \frac{2P'_{n-1}}{t_n} - \frac{P'_n}{t_n} \right] + \\ + 6t_n \left[\frac{2(P_{n-1} - P_n)}{t_n^3} + \frac{P'_{n-1}}{t_n^2} + \frac{P'_n}{t_n^2} \right]. \quad (5-35)$$

Умножив (5-35) на t_n и вычитая его из (5-34), получим

$$P'_1 - P'_{n-1} - 2t_n \left[\frac{3(P_2 - P_1)}{t_2^2} - \frac{2P'_1}{t_2} - \frac{P'_2}{t_2} \right] = \\ = 3t_n^2 \left[\frac{2(P_{n-1} - P_n)}{t_n^3} + \frac{P'_{n-1}}{t_n^2} + \frac{P'_n}{t_n^2} \right] - \\ - 6t_n^2 \left[\frac{2(P_{n-1} - P_n)}{t_n^3} + \frac{P'_{n-1}}{t_n^2} + \frac{P'_n}{t_n^2} \right]$$

Вспомним, что $P'_1 = P'_n$, и перегруппируем члены:

$$2 \left(1 + \frac{t_n}{t_2} \right) P'_1 + \frac{t_n}{t_2} P'_2 + P'_{n-1} = 3 \frac{t_n}{t_2^2} (P_2 - P_1) - \frac{3}{t_n} (P_{n-1} - P_n). \quad (5-36)$$

Касательные векторы в точках внутреннего соединения опять получаются с помощью уравнения (5-15). Однако из-за того, что касательные векторы линейно зависимы ($P'_1 = P'_n$), матрица $[M]$ теперь имеет размер $(n - 1) \times (n - 1)$, где

первая строка состоит из коэффициентов уравнения (5-36):

$$\left[\begin{array}{cccccc} 2\left(1 + \frac{t_n}{t_2}\right) & \frac{t_n}{t_2} & 0 & 0 & \cdot & 1 \\ t_3 & 2(t_2 + t_3) & t_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 0 & t_n & 2(t_n + t_{n-1}) & t_{n-1} & \end{array} \right] \left[\begin{array}{c} P'_1 \\ P'_2 \\ P'_3 \\ \cdot \\ \cdot \\ P'_{n-1} \end{array} \right] =$$

$$= \left[\begin{array}{c} 3\frac{t_n}{t_2^2}(P_2 - P_1) - \frac{3}{t_n}(P_{n-1} - P_n) \\ \frac{3}{t_2^2}\{t_2^2(P_3 - P_2) + t_3^2(P_2 - P_1)\} \\ \frac{3}{t_2 t_3}\{t_3^2(P_4 - P_3) + t_4^2(P_3 - P_2)\} \\ \frac{3}{t_3 t_4}\{t_4^2(P_5 - P_4) + t_5^2(P_4 - P_3)\} \\ \cdot \\ \cdot \\ \cdot \\ \frac{3}{t_{n-1} t_n}\{t_{n-1}^2(P_n - P_{n-1}) + t_n^2(P_{n-1} - P_{n-2})\} \end{array} \right]. \quad (5-37)$$

Эта матрица уже не трехдиагональная. Ациклический сплайн похож на циклический за исключением того, что

$$P'_1(0) = -P'_n(t_n), \quad (5-38)$$

$$P''_1(0) = -P''_n(t_n). \quad (5-39)$$

Та же процедура, что и для циклических граничных условий, дает

$$2\left(1 + \frac{t_n}{t_2}\right)P'_1 + \frac{t_n}{t_2}P'_2 - P'_{n-1} = 3\frac{t_n}{t_2^2}(P_2 - P_1) + \frac{3}{t_n}(P_{n-1} - P_n). \quad (5-40)$$

Из уравнения (5-40) видно, что единственное отличие — это разные знаки у 1 в $M(1, n-1)$ в матрице $[M]$ и у второго члена $R(1, 1)$ в уравнении (5-37). Ациклические спайны полезны при изображении кривых, у которых касательные векторы в концах имеют одинаковую величину и противоположные направления, например как у многослойной деревянной теннисной ракетки. В табл. 5-3 собраны граничные условия для кубических сплайнов.

У кубических сплайнов первые и вторые производные непрерывны для любых граничных условий, но с увеличением количества заданных точек время обращения матрицы касательных векторов может стать слишком большим.

На рис. 5-13 изображены два кубических сплайна, проходящих через пять точек: с закрепленными и слабыми граничными условиями. Касательные векторы в закрепленных концах — $[-1 \ 1]$ и $[1 \ 1]$, соответственно. В этом случае различие кривых несущественно. Другие касательные векторы, например $[1 \ 1]$ и $[1 \ 1]$, могут существенно изменить форму сплайна с закрепленными концами.

На рис. 5-14 показано влияние изменения величины, а не направления касательных векторов в концах замкнутого кубического сплайна. Эти кривые симметричны, так как квадратная матрица в уравнении (5-15) трехдиагональна.

Таблица 5-3 Граничные условия для кубических сплайнов

Граничное условие	Ненулевые элементы в первой и последней строках $[M]$	Первая и последняя строки $[R]$
Закрепленное	$M(1, 1) = 1$ $M(n, n) = 1$	$R(1, 1) = P'_1$ $R(n, 1) = P'_n$
Слабое	$M(1, 1) = 1$ $M(1, 2) = 1/2$ $M(n, n-1) = 2$ $M(n, n) = 4$	$R(1, 1) = \frac{3}{2t_2}(P_2 - P_1)$ $R(n, 1) = \frac{6}{t_n}(P_n - P_{n-1})$
Циклическое	$M(1, 1) = 2\left(1 + \frac{t_n}{t_2}\right)$ $M(1, 2) = \frac{t_n}{t_2}$ $M(1, n-1) = 1$	$R(1, 1) = 3\left(\frac{t_n}{t_2^2}\right)(P_2 - P_1)$ $-\frac{3}{t_n}(P_{n-1} - P_n)$ $R(n, 1)$ не определено
Ациклическое	$M(1, 1) = 2\left(1 + \frac{t_n}{t_2}\right)$ $M(1, 2) = \frac{t_n}{t_2}$ $M(1, n-1) = -1$	$R(1, 1) = 3\left(\frac{t_n}{t_2^2}\right)(P_2 - P_1)$ $+\frac{3}{t_n}(P_{n-1} - P_n)$ $R(n, 1)$ не определено

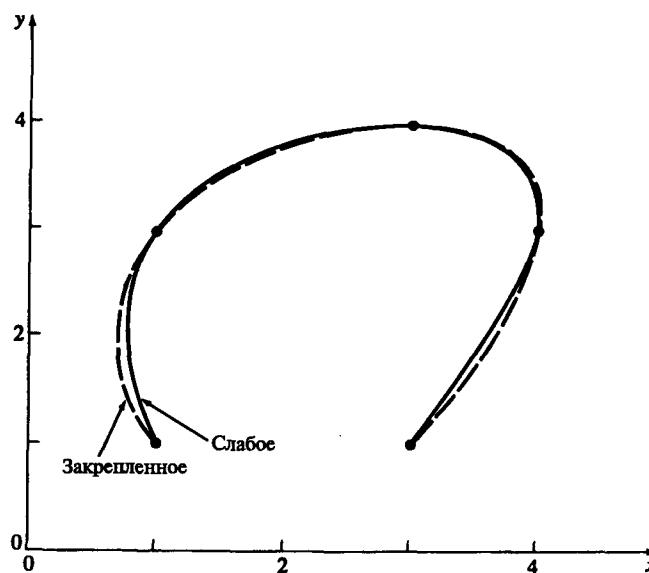


Рис. 5-13 Сравнение закрепленного и слабого граничных условий для нормализованного кусочно кубического сплайна.

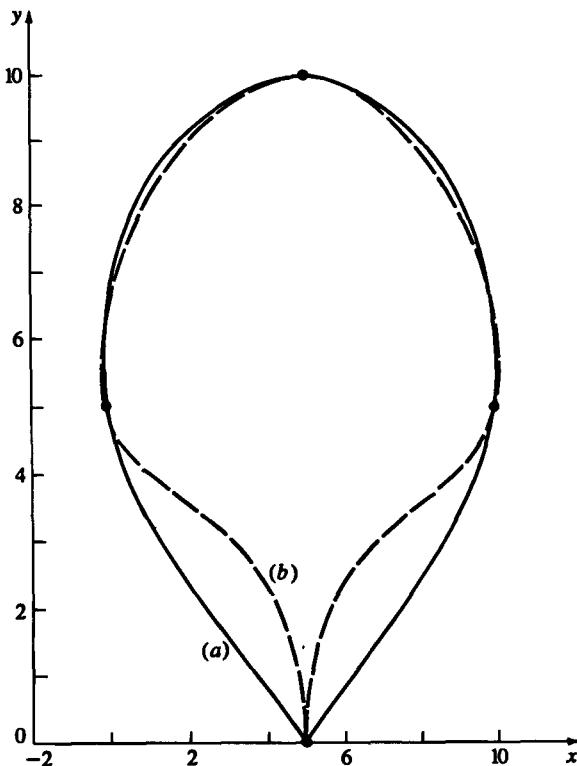


Рис. 5-14 Влияние величины касательного вектора на форму нормализованного кусочно кубического сплайна. (а) $P'_1[0 \quad 1]$, $P'_2[0 \quad -1]$; (б) $P'_1[0 \quad 10]$, $P'_2[0 \quad -10]$.

На рис. 5-15 через те же пять точек проходит сплайн со слабыми и циклическими граничными условиями. Обратите внимание на то, что сплайн со слабыми условиями симметричен, а циклический — нет.

На рис. 5-16 сравниваются слабое и циклическое граничные условия. Циклическое условие можно использовать и для незамкнутых кривых: направление начального касательного вектора должно совпадать с направлением касательного вектора в конце.

Несмотря на то, что параметрические кусочные сплайны удобны и применяются в ряде отраслей (производстве автомобилей, судостроении и авиастроении), у них есть некоторые недостатки. Так, они неточно представляют конические сечения и асимптотические кривые и часто приводят к видимой осцилляции. Осцилляция возникает из-за того, что сплайн испытывает локальное влияние каждой точки, а третья производная только кусочно постоянна. Разрывы третьей производной порождают отклонения, и полученная кривая не является гладкой, несмотря на то, что ее вторая производная непрерывна.

Один из методов борьбы с осцилляциями — математический аналог приложения силы к концам сплайна. Пусть имеется физический сплайн — тонкая, гибкая

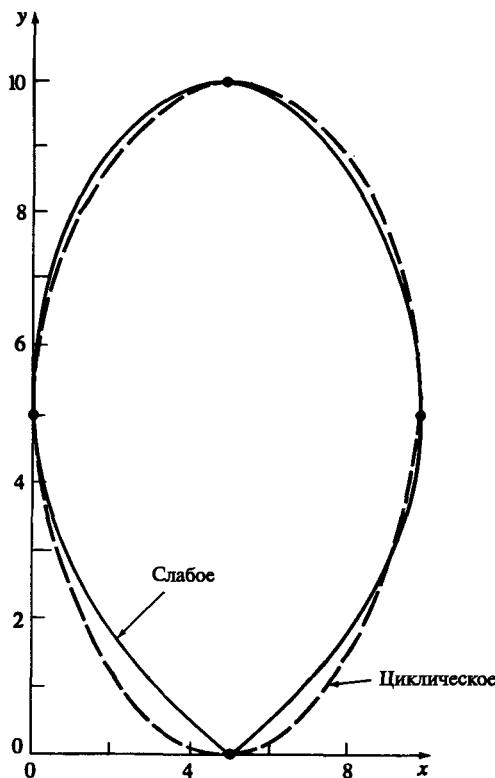


Рис. 5-15 Сравнение слабого и циклического граничных условий для нормализованного кусочно кубического сплайна.

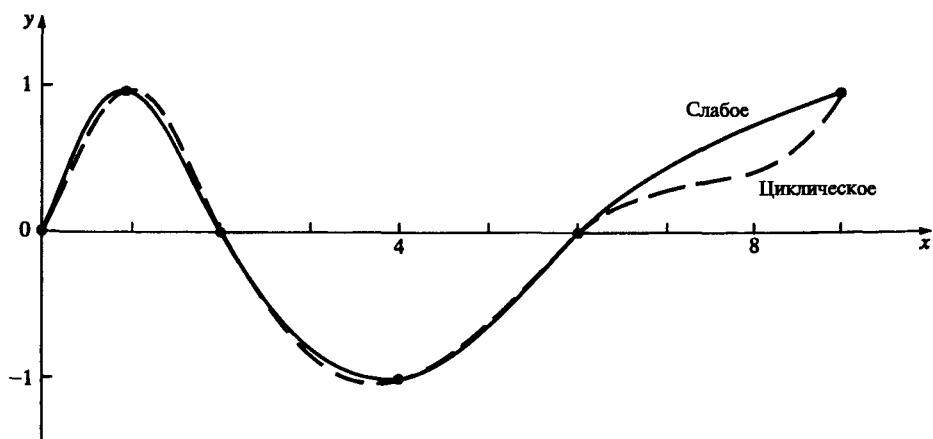


Рис. 5-16 Сравнение слабого и циклического граничных условий для открытых кривых.

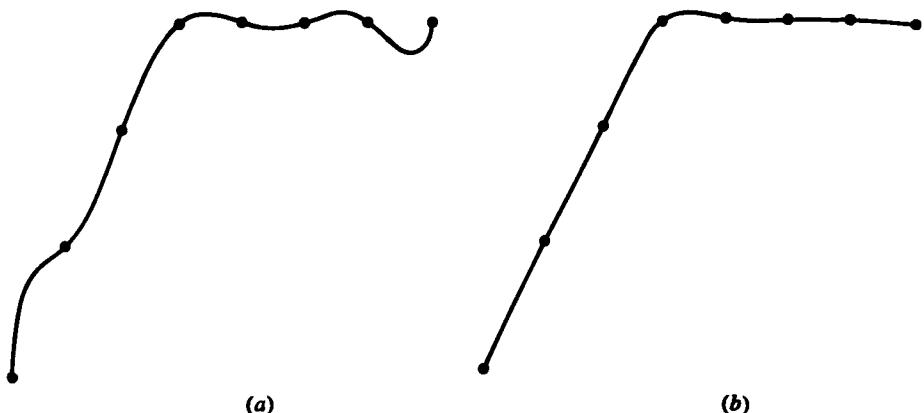


Рис. 5-17 Эффект напряжения кусочно кубического сплайна. (a) Без напряжения; (b) с напряжением.

рейка, нагруженная в некоторых точках. Чтобы устраниТЬ малые колебания, нужно приложить силу к концам рейки. Изучение нагруженных сплайнов выходит за рамки этой книги, однако мы сделаем несколько замечаний. В первых разработках [5-6] и [5-7] рассматривается экспоненциальный сплайн, требующий больших вычислений. В работе [5-8] рассматривается другой полиномиальный вид сплайна, ν -сплайн. Эффект напряжения кубического сплайна показан на рис. 5-17.

5-6 ПАРАБОЛИЧЕСКАЯ ИНТЕРПОЛЯЦИЯ

Кубические сплайны — это мощное и удобное средство, но и они небезупречны: необходимо учитывать влияние направления и величины касательных векторов, указывать все точки кривой до ее изображения, невозможна локальная коррекция кривой. Последнее особенно важно для интерактивной работы. Расчет кубического сплайна требует обращения большой матрицы, зависящей от всех элементов сплайна; т. е. изменение любого сегмента затрагивает все остальные сегменты. Воздействие уменьшается при удалении от точки возмущения, но полностью пре-небречь им нельзя. Параболическая интерполяция разрешает большинство этих проблем за счет того, что она только непрерывна, т. е. в точках соединения сегментов сохраняется непрерывность лишь первой производной. Для многих прикладных задач этого достаточно, причем параболическая интерполяция не требует больших расчетов.

Параболическая интерполяция была разработана Оверхаузером¹ [5-9]. Оверхаузер строил кривую интерполяции, исходя из геометрических соображений. Идея состоит в линейной интерполяции пересекающихся частей двух парабол. Параболы заданы четырьмя последовательными точками: первая — тремя первыми точками, вторая — тремя последними. Пересечение лежит между второй и третьей

¹ В первом издании данной книги был рассмотрен метод Оверхаузера.

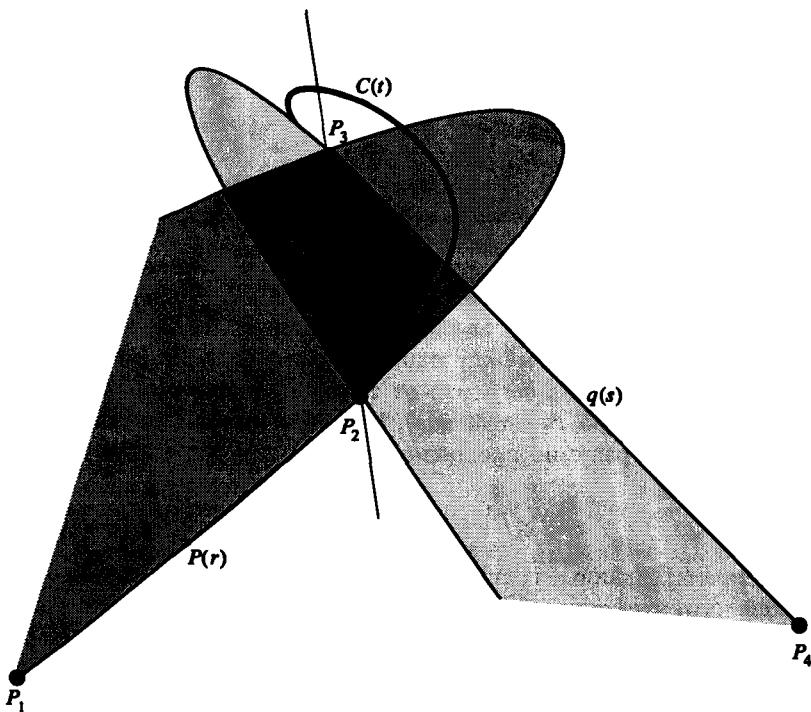


Рис. 5-18 Параболическая интерполяция.

точками. Несмотря на то, что параболы — плоские кривые, их линейная интерполяция это кубическая пространственная кривая, как показано на рис. 5-18.

В работе [5-10] рассматривается метод построения одного из семейств параболически интерполированных кривых с помощью матриц. Этот алгоритм требует меньше вычислений. Рассмотрим обобщенный вывод для всего семейства.

Параболически интерполированная кривая имеет вид

$$C(t) = (1 - t)p(r) + tq(s), \quad (5-41)$$

где \$r, s, t\$ — параметры, \$p(r), q(s)\$ — параметрические параболы, проходящие через \$P_1, P_2, P_3\$ и \$P_2, P_3, P_4\$, соответственно, как показано на рис. 5-19. Для простоты параболы на рис. 5-19 лежат в одной плоскости, но это не обязательно (рис. 5-18).

Параметрическое представление \$p(r)\$ и \$q(s)\$ следующее:

$$p(r) = [r^2 \quad r \quad 1][B], \quad (5-42)$$

$$q(s) = [s^2 \quad s \quad 1][D], \quad (5-43)$$

где \$[B]\$ и \$[D]\$ — матрицы, представляющие положение вектор-точек \$P_1, P_2, P_3\$ и \$P_2, P_3, P_4\$, соответственно. Результат интерполяции — кубическая кривая

$$C(t) = [t^3 \quad t^2 \quad t \quad 1][A][G] = [T][A][G], \quad (5-44)$$

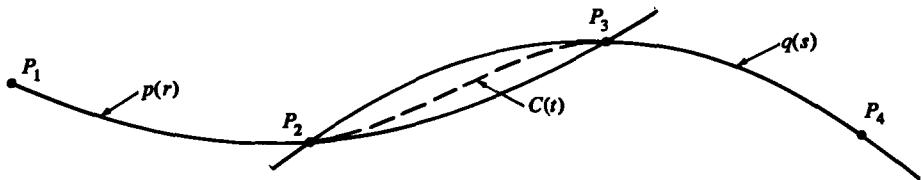


Рис. 5-19 Обозначение для параболической интерполяции.

где $[T][A]$ — матрица интерполяционных функций, а $[G]$ — геометрическая матрица вектор-точек P_1, P_2, P_3, P_4 . Чтобы определить $[B]$ и $[D]$, а затем $[A]$ и $[G]$, необходимо установить связь между параметрами r, s, t . Из рис. 5-19, замечая, что r меняется от 0 до 1 на сегменте от P_1 до P_3 вдоль $p(r)$, s меняется от 0 до 1 на сегменте от P_2 до P_4 вдоль $q(s)$, и t меняется от 0 до 1 на сегменте от P_2 до P_3 вдоль $C(t)$, разумно предположить, что r и t , а также s и t связаны линейно. Отсюда

$$r = k_1 t + k_2, \quad s = k_3 t + k_4, \quad (5-45)$$

где k_i — константы, заданные граничными условиями в вектор-точках P_1, P_2, P_3 и P_4 . Предположим, что данные распределены равномерно или почти равномерно, и диапазон параметров нормализован, т. е. $0 \leq r, s, t \leq 1$. Тогда можно условиться, что

$$p(0) = P_1, \quad p(1/2) = P_2, \quad p(1) = P_3, \quad (5-46a)$$

$$q(0) = P_2, \quad q(1/2) = P_3, \quad q(1) = P_4, \quad (5-46b)$$

$$C(0) = P_2, \quad C(1) = P_3. \quad (5-46c)$$

Здесь основные предположения таковы: $p(r) = P_2$ для $r = 1/2$ и $q(s) = P_3$ для $s = 1/2$. Эти предположения сделаны в работе [5-10]. В результате получаем единственный член семейства параболически интерполированных кривых, как будет показано ниже.

В предположениях уравнений (5-45)

$$@ P_2 : r = 1/2, t = 0 \Rightarrow k_2 = 1/2$$

$$@ P_3 : r = 1, t = 1 \Rightarrow k_1 + k_2 = 1 \Rightarrow k_1 = 1/2$$

$$@ P_2 : s = 0, t = 0 \Rightarrow k_4 = 0$$

$$@ P_3 : s = 1/2, t = 1 \Rightarrow k_3 = 1/2$$

Итак

$$r(t) = \frac{1}{2}(1+t), \quad s(t) = \frac{1}{2}t. \quad (5-47)$$

Вспомним уравнение (5-42) и используем уравнение (5-46a), чтобы выразить $[B]$ через P_1, P_2, P_3 ,

$$p(0) = P_1 = [0 \ 0 \ 1][B], \quad (5-48a)$$

$$p(1/2) = P_2 = [1/4 \ 1/2 \ 1][B], \quad (5-48b)$$

$$p(1) = P_3 = [1 \ 1 \ 1][B]. \quad (5-48c)$$

Запишем в виде одной матрицы

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1/4 & 1/2 & 1 \\ 1 & 1 & 1 \end{bmatrix} [B] = [M][B].$$

Отсюда

$$[B] = [M]^{-1} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}. \quad (5-49)$$

Аналогично через P_2 , P_3 , P_4 находится выражение $[D]$. Пользуясь уравнением (5-46b), получаем

$$q(0) = P_2 = [0 \ 0 \ 1][D], \quad (5-50a)$$

$$q(1/2) = P_3 = [1/4 \ 1/2 \ 1][D], \quad (5-50b)$$

$$q(1) = P_4 = [1 \ 1 \ 1][D]. \quad (5-50c)$$

Сравнение с уравнениями (5-48) сразу же дает

$$[D] = [M]^{-1} \begin{bmatrix} P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} 2 & -4 & 2 \\ -3 & 4 & -1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_2 \\ P_3 \\ P_4 \end{bmatrix}. \quad (5-51)$$

Вспомним уравнение (5-41) и подставим уравнения (5-42) и (5-43):

$$C(t) = (1-t)[r^2 \ r \ 1][B] + t[s^2 \ s \ 1][D].$$

Используем уравнение (5-47), чтобы переписать это только в терминах параметра t ,

$$C(t) = \left[-\frac{1}{4}(t^2 + t^2 - t - 1) \quad -\frac{1}{2}(t^2 - 1) \quad 1 - t \right] [B] + \left[\frac{t^3}{4} \quad \frac{t^2}{2} \quad t \right] [D].$$

Подставив $[B]$ и $[D]$ из уравнений (5-49) и (5-51), получим

$$\begin{aligned} C(t) = & \left[-\frac{t^3}{2} + t^2 - \frac{t}{2} \quad t^3 - t^2 - t + 1 \quad -\frac{t^3}{2} + \frac{t}{2} \right] \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} + \\ & + \left[\frac{t^3}{2} - \frac{3}{2}t^2 + t \quad -t^3 + 2t^2 \quad \frac{t^3}{2} - \frac{t^2}{2} \right] \begin{bmatrix} P_2 \\ P_3 \\ P_4 \end{bmatrix}. \end{aligned}$$

Перепишем уравнение так, чтобы включить все четыре точки P_1, P_2, P_3, P_4 :

$$C(t) = \begin{bmatrix} -\frac{t^3}{2} + t^2 - \frac{t}{2} & t^3 - t^2 - t + 1 & -\frac{t^3}{2} + \frac{t}{2} & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \\ + \begin{bmatrix} 0 & \frac{t^3}{2} - \frac{3}{2}t^2 + t & -t^3 + 2t^2 & \frac{t^3}{2} - \frac{t^2}{2} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}.$$

Наконец, перепишем результат в форме уравнения (5.44)

$$C(t) = [t^3 \quad t^2 \quad t \quad 1][A][G] = [T][A][G], \quad (5.44)$$

где

$$[A] = \left(\frac{1}{2}\right) \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \quad (5.52)$$

и

$$[G]^T = [P_1 \quad P_2 \quad P_3 \quad P_4]. \quad (5.53)$$

Заметим, что снова (см. разд. 5-3 и уравнение (5-22)) результат имеет вид произведения матрицы интерполяционных функций и геометрической матрицы. Интерполяционные функции $[F] = [T][A]$ показаны на рис. 5-20. Реализация алгоритма очень проста. Рассмотрим подробности на примере.

Пример 5-5 Параболическая интерполяция

Рассмотрим снова четыре вектор-точки $P_1[0 \quad 0]$, $P_2[1 \quad 1]$, $P_3[2 \quad -1]$ и $P_4[3 \quad 0]$ из предыдущих примеров 5-2 и 5-3 (см. рис. 5-21). Найти параболическую интерполяцию между P_2 и P_3 . Вычислить промежуточные точки при $t = 1/3, 2/3$.

Из уравнений (5-44), (5-52) и (5-53) для $t = 1/3$ получаем

$$C\left(\frac{1}{3}\right) = \left(\frac{1}{2}\right) \begin{bmatrix} \frac{1}{27} & \frac{1}{9} & \frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 2 & -1 \\ 3 & 0 \end{bmatrix} = \\ = \begin{bmatrix} \frac{4}{3} & \frac{4}{9} \end{bmatrix}.$$

Аналогично при $t = 2/3$ имеем $C(2/3) = [5/3 - 4/9]$. Полученная кривая изображена на рис. 5-21.

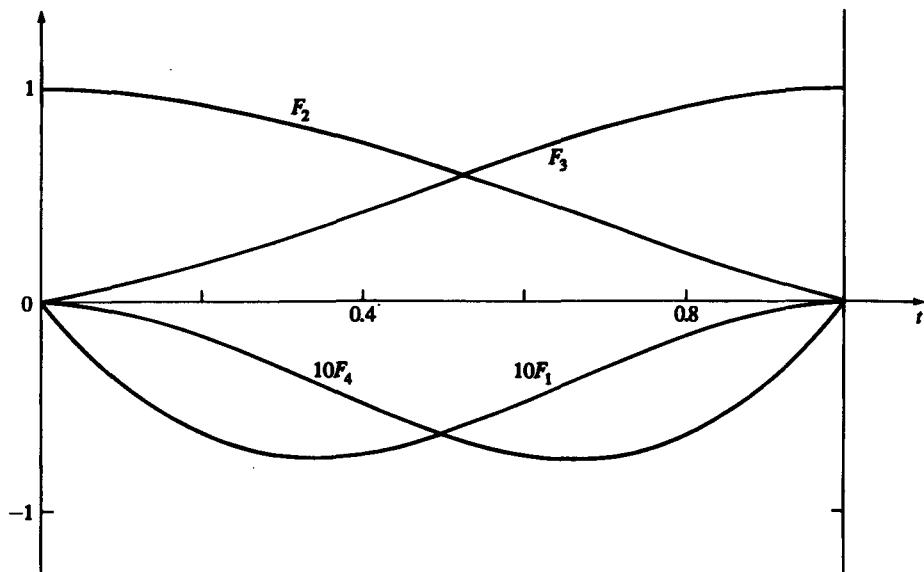


Рис. 5-20 Весовые функции для параболически интерполированных кривых в случае $\alpha = \beta = 0.5$.

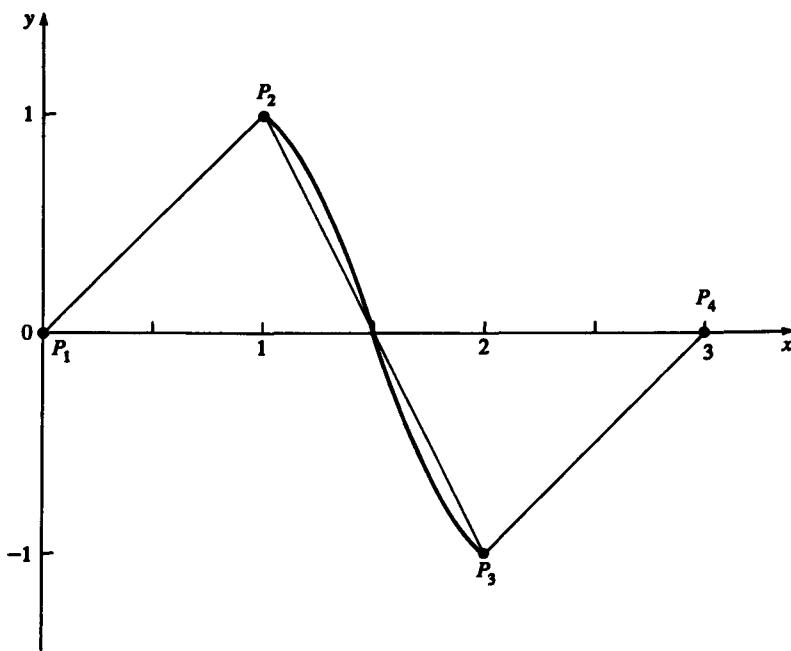


Рис. 5-21 Параболически интерполированная кривая для примера 5-5.

5-7 ОБОБЩЕННАЯ ПАРАБОЛИЧЕСКАЯ ИНТЕРПОЛЯЦИЯ

В примере параболической интерполяции в разд. 5-6 предполагается, что параметры r и s в точках P_2 и P_3 , соответственно, равны $1/2$. Если данные распределены неравномерно, кривая становится менее гладкой. Более общим предположением была бы нормализованная хордовая аппроксимация. Пусть

$$\alpha = \frac{|[P_2] - [P_1]|}{|[P_3] - [P_2]| + |[P_2] - [P_1]|}, \quad 0 < \alpha < 1, \quad (5-54)$$

и

$$\beta = \frac{|[P_3] - [P_2]|}{|[P_4] - [P_3]| + |[P_3] - [P_2]|}, \quad 0 < \beta < 1. \quad (5-55)$$

Тогда уравнения (5-46) принимает вид

$$p(0) = P_1, \quad p(\alpha) = P_2, \quad p(1) = P_3, \quad (5-56a)$$

$$q(0) = P_2, \quad q(\beta) = P_3, \quad q(1) = P_4, \quad (5-56b)$$

$$C(0) = P_2, \quad C(1) = P_3. \quad (5-56c)$$

В этих предположениях получаем линейные выражения $r(t)$ и $s(t)$ (см. (5-47))

$$r(t) = (1 - \alpha)t + \alpha, \quad s(t) = \beta t. \quad (5-57)$$

Из уравнений (5-42) и (5-56a)

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \alpha^2 & \alpha & 1 \\ 1 & 1 & 1 \end{bmatrix} [B] = [M][B].$$

Отсюда

$$[B] = [M]^{-1} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha} & \frac{-1}{\alpha(1-\alpha)} & \frac{1}{1-\alpha} \\ \frac{-(1+\alpha)}{\alpha} & \frac{1}{\alpha(1-\alpha)} & \frac{-\alpha}{1-\alpha} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}. \quad (5-58)$$

Аналогично

$$[D] = \begin{bmatrix} \frac{1}{\beta} & \frac{-1}{\beta(1-\beta)} & \frac{1}{1-\beta} \\ \frac{-(1+\beta)}{\beta} & \frac{1}{\beta(1-\beta)} & \frac{-\beta}{1-\beta} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_2 \\ P_3 \\ P_4 \end{bmatrix}. \quad (5-59)$$

Вспомним уравнение (5-41) и подставим уравнения (5-58) и (5-59):

$$\begin{aligned}
 C(t) &= (1-t)[\{(1-\alpha)t+\alpha\}^2 \quad (1-\alpha)t+\alpha \quad 1] \times \\
 &\quad \times \begin{bmatrix} 1 & -1 & 1 \\ \frac{\alpha}{\alpha(1-\alpha)} & \frac{1}{\alpha(1-\alpha)} & \frac{-\alpha}{1-\alpha} \\ \frac{-(1+\alpha)}{\alpha} & \frac{1}{\alpha(1-\alpha)} & \frac{0}{1-\alpha} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} + \\
 &\quad + t[(\beta t)^2 \quad \beta t \quad 1] \begin{bmatrix} 1 & -1 & 1 \\ \frac{\beta}{\beta(1-\beta)} & \frac{1}{\beta(1-\beta)} & \frac{-\beta}{1-\beta} \\ \frac{-(1+\beta)}{\beta} & \frac{1}{\beta(1-\beta)} & \frac{0}{1-\beta} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_2 \\ P_3 \\ P_4 \end{bmatrix}.
 \end{aligned}$$

Так же, как в предыдущем разделе, это можно записать в матричном виде

$$C(t) = [t^3 \quad t^2 \quad t \quad 1][A][G] - [T][A][G], \quad (5-44)$$

где

$$[A] = \begin{bmatrix} -(1-\alpha)^2 & \frac{(1-\alpha)+\alpha\beta}{\alpha} & \frac{-(1-\alpha)-\alpha\beta}{1-\beta} & \frac{\beta^2}{1-\beta} \\ \frac{\alpha}{2(1-\alpha)^2} & \frac{-2(1-\alpha)-\alpha\beta}{\alpha} & \frac{2(1-\alpha)-\beta(1-2\alpha)}{(1-\beta)} & \frac{-\beta^2}{1-\beta} \\ \frac{\alpha}{-(1-\alpha)^2} & \frac{(1-2\alpha)}{\alpha} & \alpha & 0 \\ \frac{\alpha}{0} & \frac{1}{1} & 0 & 0 \end{bmatrix} \quad (5-60)$$

и опять

$$[G]^T = [P_1 \quad P_2 \quad P_3 \quad P_4], \quad (5-53)$$

$$F_1(t) = \frac{(1-\alpha)^2}{\alpha}(-t^3 + 2t_2 - t), \quad (5-61a)$$

$$F_2(t) = \frac{[(1-\alpha)+\alpha\beta]}{\alpha}t^3 + \frac{[-2(1-\alpha)-\alpha\beta]}{\alpha}t^2 + \frac{(1-2\alpha)}{\alpha}t + 1, \quad (5-61b)$$

$$F_3(t) = \frac{[-(1-\alpha)-\alpha\beta]}{1-\beta}t^3 + \frac{[2(1-\alpha)-\beta(1-2\alpha)]}{\alpha}t^2 + \alpha t, \quad (5-61c)$$

$$F_4(t) = \frac{\beta^2}{1-\beta}(t^3 - t^2). \quad (5-61d)$$

На рис. 5-22 изображены интерполяционные функции $F_i(t)$ для частного случая $\beta = 1 - \alpha$, т. е. когда длина хорды от P_1 до P_2 равна длине от P_3 до P_4 : $|(P_2 - P_1)| = |(P_4 - P_3)|$.

Рассмотрим это на примере.

Пример 5-6 Обобщенная параболическая интерполяция

Снова рассмотрим данные из предыдущего примера (рис. 5-22). Найти параболическую интерполяцию между P_2 и P_3 в обобщенной формулировке. Вычислить промежуточные точки при $t = 1/3, 2/3$.

Найдем α и β из уравнений (5-54) и (5-55):

$$\begin{aligned} c_{21} &= |[P_2] - [P_1]| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \\ &= \sqrt{(1 - 0)^2 + (1 - 0)^2} = \sqrt{2}, \\ c_{32} &= |[P_3] - [P_2]| = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} = \\ &= \sqrt{(2 - 1)^2 + (-1 - 1)^2} = \sqrt{5}, \\ c_{43} &= |[P_4] - [P_3]| = \sqrt{(x_4 - x_3)^2 + (y_4 - y_3)^2} = \\ &= \sqrt{(3 - 2)^2 + (0 + 1)^2} = \sqrt{2}. \end{aligned}$$

Итак,

$$\alpha = \frac{c_{21}}{c_{32} + c_{21}} = \frac{\sqrt{2}}{\sqrt{5} + \sqrt{2}} = 0.387,$$

$$\beta = \frac{c_{32}}{c_{43} + c_{32}} = \frac{\sqrt{5}}{\sqrt{2} + \sqrt{5}} = 0.613.$$

Заметим, что $\beta = 1 - \alpha$.

Из уравнения (5-60)

$$[A] = \begin{bmatrix} -0.969 & 2.194 & -2.194 & 0.969 \\ 1.937 & -3.775 & 2.806 & -0.969 \\ -0.969 & 0.581 & 0.387 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

и из уравнения (5-44)

$$C\left(\frac{1}{3}\right) = \begin{bmatrix} \frac{1}{27} & \frac{1}{9} & \frac{1}{3} & 1 \end{bmatrix} [A] \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 2 & -1 \\ 3 & 0 \end{bmatrix} = [1.360 \quad 0.496].$$

Аналогично, при $t = 2/3$, $C(2/3) = [1.640 \quad -0.496]$. На рис. 5-23 изображена получившаяся кривая, а также кривые из примера 5-5. Заметен большой изгиб около заданных точек.

На рис. 5-24 показана локальная коррекция параболически интерполированных кривых. Здесь имеется 11 заданных точек, или радиус-векторов, и 9 параболических составных сегментов. Центральная точка располагается в трех положениях. Заметим, что влияние перемещения этой точки на форму кривой ограничивается ± 2 -я сегментами.

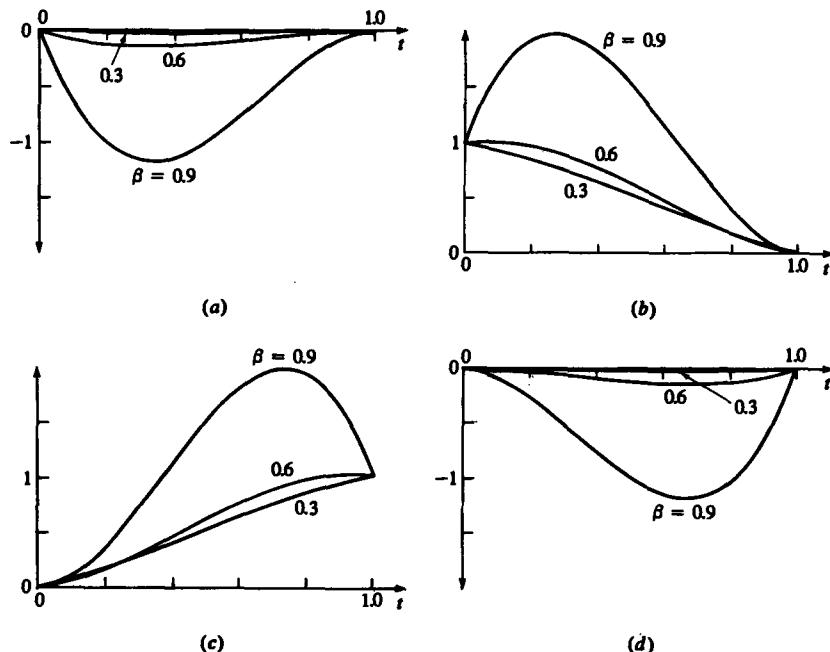


Рис. 5-22 Обобщенные параболические весовые функции, $\alpha = 1 - \beta$. (а) F_1 ; (б) F_2 ; (в) F_3 ; (г) F_4 .

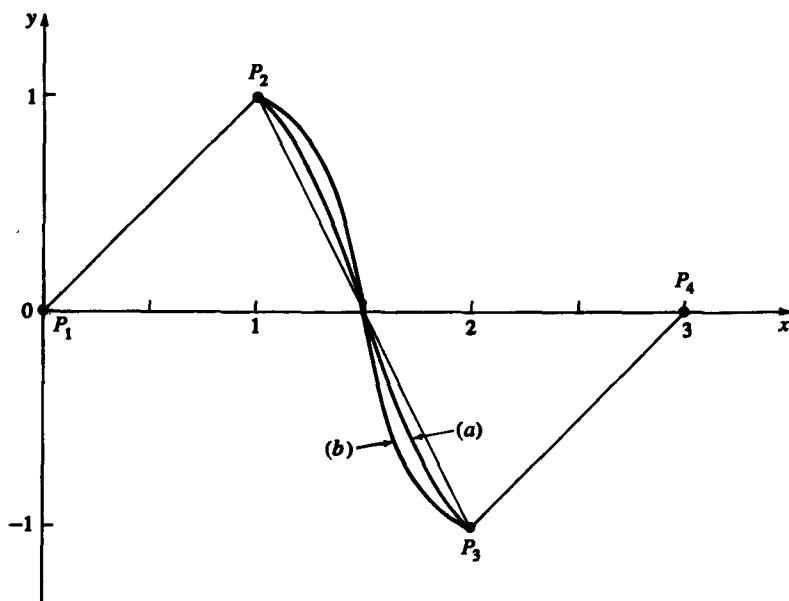


Рис. 5-23 Сравнение результатов параболической интерполяции. (а) Пример 5-5, $\alpha = 1/2$; (б) Пример 5-6, $\alpha = 1 - \beta$.

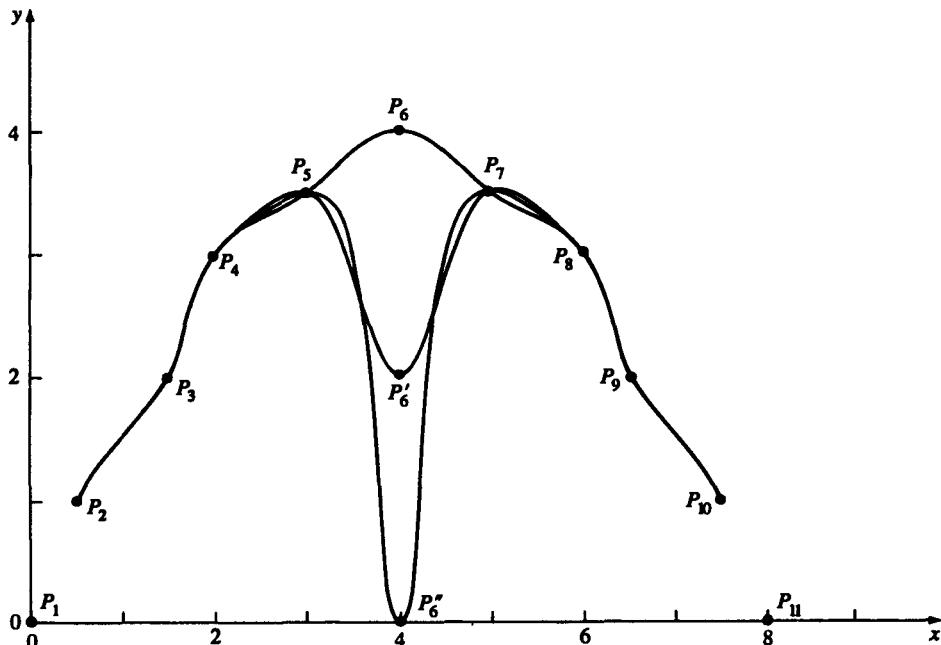


Рис. 5-24 Локальная коррекция параболически интерполированных кривых.

5-8 КРИВЫЕ БЕЗЬЕ

До сих пор мы обсуждали, как провести кривую через заданное множество точек. Рассмотренные методы во многих случаях дают прекрасные результаты и особенно удобны при описании формы, основа которой получена с помощью экспериментов или математических расчетов. Это, например, крыло самолета, составные части мотора, механические и структурные детали. Существует, однако, и другой класс задач, когда решение зависит как от функциональных, так и от эстетических требований, например дизайн поверхности машины, фюзеляжа самолета, формы корабля, мебели или посуды. Кроме количественных критериев здесь требуется учет практического опыта, и часто необходимо интерактивное вмешательство разработчика.

Рассмотренные выше методы, в частности кубические сплайны, неудобны для интерактивной работы. Направление и величина касательных не дают необходимого интуитивного представления о кривой, так как неочевидна связь между набором чисел и формой соответствующей кривой.

Пьер Безье предложил другой метод создания кривых и поверхностей любой формы. Безье вывел математическую основу своего метода из геометрических соображений [5-11] — [5-13], но в работах [5-14] и [5-15] показано, что его результат эквивалентен базису Бернштейна или функции полиномиальной аппроксимации.

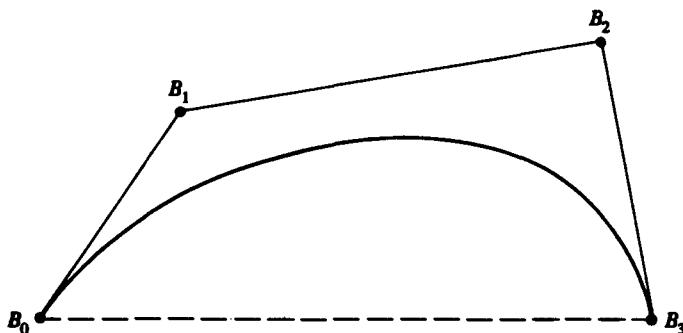


Рис. 5-25 Кривая Безье и определяющие ее точки.

Кривая Безье задается многоугольником, как показано на рис. 5-25. Так как базис Безье является бернштейновским, сразу же известны некоторые свойства кривых Безье. Например:

Функции базиса вещественны.

Степень многочлена, определяющего участок кривой, на единицу меньше количества точек соответствующего многоугольника.

Основа формы кривой повторяет очертания многоугольника.

Первая и последняя точки кривой совпадают с соответствующими точками определяющего многоугольника.

Векторы касательных в концах кривой по направлению совпадают с первой и последней сторонами многоугольника.

Кривая лежит внутри выпуклой оболочки многоугольника, т. е. внутри самого большого многоугольника, построенного по заданным точкам. На рис. 5-25 выпуклая оболочка обозначена штриховой и тонкой линиями.

Кривая обладает свойством уменьшения вариации. Это означает, что кривая пересекает любую прямую линию не чаще, чем определяющий многоугольник.

Кривая инвариантна относительно аффинных преобразований.

На рис. 5-26 показано несколько четырехточечных многоугольников Безье и соответствующих кривых. На основе перечисленных выше свойств можно легко научиться предсказывать форму кривой по виду многоугольника.

Математическое параметрическое представление кривой Безье имеет вид

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t), \quad 0 \leq t \leq 1, \quad (5-62)$$

где базис Безье или Бернштейна, или функция аппроксимации

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad (5-63)$$

с

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (5-64)$$

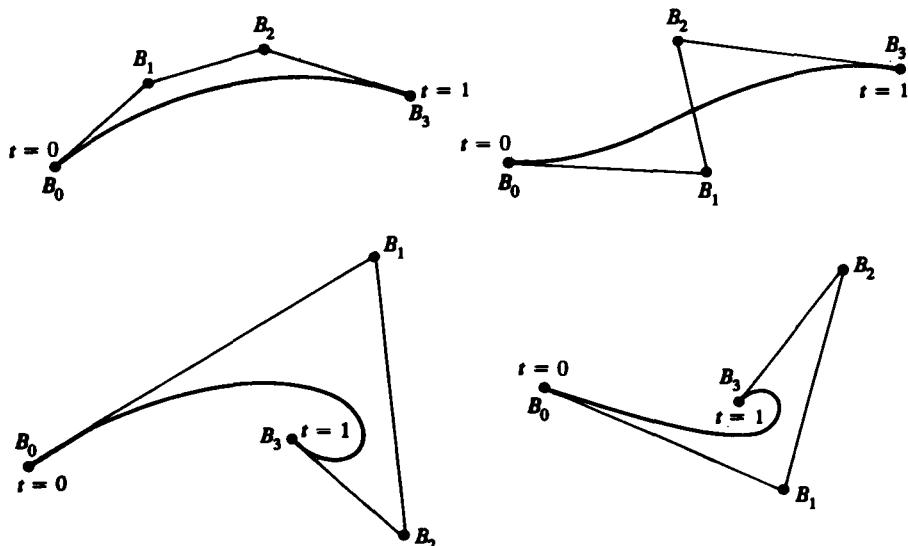


Рис. 5-26 Многоугольники Безье для кубических кривых.

$J_{n,i}(t)$ — это i -я функция базиса Бернштейна порядка n . Здесь n — порядок определяющей функции базиса Бернштейна — и, следовательно, сегмента полиномиальной кривой, на единицу меньше количества точек определяющего многоугольника. Как показано на рис. 5-25, вершины многоугольника Безье нумеруются от 0 до n^1 . Поэтому $(0)^0 \equiv 1$ и $0! \equiv 1$.

На рис. 5-27 изображены аппроксимирующие функции для разных значений n . Заметим, что функции симметричны. Каждая функция имеет порядок n , например все четыре функции на рис. 5-27б для $n = 3$ кубические, Максимум каждой функции достигается при $t = i/n$ и равен (5-14)

$$J_{n,i}\left(\frac{i}{n}\right) = \binom{n}{i} \frac{i^i(n-i)^{n-i}}{n^n}. \quad (5-65)$$

¹ Обозначения и отсчет выбраны в соответствии с традициями, принятыми в литературе по кривым Безье и базису Бернштейна. Для программирования удобнее другое представление

$$P(t) = \sum_{I=1}^N B_I J_{N,I}(t),$$

где

$$J_{N,I} = \binom{N-1}{I-1} t^{I-1} (1-t)^{N-I}$$

и

$$\binom{N-1}{I-1} = \frac{(N-1)!}{(I-1)!(N-I)!}.$$

Точки многоугольника нумеруются от 1 до N . Для перехода к этим обозначениям положим $n = N - 1$ и $i = I - 1$.

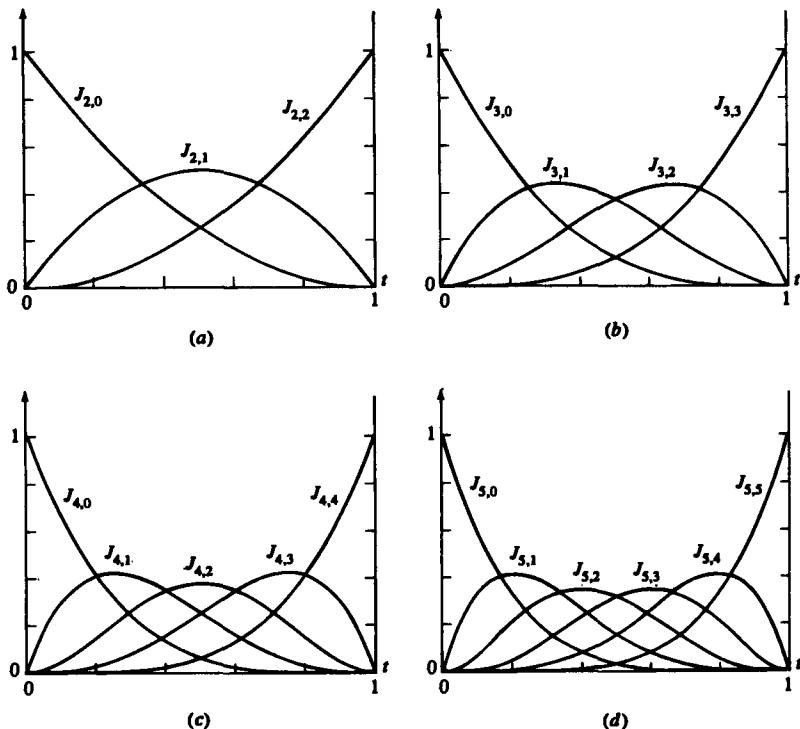


Рис. 5-27 Весовые функции Безье/Бернштейна. (a) Многоугольник из трех точек, $n = 2$; (b) из четырех точек, $n = 3$; (c) из пяти точек, $n = 4$; (d) из шести точек, $n = 5$.

Например, для кубической кривой $n = 3$. Максимум $J_{3,1}$ и $J_{3,2}$ достигается в $1/3$ и $2/3$, соответственно, и имеет значения

$$J_{3,1}\left(\frac{1}{3}\right) = \frac{4}{9} \quad \text{и} \quad J_{3,2}\left(\frac{2}{3}\right) = \frac{4}{9}.$$

Рисунок 5-27b иллюстрирует этот пример.

Рассмотрим уравнения (5-62) и (5-64) для первой точки на кривой, т. е. при $t = 0$

$$J_{n,0}(0) = \frac{n!(1)(1-0)^{n-0}}{n!} = 1, \quad i = 0,$$

и

$$J_{n,i}(0) = \frac{n!(0)^i(1-0)^{n-i}}{i!(n-i)!} = 0, \quad i \neq 0.$$

Итак,

$$P(0) = B_0 J_{n,0}(0) = B_0,$$

первая точка кривой совпадает с первой точкой многоугольника.

Аналогично, для последней точки кривой, т. е. при $t = 1$

$$\begin{aligned} J_{n,n}(1) &= \frac{n!(1)^n(0)^{n-n}}{n!(1)} = 1, & i = n, \\ J_{n,i}(1) &= \frac{n!}{i!(n-i)!} t^i (1-1)^{n-i} = 0, & i \neq n. \end{aligned}$$

Итак,

$$P(1) = B_n J_{n,n}(1) = B_n$$

и последняя точка на кривой Безье совпадает с последней точкой определяющего многоугольника.

Далее, можно показать, что для любого значения параметра t сумма базисных функций равна 1, т. е.

$$\sum_{i=0}^n J_{n,i}(t) = 1. \quad (5-66)$$

Рассмотрим метод построения Безье на примере.

Пример 5-7 Кривая Безье

Пусть заданы вершины многоугольника Безье $B_0[1 \ 1]$, $B_1[2 \ 3]$, $B_2[4 \ 3]$ и $B_3[3 \ 1]$. Найти семь точек, лежащих на кривой Безье.

Рассмотрим уравнения (5-62) – (5-64):

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t),$$

где

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

и

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}.$$

В нашем случае $n = 3$, так как имеется четыре вершины. Отсюда

$$\binom{n}{i} = \binom{3}{i} = \frac{6}{i!(3-1)!}$$

и

$$\begin{aligned} J_{3,0}(t) &= (1)t^0(1-t)^3 = (1-t)^3, \\ J_{3,1}(t) &= 3t(1-t)^2, \\ J_{3,2}(t) &= 3t^2(1-t), \\ J_{3,3}(t) &= t^3. \end{aligned}$$

Итак,

$$\begin{aligned} P(t) &= B_0 J_{3,0} + B_1 J_{3,1} + B_2 J_{3,2} - B_3 J_{3,3} \\ &= (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t) P_2 + t^3 P_3. \end{aligned}$$

Таблица 5-4 Коэффициенты для кривой Безье

t	$J_{3,0}$	$J_{3,1}$	$J_{3,2}$	$J_{3,3}$
0	1	0	0	0
0.15	0.614	0.325	0.058	0.003
0.35	0.275	0.444	0.239	0.042
0.5	0.125	0.375	0.375	0.125
0.65	0.042	0.239	0.444	0.275
0.85	0.003	0.058	0.325	0.614
1	0	0	0	1

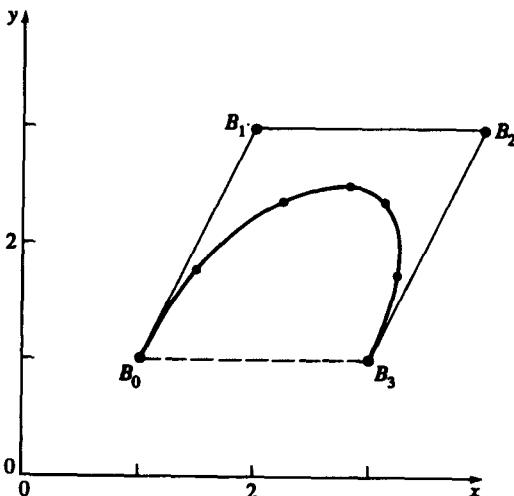


Рис. 5-28 Сегмент кривой Безье, пример 5-7.

Значения $J_{n,i}$ для различных значений t приведены в табл. 5-4.
Точки на кривой:

$$\begin{aligned}
 P(0) &= B_0 = [1 \quad 1], \\
 P(0.15) &= 0.614B_0 + 0.325B_1 + 0.058B_2 + 0.003B_3 = [1.5 \quad 1.765], \\
 P(0.35) &= 0.275B_0 + 0.444B_1 + 0.239B_2 + 0.042B_3 = [2.248 \quad 2.367], \\
 P(0.5) &= 0.125B_0 + 0.375B_1 + 0.375B_2 + 0.125B_3 = [2.75 \quad 2.5], \\
 P(0.65) &= 0.042B_0 + 0.239B_1 + 0.444B_2 + 0.275B_3 = [3.122 \quad 2.367], \\
 P(0.85) &= 0.003B_0 + 0.058B_1 + 0.325B_2 + 0.614B_3 = [3.248 \quad 1.765], \\
 P(1) &= B_3 = [3 \quad 1].
 \end{aligned}$$

Эти точки показаны на определяющем многоугольнике на рис. 5-28.

Уравнение кривой Безье можно записать в матричном виде, так же как уравнения для кубических сплайнов и параболической интерполяции (см. уравнения

5-27 и 5-44):

$$P(t) = [T][N][G] = [F][G]. \quad (5-67)$$

Здесь $[F] = [J_{n,0} \ J_{n,1} \ \cdots \ J_{n,n}]$ и $[G]^T = [B_0 \ B_1 \ \cdots \ B_n]$.

Особенный интерес представляют матричные формы для малых значений n . Для многоугольника из четырех точек ($n = 3$) кривая Безье имеет вид

$$P(t) = [(1-t)^3 \ 3t(1-t)^2 \ 3t^2(1-t) \ t^3] \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix}.$$

Группируя коэффициенты, получим

$$P(t) = [T][N][G] = [t^3 \ t^2 \ t \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix}. \quad (5-68)$$

Аналогично, кривая Безье четвертого порядка ($n = 4$), заданная многоугольником из пяти точек:

$$P(t) = [t_4 \ t_3 \ t_2 \ t \ 1] \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 12 & -12 & 4 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}. \quad (5-69)$$

В работе [5-19] приводится обобщенное представление:

$$P(t) = [T][N][G],$$

где

$$[T] = [t^n \ t^{n-1} \ \cdots \ t \ 1],$$

$$[N] = \begin{bmatrix} \binom{n}{0} \binom{n}{n} (-1)^n & \binom{n}{1} \binom{n-1}{n-1} (-1)^{n-1} & \cdots & \binom{n}{n} \binom{n-n}{n-n} (-1)^0 \\ \binom{n}{0} \binom{n}{n-1} (-1)^{n-1} & \binom{n}{1} \binom{n-1}{n-2} (-1)^{n-2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \binom{n}{0} \binom{n}{1} (-1)^1 & \binom{n}{1} \binom{n-1}{0} (-1)^0 & \cdots & 0 \\ \binom{n}{0} \binom{n}{0} (-1)^0 & 0 & \cdots & 0 \end{bmatrix}. \quad (5-70)$$

Матрица $[G]^T$ — это опять $[B_0 \ B_1 \ \cdots \ B_n]$. Отдельные члены матрицы $[N]$ таковы:

$$(N_{i+1,j+1})_{i,j=0}^n = \begin{cases} \binom{n}{j} \binom{n-j}{n-i-j} (-1)^{n-i-j}, & 0 \leq i + j \leq n, \\ 0 & \text{иначе} \end{cases}.$$

Уравнение (5-70) можно записать в более удобном виде

$$[N] = [C][D], \quad (5-71)$$

где

$$[C] = \begin{bmatrix} \binom{n}{n}(-1)^n & \binom{n}{1}\binom{n-1}{n-1}(-1)^{n-1} & \cdots & \binom{n}{n}\binom{n-n}{n-n}(-1)^0 \\ \binom{n}{n-1}(-1)^{n-1} & \binom{n}{1}\binom{n-1}{n-2}(-1)^{n-2} & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \binom{n}{1}(-1)^1 & \binom{n}{1}\binom{n-1}{0}(-1)^0 & & 0 \\ \binom{n}{0}(-1)^0 & 0 & \cdots & 0 \end{bmatrix},$$

$$[D] = \begin{bmatrix} \binom{n}{0} & \cdots & 0 \\ \vdots & \binom{n}{0} & \vdots \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \binom{n}{n} \end{bmatrix}.$$

Уравнения (5-70) или (5-71) удобнее для расчета при больших значениях n . Заметим, что для всех n матрица $[N]$ симметрична относительно главной диагонали и правый нижний угол состоит из нулей.

Для каждой отдельной кривой Безье необходимо знать касательные векторы в ее концах, но если необходимо соблюдать непрерывность кривизны и нахлона в точках соединения кривых, рассчитывать нормали к поверхности для освещения, вычислять локальную кривизну, то требуется знать как первую, так и вторую производную кривой Безье.

Из уравнения (5-62) первая производная кривой Безье имеет вид:

$$P'(t) = \sum_{i=0}^n B_i J'_{n,i}(t). \quad (5-72)$$

Вторая производная такова:

$$P''(t) = \sum_{i=0}^n B_i J''_{n,i}(t). \quad (5-73)$$

Формально дифференцируя уравнение (5-63), получаем производные базисных функций

$$\begin{aligned} J'_{n,i}(t) &= \binom{n}{i} \{ i t^{i-1} (1-t)^{n-i} - (n-i) t^i (1-t)^{n-i-1} \} \\ &= \binom{n}{i} t^i (1-t)^{n-i} \left\{ \frac{i}{t} - \frac{(n-i)}{(1-t)} \right\} \\ &= \frac{(i-nt)}{t(1-t)} J_{n,i}(t). \end{aligned} \quad (5-74)$$

Аналогично, вторые производные имеют вид:

$$J''_{n,i}(t) = \left\{ \frac{(i - nt)^2 - nt^2 - i(1 - 2t)}{t^2(1-t)^2} \right\} J_{n,i}(t). \quad (5-75)$$

В начале и конце кривой Безье, т. е. при $t = 0$ и $t = 1$, численный расчет уравнений (5-74) и (5-75) представляет затруднения¹.

Другой способ вычисления r -й производной при $t = 0$:

$$P^r(0) = \frac{n!}{(n-r)!} \sum_{i=0}^r (-1)^{r-i} \binom{r}{i} B_i \quad (5-76)$$

и при $t = 1$:

$$P^r(1) = \frac{n!}{(n-r)!} \sum_{i=0}^r (-1)^i \binom{r}{i} B_{n-i}. \quad (5-77)$$

Отсюда первые производные в концах будут

$$P'(0) = n(B_1 - B_0) \quad (5-78)$$

и

$$P'(1) = n(B_n - B_{n-1}). \quad (5-79)$$

Это показывает, что касательные к кривой Безье в первой и последней точках параллельны соответствующим сторонам многоугольника.

Аналогично, вторые производные в концах таковы:

$$P''(0) = n(n-1)(B_0 - 2B_1 + B_2) \quad (5-80a)$$

и

$$P''(1) = n(n-1)(B_n - 2B_{n-1} + B_{n-2}). \quad (5-80b)$$

Вторые производные в концах зависят от двух ближайших сторон, т. е. от трех ближайших вершин. В общем случае, r -я производная в начальной и конечной точках зависит от этих точек и r ближайших вершин многоугольника.

Подробнее рассмотрим это на примере.

Пример 5-8 Производные кривых Безье

Рассмотрим четырехточечный многоугольник Безье, например, как на рис. 5-26 и 5-28. Вспомним представление кривой

$$P(t) = B_0 J_{3,0}(t) + B_1 J_{3,1}(t) + B_2 J_{3,2}(t) + B_3 J_{3,3}(t).$$

Отсюда первая производная

$$P'(t) = B_0 J'_{3,0}(t) + B_1 J'_{3,1}(t) + B_2 J'_{3,2}(t) + B_3 J'_{3,3}(t).$$

¹ Правильный результат получается при алгебраическом вычислении и подстановке до численного расчета (см. пример 5-8).

Вспомним пример 5-7 и непосредственно продифференцируем базисные функции

$$\begin{aligned} J_{3,0}(t) &= (1-t)^3 & \Rightarrow & J'_{3,0}(t) = -3(1-t)^2, \\ J_{3,1}(t) &= 3t(1-t)^2 & \Rightarrow & J'_{3,1}(t) = 3(1-t)^2 - 6t(1-t), \\ J_{3,2}(t) &= 3t^2(1-t) & \Rightarrow & J'_{3,2}(t) = 6t(1-t) - 3t^2, \\ J_{3,3}(t) &= t^3 & \Rightarrow & J'_{3,3}(t) = 3t^2 \end{aligned}$$

Подставим $t = 0$:

$$J'_{3,3}(0) = -3, \quad J'_{3,1}(0) = 3, \quad J'_{3,2}(0) = 0, \quad J'_{3,3}(0) = 0.$$

Подстановка дает

$$P'(0) = -3P_0 + 3P_1 = 3(P_1 - P_0).$$

Поэтому направление касательной в начале кривой совпадает с первой стороной многоугольника (см. рис. 5-28).

В конце кривой $t = 1$ и

$$J'_{3,0}(1) = 0, \quad J'_{3,1}(1) = 0, \quad J'_{3,2}(1) = -3, \quad J'_{3,3}(1) = 3.$$

Аналогично, подстановка дает

$$P'(1) = -3P_2 + 3P_3 = 3(P_3 - P_2)$$

и направление касательного вектора в конце кривой совпадает с последней стороной многоугольника.

Чтобы вычислить производные вдоль кривой, воспользуемся функциями базиса и уравнениями (5-74) и (5-75):

$$\begin{aligned} J'_{3,0}(t) &= \frac{(0-3t)}{t(1-t)}(1-t)^3 = -3(1-t)^2, \\ J'_{3,1}(t) &= \frac{(1-3t)}{t(1-t)}(3t)(1-t)^2 = 3(1-3t)(1-t) = 3(1-4t+3t^2), \\ J'_{3,2}(t) &= \frac{(2-3t)}{t(1-t)}(3t^2)(1-t) = 3t(2-3t), \\ J'_{3,3}(t) &= \frac{3(1-t)}{t(1-t)}t^3 = 3t^2. \end{aligned}$$

Результаты легко вычисляются как для $t = 0$, так и для $t = 1$. Подставляя в уравнение (5-72), получаем первую производную в любой точке кривой. Например, при $t = 1/2$ имеем

$$\begin{aligned} P'\left(\frac{1}{2}\right) &= -3\left(1-\frac{1}{2}\right)^2 B_0 + 3\left(1-\frac{3}{2}\right)\left(1-\frac{1}{2}\right)B_1 + \left(\frac{3}{2}\right)\left(2-\frac{3}{2}\right)B_2 + \frac{3}{4}B_3 \\ &= -\frac{3}{4}B_0 - \frac{3}{4}B_1 + \frac{3}{4}B_2 + \frac{3}{4}B_3 = -\frac{3}{4}(B_0 + B_1 - B_2 - B_3). \end{aligned}$$

Результат для точек B_0, B_1, B_2, B_3 из примера 5-7 изображен на рис. 5-29.

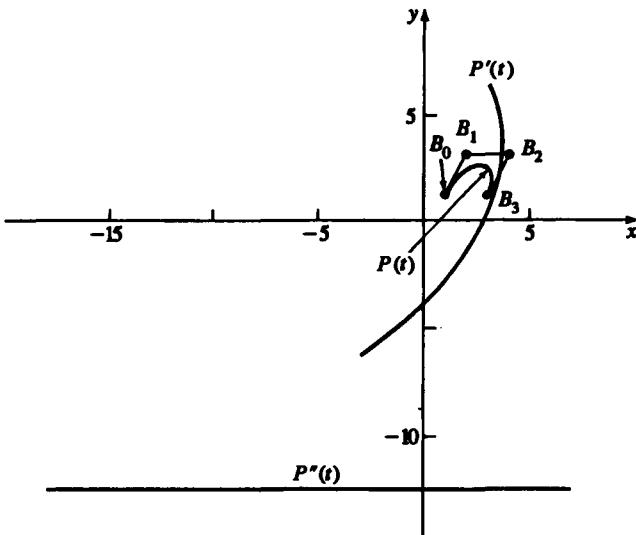


Рис. 5-29 Кривая Безье и ее производные: $P(t)$; $P'(t)$; $P''(t)$.

Аналогично, вторые производные имеют вид:

$$J''_{3,0}(t) = \frac{\{(-3t)^2 - 3t^2\}}{t^2(1-t)^2} (1-t)^3 = 6(1-t),$$

$$J''_{3,1}(t) = \frac{\{(1-3t)^2 - 3t^2 - (1-2t)\}}{t^2(1-t)^2} (3t)(1-t)^2 = -6(2-3t),$$

$$J''_{3,2}(t) = \frac{\{(2-3t)^2 - 3t^2 - 2(1-2t)\}}{t^2(1-t)^2} (3t^2)(1-t) = 6(1-3t),$$

$$J''_{3,3}(t) = \frac{\{(3-3t)^2 - 3t^2 - 3(1-2t)\}}{t^2(1-t)^2} t^3 = 6t.$$

Уравнение (5-73) при $t = 1/2$ дает

$$\begin{aligned} P''\left(\frac{1}{2}\right) &= 6\left(1-\frac{1}{2}\right)B_0 - 6\left(2-\frac{3}{2}\right)B_1 + 6\left(1-\frac{3}{2}\right)B_2 + 3B_3 = \\ &= 3B_0 - 3B_1 - 3B_2 + 3B_3 = 3(B_0 - B_1 - B_2 + B_3). \end{aligned}$$

Иллюстрация также приведена на рис. 5-29. Заметим, что вектор из начала координат в любую точку на каждой из кривых представляет соответственно направление и величину радиуса-вектора и приблизительную кривизну в этой точке на кривой.

Условие непрерывности соседних кривых Безье формулируется очень просто. Пусть кривая Безье $P(t)$ степени n задана вершинами B_i , а соседняя кривая Безье $Q(s)$ степени m — вершинами C_i . Тогда непрерывность первой производной в точке соединения выражается соотношением

$$P'(1) = gQ'(0),$$

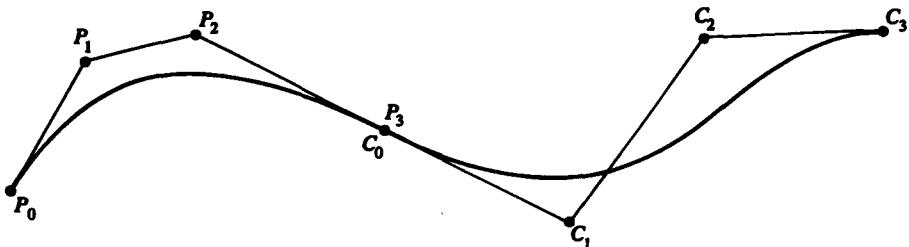


Рис. 5-30 Непрерывность первой производной для кубических кривых Безье.

где g — скаляр. Пользуясь уравнениями (5-78) и (5-79), получим

$$C_1 - C_0 = \frac{n}{m} (B_n - B_{n-1}).$$

Из непрерывности кривой следует, что $C_0 = B_n$ и

$$C_1 = \frac{n}{m} (B_n - B_{n-1}) + B_n.$$

Отсюда направления касательных на стыке совпадают, если три вершины B_{n-1} , $B_n = C_0$, C_1 коллинеарны, т. е. B_n должна лежать на линии между B_{n-1} и C_1 .

Если совпадают еще и величины касательных векторов, то $B_n = C_0$ является серединой отрезка от B_{n-1} до C_1 :

$$C_1 - C_0 = B_n - B_{n-1} = C_0 - B_{n-1}$$

или

$$C_1 + B_{n-1} = 2C_0 = 2B_n.$$

На рис. 5-30 приведена иллюстрация для $n = m = 3$, т. е. для двух кубических кривых Безье.

Условие непрерывности вторых производных между соседними кривыми:

$$m(m-1)(C_0 - 2C_1 + C_2) = n(n-1)(B_{n-2} - 2B_{n-1} + B_n).$$

Из условий C^0 и C^1 непрерывности в точке соединения получаем положение третьей определяющей вершины второго многоугольника, выраженное через три последние вершины первого многоугольника, а именно

$$C_2 = \frac{n(n-1)}{m(m-1)} B_{n-2} - 2 \left\{ \frac{n(n-1)}{m(m-1)} + \frac{n}{m} \right\} B_{n-1} + \left\{ 1 + 2 \frac{n}{m} + \frac{n(n-1)}{m(m-1)} \right\} B_n$$

Это означает, что вершины B_{n-2} , B_{n-1} , $B_n = C_0$, C_1 , C_2 должны либо составлять выпуклый многоугольник, либо лежать на одной прямой, чтобы сохранить C^2

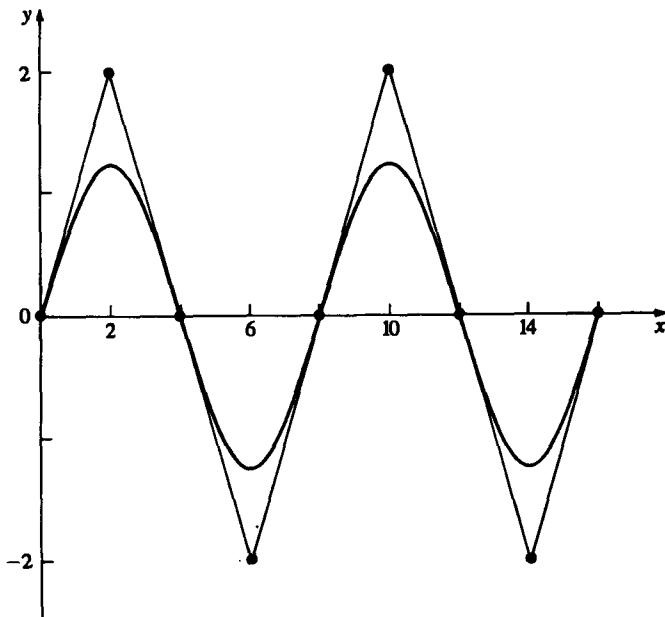


Рис. 5-31 Непрерывность второй производной для кривых Безье четвертой степени.

непрерывность на стыке. Для кубических кривых Безье ($n = m = 3$) это условие имеет вид

$$C_2 = B_{n-1} - 4(B_{n-1} - B_n).$$

Несколько карандашных набросков на бумаге покажут, что данное требование существенно ограничивает множество кривых; поэтому на практике для соблюдения непрерывности вторых производных используются полиномиальные кривые более высокого порядка. На рис. 5-31 приведен пример непрерывности вторых производных для двух пятиточечных кривых Безье.

Если необходима более гибкая кривая Безье, увеличивается количество определяющих точек и степень полинома. Для каждой точки на кривой Безье с n определяющими вершинами многоугольника B_0, \dots, B_n та же самая точка на новой кривой Безье с $n + 1$ определяющими вершинами B_0^*, \dots, B_{n+1}^* задается в виде

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t) = \sum_{i=0}^{n+1} B_i^* J_{n+1,i}(t), \quad (5-81)$$

где

$$B_0^* = B_0,$$

$$B_i^* = \alpha_i B_{i-1} + (1 - \alpha_i) B_i, \quad \alpha_i = \frac{i}{n+1}, \quad i = 1, \dots, n,$$

$$B_{n+1}^* = B_n.$$

Здесь можно с успехом применить методику из работы [5-14]. В предельном случае многоугольник сходится к кривой. Дополнительную гибкость кривой можно также получить разбиением кривой Безье на две новых таким образом, чтобы они вместе совпадали с исходной кривой. В работах Барского [5-16, 5-18] было показано, что любая кривая Безье может быть разбита с использованием произвольного параметра в диапазоне $0 \leq t \leq 1$. Простейший случай — это средняя точка, т. е. $t = 1/2$ (см. [5-18]). При разбиении средней точкой получаются два специальных вида кубических кривых Безье.

Кубическая кривая Безье (см. упражнение 5-7) задается в виде

$$P(t) = (1-t)^3 B_0 + 3t(1-t)^2 B_1 + 3t^2(1-t) B_2 + t^3 B_3, \quad 0 \leq t \leq 1.$$

с определяющими вершинами B_0, B_1, B_2, B_3 . Тогда многоугольник C_0, C_1, C_2, C_3 определяет кривую Безье $Q(u)$, $0 \leq u \leq 1$, соответствующую первой половине исходной кривой, а именно $P(t)$, $0 \leq t \leq 1/2$, как и требовалось. Подобным образом, многоугольник D_0, D_1, D_2, D_3 определяет кривую Безье $R(v)$, $0 \leq v \leq 1$, соответствующую второй половине исходной кривой, т. е. $P(t)$, $1/2 \leq t \leq 1$, как и требовалось. Новые определяющие вершины многоугольника C_i и D_i получаются путем приравнивания радиус-векторов и касательных векторов при $u = 0$, $t = 0$; $u = 1$, $t = 1/2$ и $v = 0$, $t = 1/2$; $v = 1$, $t = 1$. Из соотношений (5-62) и (5-72) получаем

$$\begin{aligned} C_0 &= B_0, \\ 3(C_1 - C_0) &= \frac{3}{2}(B_1 - B_0), \\ 3(C_3 - C_2) &= \frac{3}{8}(B_3 + B_2 - B_1 - B_0), \\ C_3 &= \frac{1}{8}(B_3 + 3B_2 + 3B_1 + B_0). \end{aligned}$$

Решение этих уравнений дает

$$\begin{aligned} C_0 &= B_0, \\ C_1 &= \frac{1}{2}(B_1 + B_0), \\ C_2 &= \frac{1}{4}(B_2 + 2B_1 + B_0), \\ C_3 &= \frac{1}{8}(B_3 + 3B_2 + 3B_1 + B_0). \end{aligned}$$

Аналогично,

$$\begin{aligned} D_0 &= \frac{1}{8}(B_3 + 3B_2 + 3B_1 + B_0), \\ D_1 &= \frac{1}{4}(B_3 + 2B_2 + B_1), \\ D_2 &= \frac{1}{2}(B_3 + B_2), \\ D_3 &= B_3. \end{aligned}$$

Эти результаты обобщаются до

$$C_i = \sum_{j=0}^i \binom{i}{j} \frac{B_j}{2^i}, \quad i = 0, 1, \dots, n, \quad (5-82a)$$

$$D_i = \sum_{j=1}^n \binom{n-i}{n-j} \frac{B_j}{2^{n-i}}, \quad i = 0, 1, \dots, n. \quad (5-82b)$$

Дальнейшее применение определяющих многоугольников с помощью кривой Безье оставляем для самостоятельной работы.

5-9 В-СПЛАЙНЫ

С математической точки зрения кривая, заданная вершинами многоугольника, зависит от интерполяции или аппроксимации, устанавливающей связь кривой и многоугольника. Здесь основой является выбор базисных функций. Как было отмечено в разд. 5-8, базис Бернштейна порождает кривые Безье вида (5-62), но он обладает двумя свойствами, которые ограничивают гибкость кривых. Во-первых, количество вершин многоугольника жестко задает порядок многочлена. Например, кубическая кривая должна быть задана четырьмя вершинами и тремя отрезками. Многоугольник из шести точек всегда порождает кривую пятого порядка. Единственный способ понизить степень кривой — это сократить количество вершин, а повысить степень кривой — увеличить их число.

Второе ограничение следует из глобальной природы базиса Бернштейна. Это означает, что величина аппроксимирующих функций $J_{n,i}(t)$ из уравнения (5-63) ненулевая для всех значений параметра на кривой. Любая точка на кривой Безье зависит от всех определяющих вершин, поэтому изменение какой-либо одной вершины оказывает влияние на всю кривую. Локальные воздействия на кривую невозможны.

Например, так как наклон концов кривой Безье задан соответствующими сторонами многоугольника, можно передвинуть среднюю вершину пятиточечного многоугольника, не меняя направления на концах. Однако из-за глобальности базиса Бернштейна меняется форма всей кривой. Отсутствие локальной коррекции может оказаться решающим в некоторых прикладных задачах.

Существует неглобальный базис, называемый базисом В-сплайна, включающий базис Бернштейна как частный случай. В-сплайны неглобальны, так как с каждой вершиной B_i связана своя базисная функция. Поэтому влияние каждой вершины на кривую проявляется только при тех значениях параметра, где соответствующая базисная функция не равна нулю. Базис В-сплайна также позволяет менять порядок базисных функций и, следовательно, всей кривой без изменения количества вершин. Теория В-сплайнов была предложена в работе [5-20]. Рекурсивное определение для численного решения было выведено независимо Коксом [5-21] и де Буром [5-22]. Гордон и Ризенфельд [5-15], [5-23] определяли кривые через базис В-сплайна.

Пусть $P(t)$ определяет кривую как функцию от параметра t , тогда В-сплайн имеет вид

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t), \quad t_{\min} \leq t < t_{\max}, \quad 2 \leq k \leq n+1, \quad (5-83)$$

где B_i есть $n+1$ вершина многоугольника, а $N_{i,k}$ — нормализованные функции базиса В-сплайна.

Для i -й нормализованной функции базиса порядка k (степени $k-1$) функции базиса $N_{i,k}(t)$ определяются рекурсивными формулами Кокса—де Бура:

$$N_{i,1}(t) = \begin{cases} 1 & \text{если } x_i \leq t < x_{i+1} \\ 0 & \text{иначе} \end{cases} \quad (5-84a)$$

и

$$N_{i,k}(t) = \frac{(t - x_i)N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}. \quad (5-84b)$$

Величины x_i — это элементы узлового вектора, удовлетворяющие отношению $x_i \leq x_{i+1}$. Параметр t изменяется от t_{\min} до t_{\max} вдоль кривой $P(t)$ ¹. Считается, что $0/0 = 0$.

Формально В-сплайн определяется как полиномиальный сплайн порядка k (степени $k-1$), так как он удовлетворяет следующим условиям:

Функция $P(t)$ является полиномом степени $k-1$ на каждом интервале $x_i \leq t < x_{i+1}$.

$P(t)$ и ее производные порядка $1, 2, \dots, k-2$ непрерывны вдоль всей кривой.

Так, например, В-сплайн четвертого порядка — это кусочная кубическая кривая.

Из того что В-сплайн задается базисом В-сплайна, сразу следует еще несколько его свойств:

Сумма базисных функций В-сплайна для любого значения параметра t (см. работы [5-15] и [5-22])

$$\sum_{i=1}^{n+1} N_{i,k}(t) \equiv 1. \quad (5-85)$$

Каждая базисная функция положительна или равна нулю для всех значений параметра, т. е. $N_{i,k} \geq 0$.

Кроме $k=1$, все базисные функции имеют ровно один максимум.

Максимальный порядок кривой равен количеству вершин определяющего многоугольника.

Кривая обладает свойством уменьшения вариации. Кривая пересекает любую прямую не чаще, чем ее определяющий многоугольник.

¹ Отметим, что в отличие от кривых Безье вершины определяющего многоугольника нумеруются от 1 до $n+1$.

Общая форма кривой повторяет форму определяющего многоугольника.

Чтобы применить к кривой любое аффинное преобразование, необходимо применить его к вершинам определяющего многоугольника.

Кривая лежит внутри выпуклой оболочки определяющего многоугольника.

Последнее свойство В-сплайна сильнее, чем у кривых Безье. У В-сплайна порядка k (степени $k - 1$) точки кривой лежат внутри выпуклой оболочки k соседних точек. Таким образом, все точки на В-сплайне должны лежать внутри объединения всех выпуклых оболочек k последовательных вершин. На рис. 5-32 приводится иллюстрация для различных значений k , причем выпуклые оболочки выделены серым цветом. В частности, при $k = 2$ выпуклая оболочка совпадает с многоугольником, т. е. В-сплайн — это сам многоугольник.

С помощью свойства выпуклой оболочки легко показать, что если все точки многоугольника коллинеарны, то соответствующий В-сплайн — прямая линия для всех k . Далее, если в неколлинеарном определяющем многоугольнике встречаются l коллинеарных вершин, то прямые участки кривой (если они есть) начинаются и кончаются по крайней мере за $k - 2$ отрезка от начала и конца серии коллинеарных вершин. Если последовательность коллинеарных вершин полностью лежит внутри неколлинеарного многоугольника, число коллинеарных участков кривой не меньше, чем $l - 2k + 3$. Если же эта последовательность находится на конце неколлинеарного многоугольника, то число коллинеарных участков кривой не меньше $l - k + 1$. Иллюстрация приведена на рис. 5-33.

Если имеется $k - 1$ совпадающих вершин, т. е. $B_i = B_{i+1} = \dots = B_{i+k-2}$, то выпуклая оболочка вершин от B_i до B_{i+k-2} — это сама вершина. Отсюда следует, что В-сплайн должен проходить через вершину B_i . На рис. 5-34 изображен пример такой точки для $k = 3$. Далее, так как В-сплайн везде C^{k-2} непрерывен, он также C^{k-2} непрерывен в B_i .

Наконец, заметим, что свойство непрерывности плавно переводит В-сплайн во вложенные отрезки прямой, как показано на рис. 5-35.

Уравнения (5-84) указывают, что выбор узлового вектора оказывает существенное влияние на базисные функции В-сплайна $N_{i,k}(t)$ и, следовательно, на сам В-сплайн. Единственное требование к узловому вектору: $x_i \leq x_{i+1}$, т. е. это монотонно возрастающая последовательность вещественных чисел. Обычно используются три типа узловых векторов: равномерные, открытые равномерные (или открытые) и неравномерные.

Отдельные узловые значения равномерного узлового вектора распределены на одинаковом расстоянии, например

$$[0 \ 1 \ 2 \ 3 \ 4]$$

$$[-0.2 \ -0.1 \ 0 \ 0.1 \ 0.2]$$

В частности, равномерные узловые векторы обычно начинаются в нуле и увеличиваются на 1 к некоторому максимальному значению или нормируются в диапазоне между 0 и 1 равными десятичными значениями, например,

$$[0 \ 0.25 \ 0.5 \ 0.75 \ 1.0].$$

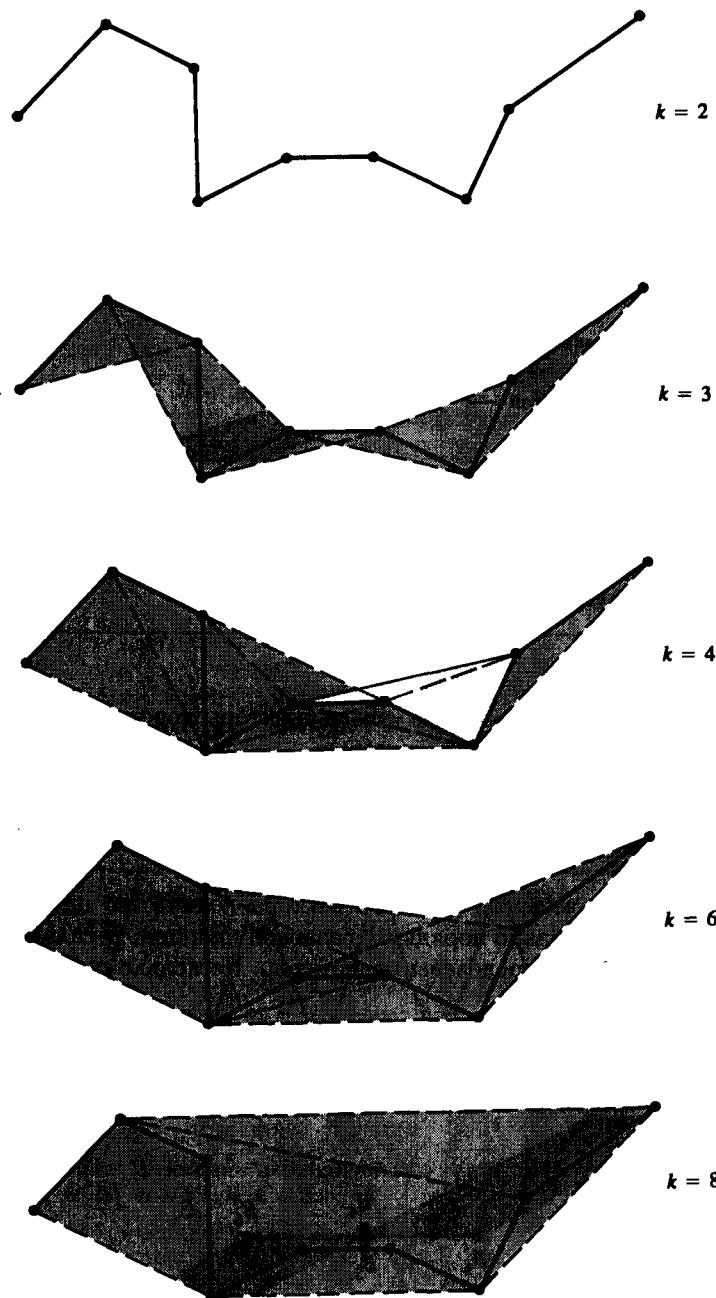


Рис. 5-32 Свойства выпуклой оболочки В-сплайнов.

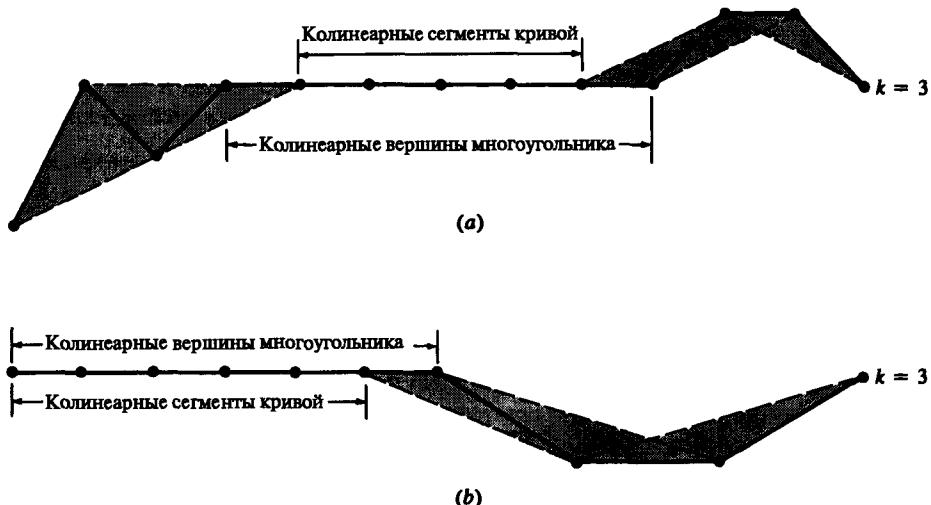


Рис. 5-33 Свойства выпуклой оболочки В-сплайнов для коллинеарных сегментов кривой. (а) Внутренние вершины определяющего многоугольника; (б) вершины в конце определяющего многоугольника.

Для данного порядка k равномерные узловые векторы порождают периодические равномерные функции базиса, для которых

$$N_{i,k}(t) = N_{i-1,k}(t-1) = N_{i+1,k}(t+1).$$

То есть каждая функция базиса — это параллельный перенос другой функции, см. рис. 5-36.

У открытого равномерного узлового вектора количество одинаковых узловых значений в концах равно порядку k базисной функции В-сплайна. Внутренние узловые значения распределены равномерно. Несколько примеров с целыми

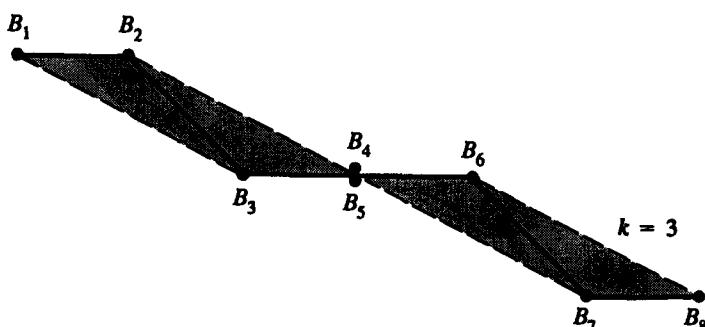


Рис. 5-34 Выпуклая оболочка для кратных вершин, $k = 3$.

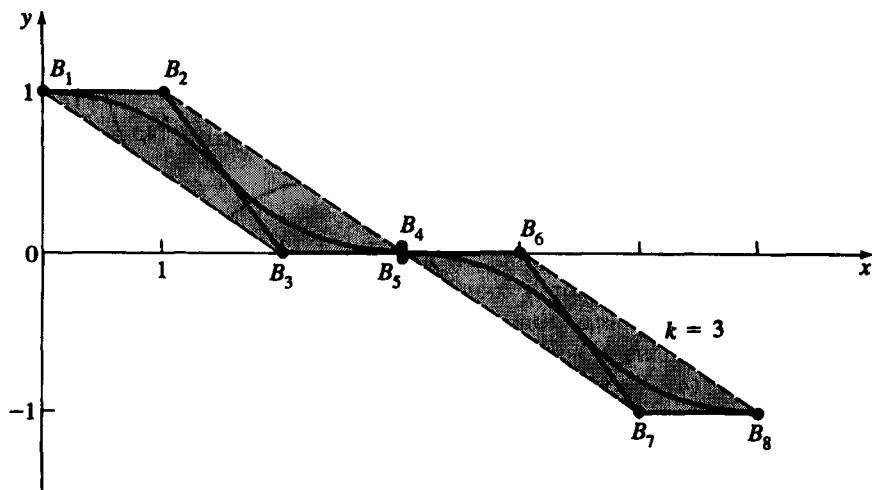


Рис. 5-35 Плавное (C^{k-2}) превращение в отрезки прямой.

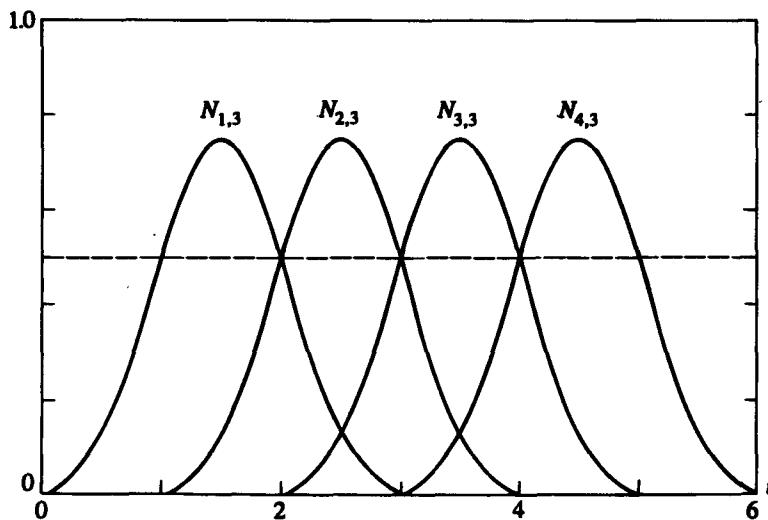


Рис. 5-36 Базисные функции периодического равномерного В-сплайна, $[X] = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6]$, $n+1=4$, $k=3$.

приращениями:

$$k=2 \quad [0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 4],$$

$$k=3 \quad [0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3],$$

$$k=4 \quad [0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2 \ 2]$$

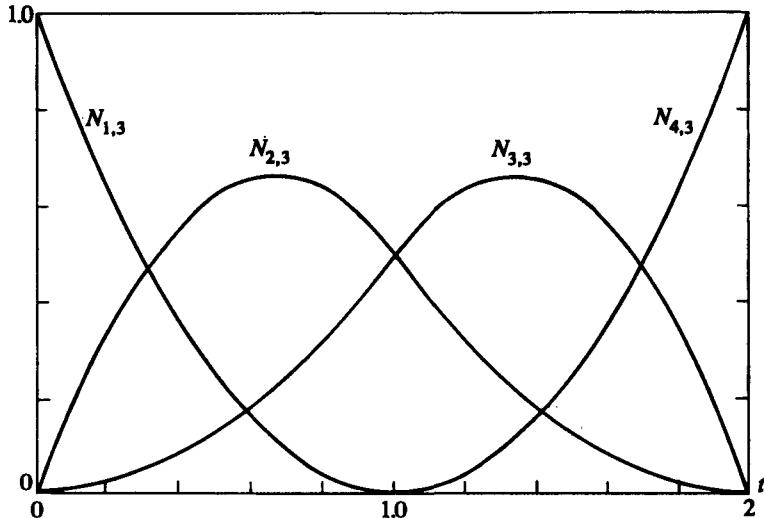


Рис. 5-37 Базисные функции открытого равномерного B-сплайна, $[X] = [0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2]$, $k = 3$, $n + 1 = 4$.

или для нормализованных приращений

$$k = 2 \quad [0 \ 0 \ 1/4 \ 1/2 \ 3/4 \ 1 \ 1],$$

$$k = 3 \quad [0 \ 0 \ 0 \ 1/3 \ 2/3 \ 1 \ 1 \ 1],$$

$$k = 4 \quad [0 \ 0 \ 0 \ 0 \ 1/2 \ 1 \ 1 \ 1 \ 1].$$

Формально открытый равномерный узловый вектор определяется как

$$x_i = 0 \quad 1 \leq i \leq k$$

$$x_i = i - k \quad k + 1 \leq i \leq n + 1$$

$$x_i = n - k + 2 \quad n + 2 \leq i \leq n + k + 1$$

Получающиеся базисные функции ведут себя примерно так же, как и кривые Безье. Фактически, если количество вершин многоугольника равно порядку базиса B-сплайна и используется открытый равномерный узловый вектор, базис B-сплайна сводится к базису Бернштейна. Отсюда B-сплайн является кривой Безье. В этом случае узловый вектор — это просто k нулей, за которыми следует k единиц. Например, для четырех вершин открытый равномерный узловый вектор:

$$[0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1].$$

В результате мы имеем кубическую кривую Безье — B-сплайн. Соответствующие базисные функции изображены на рис. 5-27б. На рис. 5-37 приведен еще один пример открытых базисных функций.

Неравномерные узловые векторы отличаются тем, что их внутренние узловые величины располагаются на разном расстоянии друг от друга и/или совмещаются.

Векторы могут быть периодическими или открытыми, например

$$[0 \ 0 \ 0 \ 1 \ 1 \ 2 \ 2 \ 2],$$

$$[0 \ 1 \ 2 \ 2 \ 3 \ 4],$$

$$[0 \ 0.28 \ 0.5 \ 0.72 \ 1].$$

На рис. 5.38б-е показаны примеры неравномерных базисных функций В-сплайна порядка $k = 3$. У соответствующих узловых векторов на концах находится по k совмещенных одинаковых значений. Для сравнения на рис. 5.38а приведены базисные функции для открытого равномерного вектора. Отметим, что на рис. 5.38а и б функции симметричны, а также что у неравномерных базисов симметрия нарушается: 5.38с-е. Кроме того, при совмещенных узловых значениях у одной из функций появляется излом. На рис. 5.38д и е видно, что положение излома зависит от расположения совмещенного значения в узловом векторе.

Формула Кокса–де Бура (5.84) для расчета базисных функций В-сплайна рекурсивна, поэтому функция порядка k зависит от базисных функций более низкого порядка вплоть до 1. Пусть дана базисная функция $N_{i,k}$. Тогда эту зависимость можно выразить в виде треугольника

$$\begin{array}{ccccccc} N_{i,k} & & & & & & \\ N_{i,k-1} & N_{i+1,k-1} & & & & & \\ N_{i,k-1} & N_{i+1,k-2} & N_{i+2,k-2} & & & & \\ & \cdot & & & & & \\ & N_{i,1} & N_{i+1,1} & N_{i+2,1} & N_{i+3,1} & \cdots & N_{i+k-1,1} \end{array}$$

Тогда обратная зависимость, т. е. влияние одной базисной функции первого порядка $N_{i,1}$ на функции более высоких порядков, такова:

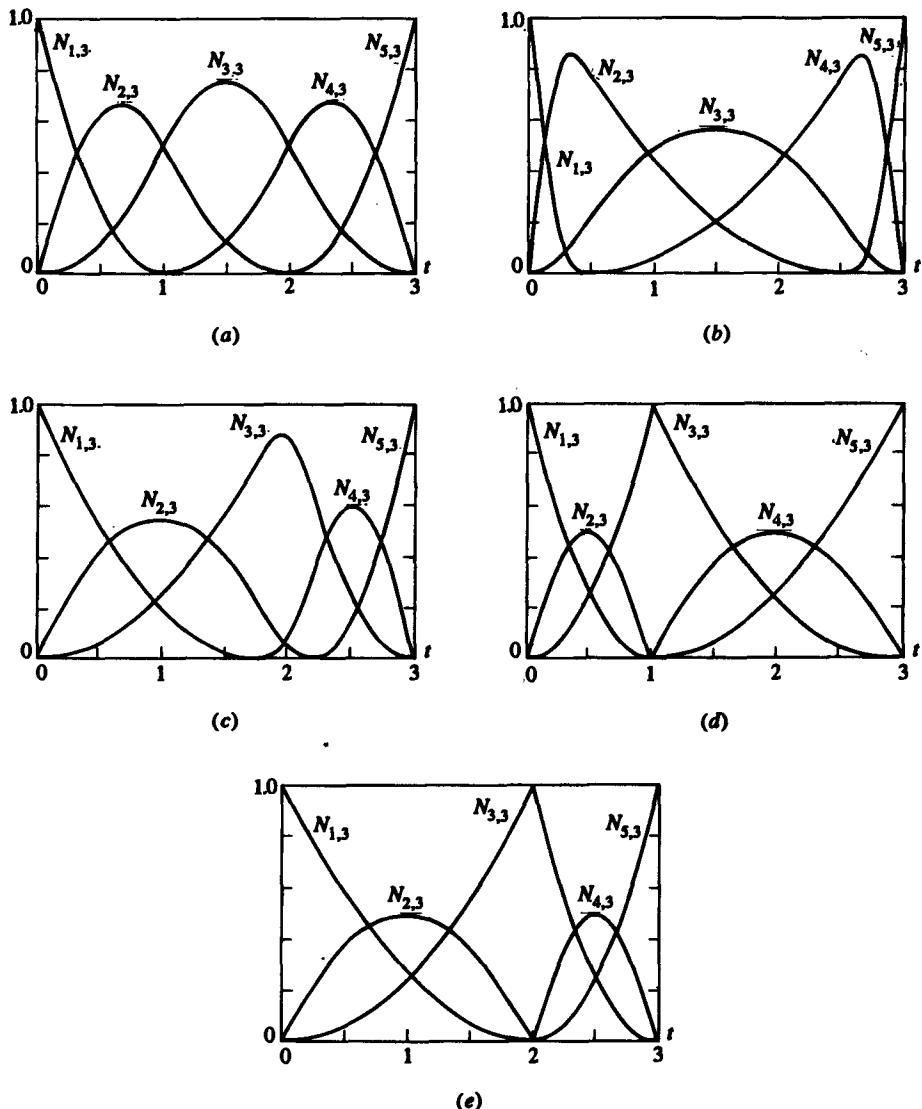
$$\begin{array}{ccccccc} N_{i-k+1,k} & \cdots & N_{i+k-1,k} & N_{i,k} & N_{i+1,k} & \cdots & N_{i+k-1,k} \\ & \cdot & & & & & \\ & N_{i-1,2} & N_{i,2} & N_{i+1,2} & & & \\ & & N_{i,1} & & & & \end{array}$$

Рассмотрим пример расчета базисных функций.

Пример 5-9 Расчет периодических базисных функций

Найти четыре базисные функции $N_{i,3}(t)$, $i = 1, 2, 3, 4$, третьего порядка ($k = 3$). Число $n+1$ в данном случае равно 4. Зависимость базисных функций для $N_{i,3}$ изображается следующей диаграммой:

$$\begin{array}{cccccc} N_{1,3} & N_{2,3} & N_{3,3} & N_{4,3} & & \\ N_{1,2} & N_{2,2} & N_{3,2} & N_{4,2} & N_{5,2} & \\ N_{1,1} & N_{2,1} & N_{3,1} & N_{4,1} & N_{5,1} & N_{6,1} \end{array}$$

Рис. 5-38 Функции неравномерного базиса для $n + 1 = 5$, $k = 3$.

(a) $[X] = [0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3]$;

(b) $[X] = [0 \ 0 \ 0 \ .4 \ 2.6 \ 3 \ 3 \ 3]$;

(c) $[X] = [0 \ 0 \ 0 \ 1.8 \ 2.2 \ 3 \ 3 \ 3]$;

(d) $[X] = [0 \ 0 \ 0 \ 1 \ 1 \ 3 \ 3 \ 3]$;

(e) $[X] = [0 \ 0 \ 0 \ 2 \ 2 \ 3 \ 3 \ 3]$.

Обратные зависимости для $i \geq 1$:

$$\begin{array}{ccccccc} N_{1,3} & N_{2,3} & N_{3,3} & N_{4,3} & N_{5,3} & N_{6,3} \\ N_{1,2} & N_{2,2} & N_{3,2} & N_{4,2} & N_{5,2} & \\ N_{1,1} & N_{2,1} & N_{3,1} & N_{4,1} & & \end{array}$$

Каков диапазон узлового вектора, необходимый для этого расчета? Из уравнения (5-84) следует, что для вычисления $N_{6,1}$ необходимы узловые величины x_6 и x_7 , а для $N_{1,1} - x_1$ и x_2 , т. е. необходимы значения от 0 до $n+k$. Отсюда количество узловых значений равно $n+k+1$. Узловой вектор для заданных периодических функций:

$$[X] = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6],$$

где $x_1 = 0, \dots, x_7 = 6$. Диапазон параметра $0 \leq t \leq 6$. Используя уравнение (5-84) и приведенные выше диаграммы, получим базисные функции для различных значений параметра:

$$0 \leq t < 1$$

$$N_{1,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 1$$

$$N_{1,2}(t) = t; \quad N_{i,2}(t) = 0, \quad i \neq 1$$

$$N_{1,3}(t) = \frac{t^2}{2}; \quad N_{i,3}(t) = 0, \quad i \neq 1$$

$$1 \leq t < 2$$

$$N_{2,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 2$$

$$N_{1,2}(t) = (2-t); \quad N_{2,2}(t) = (t-1); \quad N_{1,2}(t) = 0. \quad i \neq 1, 2$$

$$N_{1,3}(t) = \frac{t}{2}(2-t) + \left(\frac{3-t}{2}\right)(t-1);$$

$$N_{2,3}(t) = \frac{(t-1)^2}{2}; \quad N_{i,3}(t) = 0, \quad i \neq 1, 2, 3$$

$$2 \leq t < 3$$

$$N_{3,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 3$$

$$N_{2,2}(t) = (3-t); \quad N_{3,2}(t) = (t-2); \quad N_{i,2}(t) = 0, \quad i \neq 2, 3$$

$$N_{1,3}(t) = \frac{(3-t)^2}{2};$$

$$N_{2,3}(t) = \frac{(t-1)(3-t)}{2} + \frac{(4-t)(t-2)}{2};$$

$$N_{3,3}(t) = \frac{(t-2)^2}{2}; \quad N_{i,3}(t) = 0, \quad i \neq 1, 2, 3$$

$$3 \leq t < 4$$

$$N_{4,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 4$$

$$N_{3,2}(t) = (4-t); \quad N_{4,2}(t) = (t-3); \quad N_{i,2}(t) = 0, \quad i \neq 3, 4$$

$$N_{2,3}(t) = \frac{(4-t)^2}{2}; \quad N_{3,3}(t) = \frac{(t-2)(4-t)}{2} + \frac{(5-t)(t-3)}{2};$$

$$N_{4,3}(t) = \frac{(t-3)^2}{2}; \quad N_{i,3}(t) = 0, \quad i \neq 2, 3, 4$$

$$4 \leq t < 5$$

$$N_{5,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 5$$

$$N_{4,2}(t) = (5-t); \quad N_{5,2}(t) = (t-4); \quad N_{i,2}(t) = 0, \quad i \neq 4, 5$$

$$N_{3,3}(t) = \frac{(5-t)^2}{2};$$

$$N_{4,3}(t) = \frac{(t-3)(5-t)}{2} + \frac{(6-t)(t-4)}{2};$$

$$N_{i,3}(t) = 0, \quad i \neq 3, 4$$

$$5 \leq t < 6$$

$$N_{6,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 6$$

$$N_{5,2}(t) = (6-t); \quad N_{i,2}(t) = 0, \quad i \neq 5$$

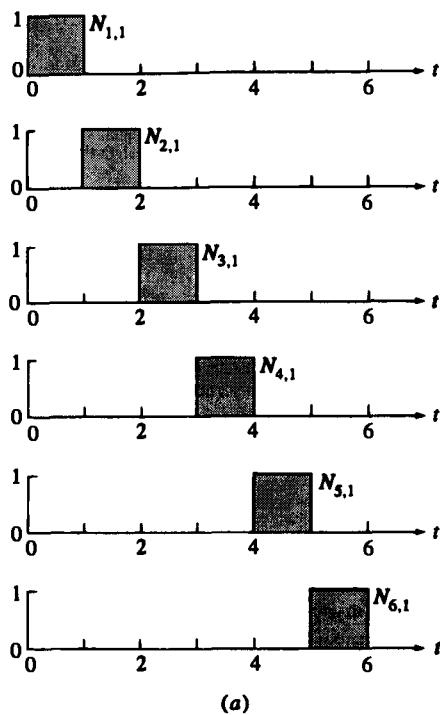
$$N_{4,3}(t) = \frac{(6-t)^2}{2}; \quad N_{i,3}(t) = 0, \quad i \neq 4.$$

Знак $<$ в определении $N_{i,1}$ приводит к тому, что при $t = 6$ все функции равны 0.

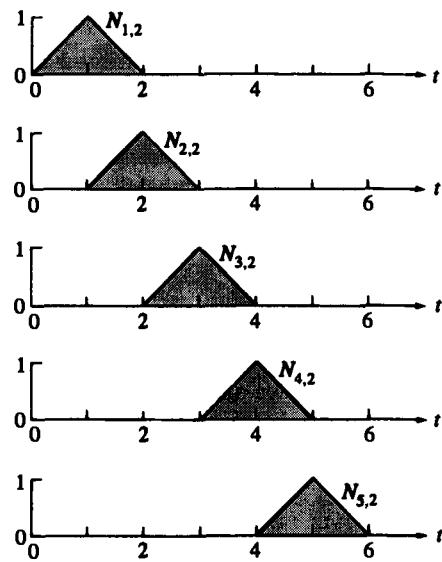
Результаты показаны на рис. 5-36 и 5-39c. Отметим, что каждая из базисных функций является кусочной параболической (квадратичной) кривой. Параболические сегменты на интервалах $x_i \rightarrow x_{i+1}$, $x_{i+1} \rightarrow x_{i+2}$, $x_{i+2} \rightarrow x_{i+3}$ объединяются и составляют базисные функции $N_{i,3}$. Каждая функция представляет собой параллельный перенос другой.

Пример 5-9 показывает, как построить базис N по функциям базиса более низкого порядка. На рис. 5-39a изображены функции первого порядка из примера 5-9, на рис. 5-39b — второго порядка, и на рис. 5-39c — третьего порядка. Обратим внимание на то, как растягивается диапазон ненулевых значений функций с увеличением их порядка. Говорят, что функция базиса обеспечивает поддержку на интервале от x_i до x_{i+k} .

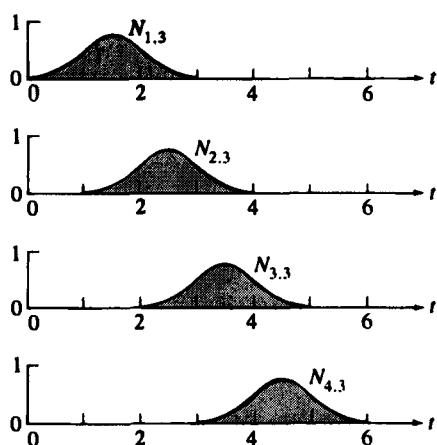
Внимательно рассматривая рис. 5-36, можно заметить важное свойство функций равномерного базиса. Из уравнения (5-85) известно, что $\sum N_{i,k}(t) = 1$ для любого значения параметра t . Отсюда следует, что все множество периодических базисных функций для $k = 3$ определено только в диапазоне $2 \leq t \leq 4$. За его границами $\sum N_{i,k}(t) \neq 1$. Для равномерного начинающегося с 0 узлового вектора с целыми приращениями пригодный диапазон параметра: $k - 1 \leq t \leq (n + k) - (k - 1) = n + 1$. Для более общих или нормализованных векторов сокращение диапазона параметра соответствует потере $k - 1$ интервала узловых значений на каждом конце вектора.



(a)



(b)



(c)

Рис. 5-39 Построение периодических базисных функций $n+1 = 4$. (а) $k = 1$; (б) $k = 2$; (в) $k = 3$.

Пример 5-10 Расчет открытого равномерного базиса

Найти четыре ($n = 3$) базисные функции $N_{i,3}(t)$, $i = 1, 2, 3, 4$, третьего порядка ($k = 3$).

Напомним, что формально открытый узловой вектор с целыми интервалами между внутренними узловыми значениями определяется как

$$\begin{aligned}x_i &= 0, \quad 1 \leq i \leq k, \\x_i &= i - k, \quad k + 1 \leq i \leq n + 1, \\x_i &= n - k + 2, \quad n + 2 \leq i \leq n + k + 1.\end{aligned}$$

Диапазон изменения параметра $0 \leq t \leq n - k + 2$, т. е. от 0 до максимального узлового значения. Как и в примере 5-9, количество узловых значений равно $n + k + 1$. Если брать узловые значения, то вектор для данного примера примет вид

$$[X] = [0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2],$$

где $x_1 = 0, \dots, x_7 = 2$. Параметр t изменяется от 0 до 2.

Пользуясь уравнениями (5-84) и диаграммами зависимости, получаем функции базиса для различных диапазонов параметра:

$$0 \leq t < 1$$

$$N_{3,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 3$$

$$N_{2,2}(t) = 1 - t; \quad N_{3,2}(t) = t; \quad N_{i,2}(t) = 0, \quad i \neq 2, 3$$

$$N_{1,3}(t) = (1 - t)^2; \quad N_{2,3}(t) = t(1 - t) + \frac{(2 - t)}{2}t;$$

$$N_{3,3}(t) = \frac{t^2}{2}; \quad N_{i,3}(t) = 0, \quad i \neq 1, 2, 3$$

$$1 \leq t < 2$$

$$N_{4,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 4$$

$$N_{3,2}(t) = (2 - t); \quad N_{4,2}(t) = (t - 1); \quad N_{i,2}(t) = 0, \quad i \neq 3, 4$$

$$N_{2,3}(t) = \frac{(2 - t)^2}{2}; \quad N_{3,3}(t) = \frac{t(2 - t)}{2} + (2 - t)(t - 1);$$

$$N_{4,3}(t) = (t - 1)^2; \quad N_{i,3}(t) = 0, \quad i \neq 2, 3, 4.$$

Эти результаты приведены на рис. 5-40.

Сравнивая результаты примера 5-10 (рис. 5-40) и 5-9 (рис. 5-39), мы видим, что они существенно различаются для периодического и открытого равномерного узловых векторов. В частности, отметим, что у открытых равномерных узловых векторов на всем диапазоне изменения параметра определен полный набор базисных функций; т. е. $\sum N_{i,k}(t) = 1$ для всех $0 \leq t \leq n - k + 2$. У периодического вектора диапазон параметра уменьшается.

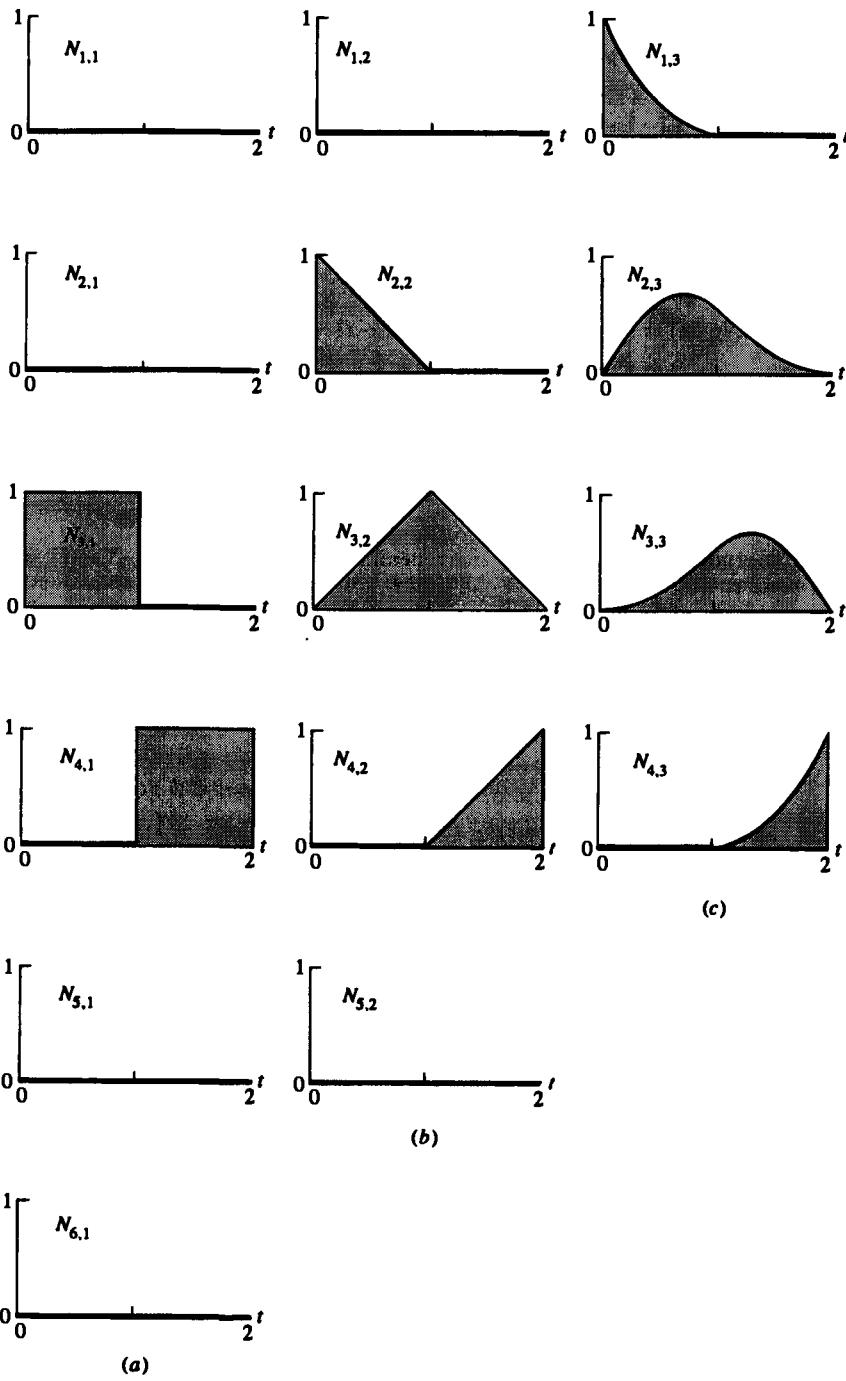


Рис. 5-40 Построение открытых базисных функций $n + 1 = 4$. (a) $k = 1$; (b) $k = 2$; (c) $k = 3$.

Пример 5-11 Расчет неравномерных базисных функций

Найти пять ($n+1 = 5$) базисных функций $N_{i,3}(t)$, $i = 1, 2, 3, 4, 5$, третьего порядка ($k = 3$) с узловым вектором $[X] = [0 \ 0 \ 0 \ 1 \ 1 \ 3 \ 3 \ 3]$, содержащим повторяющееся внутреннее узловое значение. Из уравнений (5-84) и диаграмм зависимости

$$0 \leq t < 1$$

$$N_{3,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 2$$

$$N_{2,2}(t) = 1 - t; \quad N_{3,2}(t) = t; \quad N_{i,2}(t) = 0, \quad i \neq 2, 3$$

$$N_{1,3}(t) = (1-t)^2; \quad N_{2,3}(t) = t(1-t) + (1-t)t = 2t(1-t);$$

$$N_{3,3}(t) = t^2; \quad N_{i,3}(t) = 0, \quad i \neq 1, 2, 3$$

$$1 \leq t < 3$$

$$N_{i,1}(t) = 0, \quad \text{все } i$$

$$N_{i,2}(t) = 0, \quad \text{все } i$$

$$N_{i,3}(t) = 0, \quad \text{все } i.$$

В частности, как следствие повторения узлового значения, $N_{4,1}(t) = 0$ для всех t .

$$1 \leq t < 3$$

$$N_{5,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 5$$

$$N_{4,2}(t) = \frac{(3-t)}{2}; \quad N_{5,2}(t) = \frac{(t-1)}{2}; \quad N_{i,2}(t) = 0, \quad i \neq 4, 5$$

$$N_{3,3}(t) = \frac{(3-t)^2}{4};$$

$$N_{4,3}(t) = \frac{(t-1)(3-t)}{4} + \frac{(3-t)(t-1)}{4} = \frac{(3-t)(t-1)}{2};$$

$$N_{5,3}(t) = \frac{(t-1)^2}{4}; \quad N_{i,3}(t) = 0, \quad i \neq 3, 4, 5.$$

Результат приведен на рис.5-38d.

Заметим, что для всех значений t имеем $\sum N_{i,k}(t) = 1.0$. Например, для $0 \leq t < 1$,

$$\sum_{i=1}^5 N_{i,3}(t) = (1-t)^2 + 2t(1-t) + t^2 = 1 - 2t + t^2 + 2t - 2t^2 + t^2 = 1.$$

Аналогично для $1 \leq t < 3$

$$\begin{aligned} \sum_{i=1}^5 N_{i,3}(t) &= \frac{1}{4}[(3-t)^2 + 2(3-t)(t-1) + (t-1)^2] \\ &= \frac{1}{4}[9 - 6t + t^2 - 6 + 8t - 2t^2 + 1 - 2t + t^2] \\ &= \frac{4}{4} = 1 \end{aligned}$$

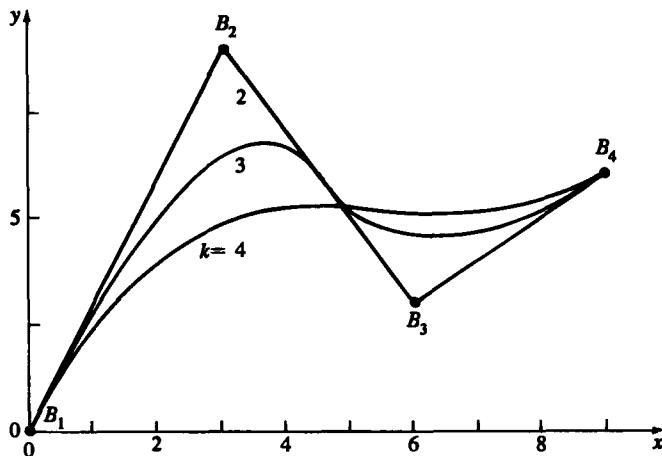


Рис. 5-41 Зависимость формы В-сплайна от его порядка.

Из высказанных видно, какое влияние оказывает выбор узлового вектора на вид базисных функций В-сплайна и, следовательно, на форму В-сплайна.

Гибкость базиса В-сплайна позволяет воздействовать на форму кривой разными способами:

Изменяя тип узлового вектора и базиса: периодический равномерный, открытый равномерный и неравномерный.

Меняя порядок k базисных функций.

Меняя количество и расположение вершин определяющего многоугольника.

Используя повторяющиеся вершины.

Используя повторяющиеся узловые значения в узловых векторах.

Рассмотрим эти способы сначала для открытых В-сплайнов, затем для равномерных периодических и неравномерных В-сплайнов.

Открытый В-сплайн по своим свойствам во многом аналогичен кривым Безье. Как уже отмечалось, если порядок В-сплайна равен количеству вершин определяющего многоугольника, то базис В-сплайна сводится к базису Бернштейна, а сам В-сплайн становится кривой Безье. У открытого В-сплайна любого порядка ($k \geq 2$) первая и последняя точки кривой совпадают с соответственными вершинами многоугольника. Далее, наклон кривой в первой и последней вершинах многоугольника равен наклону соответственных сторон многоугольника.

На рис. 5-41 изображены три открытых В-сплайна различного порядка, заданные одним набором из четырех вершин. Кривая четвертого порядка — это кривая Безье — один кубический полиномиальный сегмент. Кривая третьего порядка состоит из двух параболических сегментов, соединяющихся в центре второго отрезка с непрерывностью C^1 . Кривая второго порядка совпадает с определяющим многоугольником. Она состоит из трех линейных сегментов, соединяющихся во второй и третьей вершинах с непрерывностью C^0 . Угол наклона на концах, заданный наклоном сторон многоугольника, одинаков для всех трех кривых. Отметим

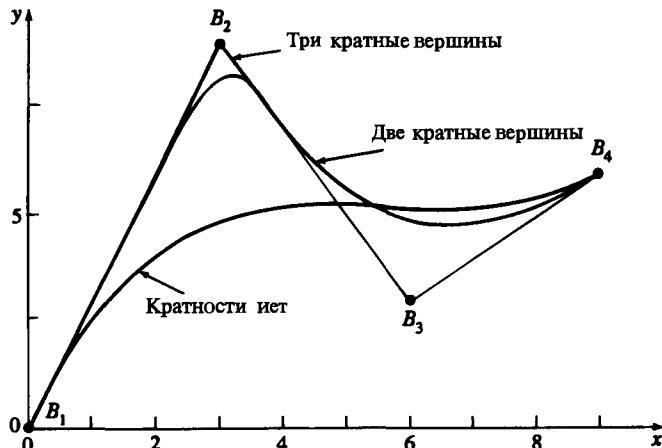


Рис. 5-42 Влияние кратности вершины на форму B-сплайна, $k = 4$.

также, что по мере возрастания порядка кривой, она все меньше напоминает исходный многоугольник и становится более гладкой.

На рис. 5-42 изображается влияние повторяющихся или совпадающих вершин. Все B-сплайны имеют порядок $k = 4$. Нижняя кривая определена четырьмя вершинами с узловым вектором $[0 \ 0 \ 0 \ 1 \ 1 \ 1]$. У средней кривой пять определяющих вершин, причем две из них повторяются во второй вершине многоугольника [3 9]. Узловой вектор — $[0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2]$. Верхняя кривая определена шестью вершинами с тремя повторяющимися в точке [3 9]. Узловой вектор — $[0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3]$. Соответствующие многоугольники для трех кривых соответственно таковы: B_1, B_2, B_3, B_4 ; B_1, B_2, B_2, B_3, B_4 и $B_1, B_2, B_2, B_2, B_3, B_4$.

Нижняя кривая состоит из единственного кубического сегмента. Средняя кривая состоит из двух сегментов, соединенных между B_2 и B_3 . Верхняя кривая состоит из трех сегментов: первый от B_1 до B_2 , второй от B_2 до середины между B_2 и B_3 , третий от этой точки до B_4 . Обратим внимание на то, что с увеличением кратности вершины B_2 кривая все ближе подходит к B_2 . Когда кратность достигается $k - 1$, возникает острый угол в соответствии со свойством выпуклой оболочки B-сплайнов. При внимательном изучении рис. 5-42 можно заметить, что на обеих сторонах кратной вершины имеется линейный участок.

Несмотря на наличие углов, сохраняется C^{k-2} дифференцируемость кривой. На первый взгляд это может показаться противоречием, однако излом определяется нулевым касательным вектором, что не исключает непрерывности его изменения. Возможность включения острых углов и изломов в непрерывно C^{k-2} дифференцируемые кривые это важное свойство B-сплайнов.

Наконец, заметим, что у всех кривых наклон в концах одинаков.

На рис. 5-43 показаны три B-сплайна четвертого порядка. Каждый определяющий многоугольник состоит из восьми вершин. Кривые отличаются тем, что точка B_5 передвигается в B'_5 и B''_5 . Передвижение точки B_5 воздействует на кривую только локально: изменяются лишь сегменты, отвечающие отрезкам B_3B_4 ,

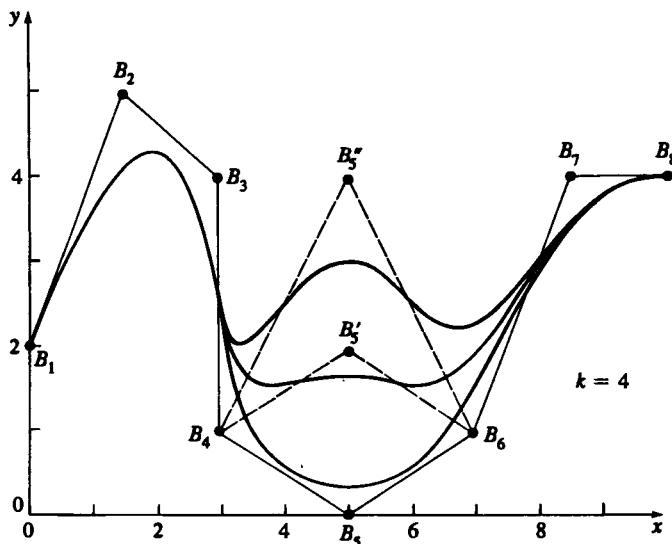


Рис. 5-43 Локальная коррекция В-сплайна.

B_4B_5 и B_5B_6 , B_6B_7 . В общем случае затрагиваются только $\pm k/2$ отрезков вокруг сдвинутой точки.

Для иллюстрации рассмотрим пример.

Пример 5-12 Расчет открытого В-сплайна

Рассмотрим многоугольник из примера 5-7: $B_1[1 \ 1]$, $B_2[2 \ 3]$, $B_3[4 \ 3]$, $B_4[3 \ 1]$. Найти В-сплайн второго и четвертого порядка. Для $k = 2$ открытый узловый вектор

$$[0 \ 0 \ 1 \ 2 \ 3 \ 3],$$

где $x_1 = 0$, $x_2 = 0, \dots, x_6 = 3$. Параметр t изменяется от 0 до 3. Кривая состоит из трех линейных ($k - 1 = 1$) сегментов. Для $0 \leq t < 3$ функции базиса имеют вид:

$$0 \leq t < 1$$

$$N_{2,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 2$$

$$N_{1,2}(t) = 1 - t; \quad N_{2,2}(t) = t; \quad N_{i,2}(t) = 0, \quad i \neq 1, 2$$

$$1 \leq t < 2$$

$$N_{3,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 3$$

$$N_{2,2}(t) = 2 - t; \quad N_{3,2}(t) = (t - 1); \quad N_{i,2}(t) = 0, \quad i \neq 2, 3$$

$$2 \leq t < 3$$

$$N_{4,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 4$$

$$N_{3,2}(t) = (3 - t); \quad N_{4,2}(t) = (t - 2); \quad N_{i,2}(t) = 0, \quad i \neq 3, 4$$

Из уравнения (5-83) получаем параметрический В-сплайн

$$P(t) = B_1N_{1,2}(t) + B_2N_{2,2}(t) + B_3N_{3,2}(t) + B_4N_{4,2}(t).$$

Для каждого из этих интервалов

$$P(t) = (1-t)B_1 + tB_2 = B_1 + (B_2 - B_1)t \quad 0 \leq t < 1,$$

$$P(t) = (2-t)B_2 + (t-1)B_3 = B_2 + (B_3 - B_2)t \quad 1 \leq t < 2,$$

$$P(t) = (3-t)B_3 + (t-2)B_4 = B_3 + (B_4 - B_3)t \quad 2 \leq t < 3.$$

В каждом случае в результате получается уравнение прямой для ребра многоугольника, т. е. кривая совпадает с многоугольником.

Последняя точка кривой ($t = t_{\max} = 3$) требует особого внимания. Из-за того что интервал в уравнении (5-84а) открыт справа, все базисные функции N_{ik} при $t = 3$ равны нулю. Следовательно, последняя точка многоугольника формально не лежит на кривой. Фактически же это не так. Рассмотрим $t = 3 - \varepsilon$, где ε – бесконечно малая величина. Если ε стремится к нулю, то в пределе последние точки кривой и многоугольника совпадают. На практике либо явно добавляют последнюю точку в описание кривой, либо определяют $N(t = t_{\max}) = 1.0$.

Для $k = 4$ порядок кривой совпадает с количеством вершин определяющего многоугольника, поэтому В-сплайн сводится к кривой Безье. Узловой вектор с $t_{\max} = n - k + 2 = 3 - 4 + 2 = 1 - [0 \ 0 \ 0 \ 1 \ 1 \ 1]$. Функции базиса таковы:

$$0 \leq t < 1$$

$$N_{4,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 4$$

$$N_{3,2}(t) = (1-t); \quad N_{4,2}(t) = t; \quad N_{i,2}(t) = 0, \quad i \neq 3, 4$$

$$N_{2,3}(t) = (1-t)^2; \quad N_{3,3}(t) = 2t(1-t);$$

$$N_{4,3}(t) = t^2; \quad N_{i,3}(t) = 0, i \neq 2, 3, 4$$

$$N_{1,4}(t) = (1-t)^3; \quad N_{2,4}(t) = t(1-t)^2 + 2t(1-t)^2 = 3t(1-t)^2;$$

$$N_{3,4}(t) = 2t^2(1-t) + (1-t)t^2 = 3t^2(1-t); \quad N_{4,4}(t) = t^3$$

Из уравнения (5-83) получаем параметрический В-сплайн

$$P(t) = B_1 N_{1,4}(t) + B_2 N_{2,4}(t) + B_3 N_{3,4}(t) + B_4 N_{4,4}(t),$$

$$P(t) = (1-t)^3 B_1 + 3t(1-t)^2 B_2 + 3t^2(1-t) B_3 + t^3 B_4.$$

Итак, при $t = 0$

$$P(0) = B_1$$

и при $t = 1/2$

$$P\left(\frac{1}{2}\right) = \frac{1}{8}B_1 + \frac{3}{8}B_2 + \frac{3}{8}B_3 + \frac{1}{8}B_4$$

и

$$\begin{aligned} P\left(\frac{1}{2}\right) &= \frac{1}{8}[1 \quad 1] + \frac{3}{8}[2 \quad 3] + \frac{3}{8}[4 \quad 3] + \frac{1}{8}[3 \quad 1] \\ &= [11/4 \quad 5/2]. \end{aligned}$$

Сравнивая с примером 5-7, видим, что результаты одинаковы. Получившаяся кривая изображена на рис. 5-28.

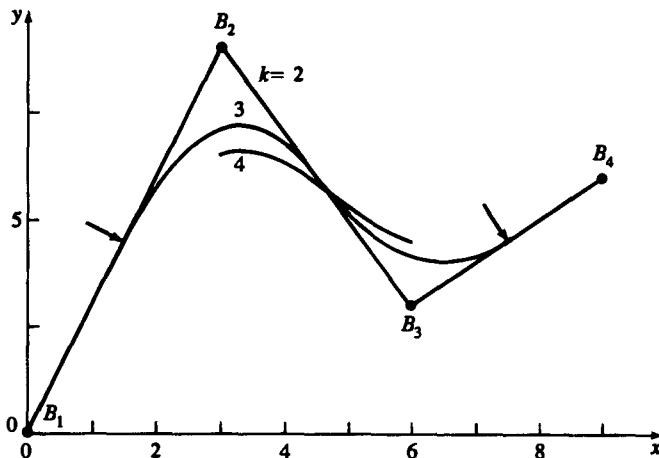


Рис. 5-44 Зависимость формы периодического В-сплайна от его порядка.

Теперь займемся периодическими В-сплайнами. На рис. 5-44 показаны три периодических В-сплайна разного порядка. Все кривые определены теми же вершинами, что и для открытых В-сплайнов на рис. 5-41. Для $k = 2$ В-сплайн опять совпадает с определяющим многоугольником. Отметим, однако, что у периодического сплайна при $k > 2$ первая и последняя точки на кривой не совпадают с первой и последней точками многоугольника. Наклон в первой и последней точках также может отличаться от наклона соответствующих сторон многоугольника. Для $k = 3$ В-сплайн начинается в середине первого ребра и оканчивается в середине последнего, как отмечено стрелками. Это происходит из-за сокращения диапазона параметра для базисных функций периодического В-сплайна. Для $k = 2$ периодический узловый вектор — $[0 \ 1 \ 2 \ 3 \ 4 \ 5]$ с диапазоном параметра $1 \leq t \leq 4$. Для $k = 3$ периодический узловый вектор — $[0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6]$ с диапазоном параметра $2 \leq t \leq 4$. Для $k = 4$ периодический узловый вектор — $[0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$ с диапазоном параметра $3 \leq t \leq 4$.

Сравнение полученных результатов и результатов для открытых узловых векторов на рис. 5-41 показывает, что кривую можно определить на полном диапазоне параметра, задавая кратные узловые значения на концах векторов. При этом кривая растягивается к концам многоугольника.

В данном случае кривая четвертого порядка опять состоит из единственного кубического сегмента; кривая третьего порядка — из двух параболических сегментов, соединенных в середине второго ребра с непрерывностью C^1 ; кривая второго порядка — из трех линейных сегментов, соединенных во второй и третьей вершинах с C^0 непрерывностью. Увеличение порядка опять сглаживает кривую, но в то же время и укорачивает ее.

Рис. 5-45 показывает, что воздействие кратных вершин одинаково для периодических и открытых В-сплайнов. Область вокруг точки изображена более подробно.

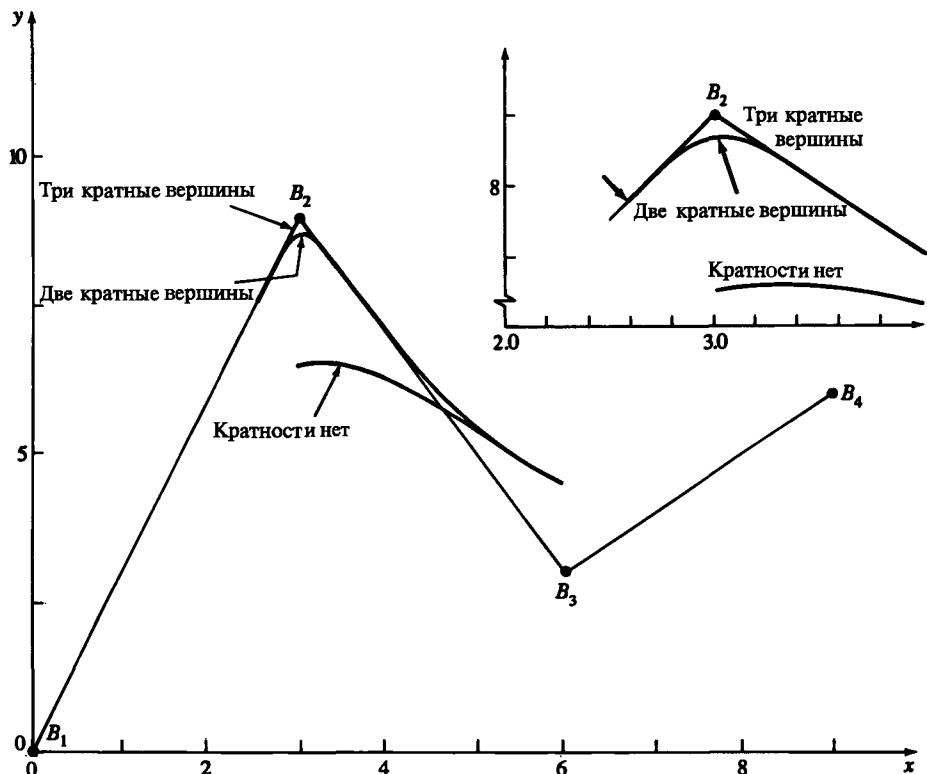


Рис. 5-45 Влияние кратности вершины на форму периодического В-сплайна, $k = 4$.

Пример 5-13 Расчет периодического В-сплайна

Снова рассмотрим многоугольник на рис. 5-44. Вершины многоугольника — $B_1[0 \ 0]$, $B_2[3 \ 9]$, $B_3[6 \ 3]$, $B_4[9 \ 6]$. Найти периодический В-сплайн четвертого порядка ($k = 4$), заданный этим многоугольником.

Для $k = 4$ на диапазоне $3 \leq t < 4$ узловый вектор для периодических базисных функций — $[0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$. Для этого диапазона базисные функции первого порядка (см. уравнение (5-84a), $(3 \leq t < 4)$):

$$N_{4,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 4$$

Из уравнения (5-84b) получаем базисные функции высших порядков:

$$N_{3,2}(t) = (4 - t); \quad N_{4,2}(t) = (t - 3); \quad N_{i,2}(t) = 0, \quad i \neq 3, 4$$

$$N_{2,3}(t) = \frac{(4 - t)^2}{2}; \quad N_{3,3}(t) = \frac{(t - 2)(4 - t)}{2} + \frac{(5 - t)(t - 3)}{2};$$

$$N_{4,3}(t) = \frac{(t - 3)^2}{2}; \quad N_{i,3}(t) = 0, \quad i \neq 2, 3, 4$$

$$N_{1,4}(t) = \frac{(4 - t)^3}{6};$$

$$N_{2,4}(t) = \frac{(t-1)(4-t)^2}{6} + \frac{(5-t)(4-t)(t-2)}{6} + \frac{(5-t)^2(t-3)}{6};$$

$$N_{3,4}(t) = \frac{(t-2)^2(4-t)}{6} + \frac{(t-2)(t-3)(5-t)}{6} + \frac{(6-t)(t-3)^2}{6};$$

$$N_{4,4}(t) = \frac{(t-3)^3}{6}$$

При $t = 3$

$$N_{1,4}(3) = \frac{(4-3)^2}{6} = \frac{1}{6}$$

$$N_{2,4}(3) = \frac{(3-1)(4-3)^2}{6} + \frac{(5-3)(4-3)(3-2)}{6} + \frac{(5-3)^2(3-3)}{6} = \frac{2}{3}$$

$$N_{3,4}(3) = \frac{(3-2)^2(4-3)}{6} + \frac{(3-2)(3-3)(5-3)}{6} + \frac{(6-3)(3-3)^2}{6} = \frac{1}{6}$$

$$N_{4,4}(3) = \frac{(3-3)^3}{6} = 0$$

Точка на В-сплайне при $t = 3$ имеет вид

$$P(3) = \frac{1}{6}B_1 + \frac{2}{3}B_2 + 0B_4 =$$

$$= \frac{1}{6}[0 \quad 0] + \frac{2}{3}[3 \quad 9] + \frac{1}{6}[6 \quad 3] + 0[9 \quad 6] =$$

$$= [3 \quad 6.5]$$

Полная кривая приведена на рис. 5-44.

Периодические В-сплайны очень удобны для построения замкнутых кривых. На рис. 5-46а изображен периодический В-сплайн четвертого порядка ($k = 4$), построенный для замкнутого многоугольника $B_1B_2B_3B_4B_5B_6B_7B_8B_1$.

Первая вершина совпадает с последней. В-сплайн получается незамкнутым из-за ограниченного диапазона параметра. Здесь периодический равномерный узловый вектор: $[0 \quad 1 \quad 2 \dots 10 \quad 11 \quad 12]$ с диапазоном параметра $3 \leq t \leq 9$.

Повторяя всего $k-2$ вершин в начале и/или конце замкнутого многоугольника, получаем замкнутый периодический В-сплайн. (Далее в этом разделе обсуждается другой метод в матричной формулировке.) Результат показан на рис. 5-46б, где определяющий многоугольник — $B_8B_1B_2B_3B_4B_5B_6B_7B_8B_1B_2$ с периодическим равномерным узловым вектором $[0 \quad 1 \quad 2 \dots 12 \quad 13 \quad 14]$ и диапазоном параметра $3 \leq t \leq 11$. Альтернативные многоугольники $B_1B_2B_3B_4B_5B_6B_7B_8B_1B_2B_3$ или $B_7B_8B_1B_2B_3B_4B_5B_6B_7B_8B_1$ приводят к тому же самому результату.

На рис. 5-47 одна вершина многоугольника B_4 передвинута на другое место. Ее воздействие распространяется на сегменты кривой, соответствующие $\pm k/2$ ребрам многоугольника по обе стороны от сдвинутой точки. На рис. 5-48 показан эффект кратной вершины B_4 . Область вокруг B_4 изображена более подробно. Опять отметим, что ее влияние распространяется также только на $\pm k/2$ сегмента кривой по обе стороны кратной вершины.

Теперь рассмотрим неравномерные В-сплайны. На рис. 5-49 кривая изменяется под воздействием кратных внутренних узловых значений. Верхняя кри-

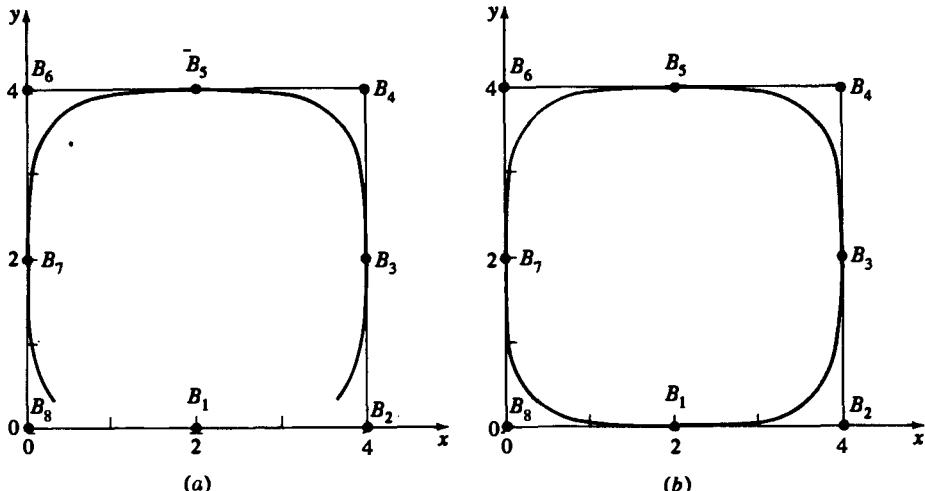


Рис. 5-46 Замкнутый периодический B-сплайн. (a) $B_1B_2B_3B_4B_5B_6B_7B_8B_1$ определяющий многоугольник; (b) $B_8B_1B_2B_3B_4B_5B_6B_7B_8B_1B_2$ — определяющий многоугольник.

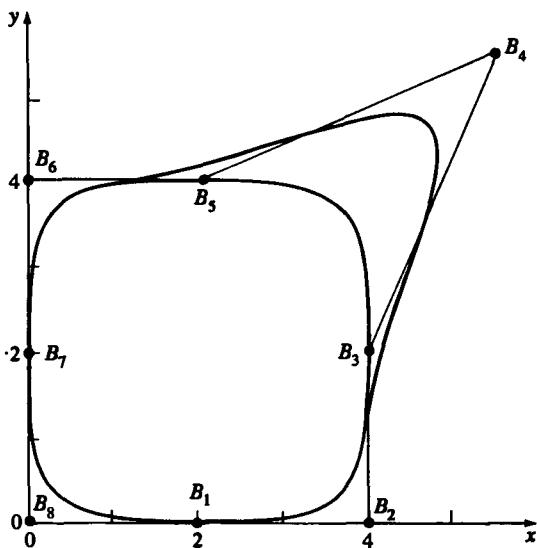


Рис. 5-47 Изменение формы В-сплайна после перемещения одной вершины многоугольника.

вая третьего порядка ($k = 3$) рассчитана для открытого узлового вектора $[0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3]$. Базисные функции для этой кривой изображены на рис. 5-38а. Нижняя кривая третьего порядка построена с неравномерным узловым вектором $[0 \ 0 \ 0 \ 1 \ 1 \ 3 \ 3 \ 3]$. Ее базисные функции — на

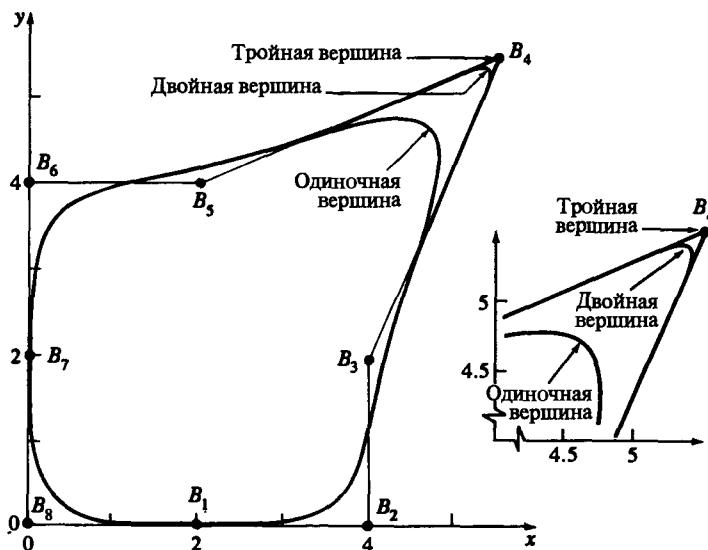


Рис. 5-48 Влияние кратных вершин на форму замкнутого периодического В-сплайна.

рис. 5-38d. Такая же кривая получается с неравномерным узловым вектором $[0 \ 0 \ 0 \ 2 \ 2 \ 3 \ 3 \ 3]$ и базисными функциями на рис. 5-38e.

Из рис. 5-49 видно, что кратные внутренние узловые значения порождают излом в вершине B_3 . Кратное значение порождает ребро нулевой длины, поэтому уменьшается диапазон поддержки базисных функций. Далее, кратные внутренние узловые значения, в отличие от кратных вершин многоугольника, снижают дифференцируемость базисной функции в x_i до C^{k-m-1} , где $m \leq k-1$ равно кратности внутреннего узлового значения. Локально неравномерная кривая на рис. 5-49 C° ($k-m-1 = 3-2-1 = 0$) непрерывная в окрестности B_3 , что и приводит к появлению угла.

Открытые неравномерные В-сплайны третьего порядка ($k=3$) на рис. 5-50 построены с помощью узлового вектора с внутренними значениями, пропорциональными длинам ребер между вершинами многоугольника. Узловой вектор имеет вид

$$x_i = 0 \quad 1 \leq i \leq k,$$

$$x_{i+k} = \left(\frac{\left(\frac{i}{n-k+2} \right) c_{i+1} + \sum_{j=1}^i c_j}{\sum_{i=1}^n c_i} \right) (n-k+2), \quad 1 \leq i \leq n-k+1,$$

$$x_i = n-k+2, \quad n+k+1 \leq i \leq n+k, \quad (5-86)$$

где $c_i = |B_{i+1} - B_i|$. Для вершин, находящихся на одинаковом расстоянии друг от друга, узловой вектор имеет целые равномерно распределенные значения, т. е. вектор является открытым и равномерным. В работе [5-25] предложен по-

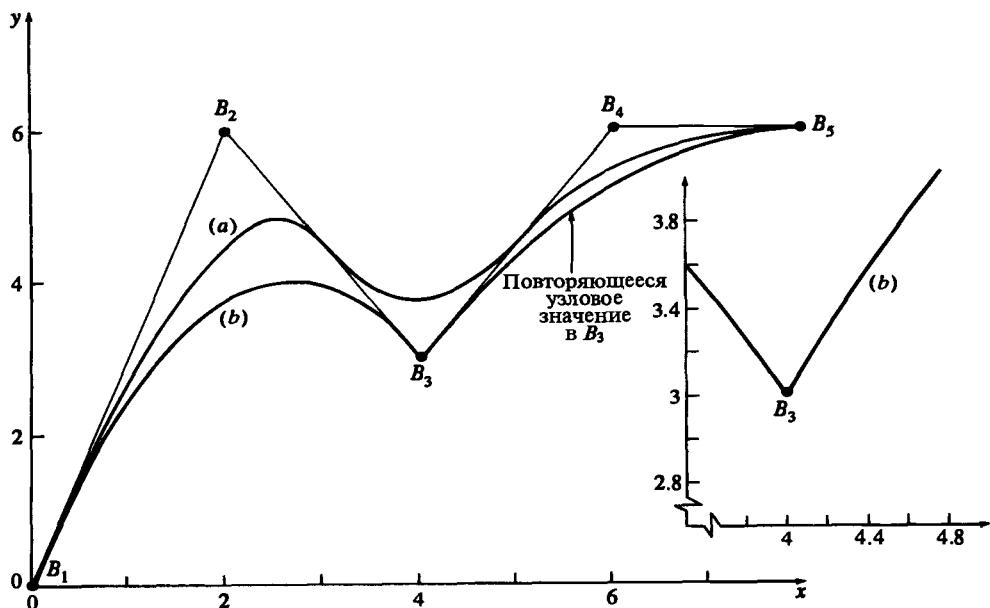


Рис. 5-49 Неравномерные В-сплайны, $k = 3$. (a) $[X] = [0 \ 0 \ 0 \ 1 \ 1 \ 3 \ 3 \ 3]$; (b) $[X] = [0 \ 0 \ 0 \ 1 \ 1 \ 3 \ 3 \ 3]$.

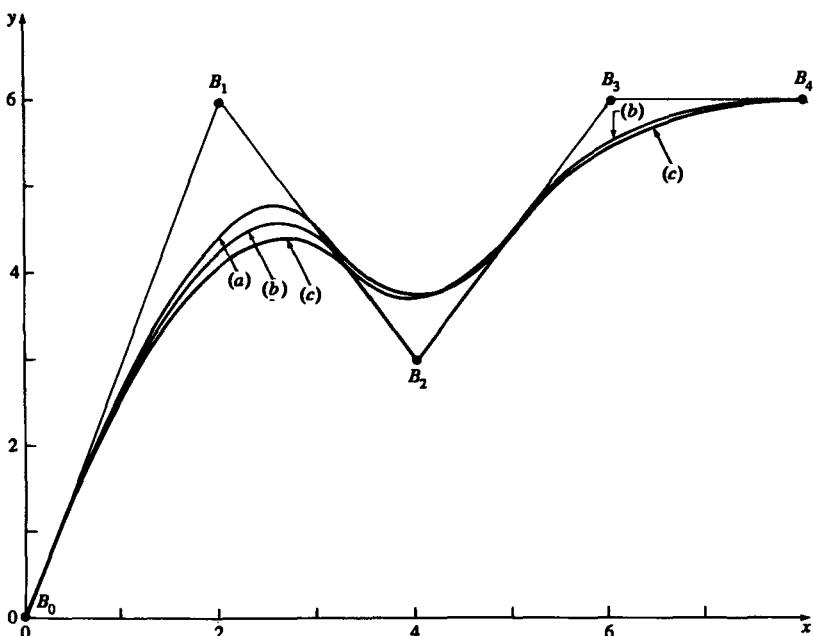


Рис. 5-50 Сравнение открытых неравномерных В-сплайнов: (a) равномерный узловый вектор; (b) неравномерный узловый вектор, пропорциональный длинам хорд; (c) неравномерный узловый вектор, пропорциональный длинам хорд, с двойной вершиной в B_2 .

хожий метод, позволяющий получить отдельные внутренние узловые значения. Для сравнения приведена кривая с открытым равномерным вектором, а также кривая с парой совпадающих вершин B_3 .

Отсюда следует, что неравномерные В-сплайны не сильно отличаются от равномерных при небольшом изменении относительного расстояния между вершинами. Рассмотрим пример.

Пример 5-14 Неравномерный В-сплайн

Найти открытый В-сплайн третьего порядка, заданный точками $B_1[0 \ 0]$, $B_2[2 \ 6]$, $B_3[4 \ 3]$, $B_4[6 \ 6]$, $B_5[8 \ 6]$ с помощью неравномерного вектора с узловыми значениями, пропорциональными длинам ребер многоугольника.

Сначала найдем длины хорд

$$c_1 = \sqrt{(2-0)^2 + (6-0)^2} = \sqrt{40} = 6.325$$

$$c_2 = \sqrt{(4-2)^2 + (3-6)^2} = \sqrt{13} = 3.606$$

$$c_3 = \sqrt{(6-4)^2 + (6-3)^2} = \sqrt{13} = 3.606$$

$$c_4 = \sqrt{(8-6)^2 + (6-6)^2} = \sqrt{4} = 2.0$$

Их суммарная длина

$$\sum_{i=1}^4 c_i = 15.537.$$

Из уравнения (5-86) получаем внутренние узловые значения

$$x_4 = \left[\frac{\frac{c_2}{3} + c_1}{15.537} \right] = 1.453,$$

$$x_5 = \left[\frac{\frac{2c_3}{3} + c_1 + c_2}{15.537} \right] (3) = 2.382.$$

Узловой вектор имеет вид

$$[x] = [0 \ 0 \ 0 \ 1.45 \ 2.382 \ 3 \ 3 \ 3]$$

где $x_1 = 0$, $x_2 = 0, \dots, x_8 = 3$. Диапазон параметра $0 \leq t \leq 3$. Кривая состоит из трех параболических ($k-1 = 2$) сегментов.

Функции базиса для $0 \leq t < 1.453$ таковы

$$N_{3,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 3$$

$$N_{2,2}(t) = \frac{(1.453-t)}{1.453}; \quad N_{3,2}(t) = \frac{t}{1.453}; \quad N_{i,2}(t) = 0, \quad i \neq 2, 3$$

$$N_{1,3}(t) = \frac{(1.453-t)^2}{(1.453)^2}; \quad N_{2,3}(t) = \frac{t}{(1.453)^2}(1.453-t) + \frac{(2.382-t)t}{(2.382)(1.453)};$$

$$N_{3,3}(t) = \frac{t^2}{(2.382)(1.453)}; \quad N_{i,3}(t) = 0, \quad i \neq 1, 2, 3$$

Функции базиса для $1.453 \leq t < 2.382$:

$$N_{4,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 4$$

$$N_{3,2}(t) = \frac{(2.382 - t)}{(2.382 - 1.453)};$$

$$N_{4,2}(t) = \frac{(t - 1.453)}{(2.382)(1.453)}; \quad N_{i,2}(t) = 0, \quad i \neq 3, 4$$

$$N_{2,3}(t) = \frac{(2.382 - t)^2}{2.382(2.382 - 1.453)};$$

$$N_{3,3}(t) = \frac{t}{2.382} \frac{(2.382 - t)}{(2.382 - 1.453)} + \frac{(3 - t)}{(3 - 1.453)} \frac{(t - 1.453)}{(2.382 - 1.453)};$$

$$N_{4,3}(t) = \frac{(t - 1.453)^2}{(3 - 1.453)(2.382 - 1.453)}; \quad N_{i,3}(t) = 0, \quad i \neq 2, 3, 4$$

Функции базиса для $2.382 \leq t < 3$:

$$N_{5,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 5$$

$$N_{4,2}(t) = \frac{(3 - t)}{(3 - 2.382)};$$

$$N_{5,2}(t) = \frac{(t - 2.382)}{(3 - 2.382)}; \quad N_{i,2}(t) = 0, \quad i \neq 4, 5$$

$$N_{3,3}(t) = \frac{(3 - t)^2}{(3 - 1.453)(3 - 2.382)};$$

$$N_{4,3}(t) = \frac{(t - 1.453)(3 - t)}{(3 - 1.453)(3 - 2.382)} + \frac{(3 - t)(t - 2.382)}{(3 - 2.382)^2};$$

$$N_{5,3}(t) = \frac{(t - 2.382)^2}{(3 - 2.382)^2}; \quad N_{i,3}(t) = 0, \quad i \neq 3, 4, 5$$

Вспомним, что любая точка кривой задается уравнением:

$$P(t) = N_{1,3}(t)B_1 + N_{2,3}(t)B_2 + N_{3,3}(t)B_3 + N_{4,3}(t)B_4 + N_{5,3}(t)B_5$$

т.е. при $t = 1/2$

$$\begin{aligned} P\left(\frac{1}{2}\right) &= 0.430B_1 + 0.498B_2 + 0.072B_3 + (0)B_4 + (0)B_5 \\ &= 0.430[0 \quad 0] + 0.498[2 \quad 6] + 0.072[4 \quad 3] \\ &= [1.284 \quad 3.202] \end{aligned}$$

и при $t = 2$

$$\begin{aligned} P(2) &= (0)B_1 + 0.066B_2 + 0.726B_3 + 0.208B_4 + (0)B_5 \\ &= 0.066[2 \quad 6] + 0.726[4 \quad 3] + 0.208[6 \quad 6] \\ &= [4.284 \quad 3.822] \end{aligned}$$

Результаты приведены на рис. 5-50.

Уравнения B-сплайна можно записать в матричной форме, подобно кубическим сплайнам, параболической интерполяции и кривым Безье (см. уравнения (5-276), (5-44) и (5-67)). У периодических B-сплайнов матричная форма особенно проста.

Вспомним, что базисные функции периодического B-сплайна являются параллельным переносом друг друга (см. рис. 5-36) и что влияние каждой функции распространяется только на k интервале. Поэтому для целых узловых значений все базисные функции на единичном интервале $0 \leq t^* < 1$ имеют один и тот же вид $N_{i,k}^*(t^*)$, т. е. иногда удобно нормализовать параметризацию. Точка на параметрическом B-сплайне на единичном интервале имеет вид

$$P_j(t^*) = \sum_{i=0}^{k-1} N_{i+1,k}^*(t^*) B_{j+i}, \quad 1 \leq j \leq n - k + 1, \quad 0 \leq t^* < 1, \quad (5-87)$$

где j — счетчик сегментов кривой, а n на единицу меньше количества вершин. Заметим, что уравнение (5-87) содержит только k членов:

$$P_j(t^*) = N_{1,k}^* B_j + N_{2,k}^* B_{j+1} + \cdots + N_{k,k}^* B_{j+k-1}.$$

Для $k = 3$ базисные функции на интервале параметра $0 \leq t^* < 1$

$$\begin{aligned} N_{1,3}^*(t^*) &= \frac{(1-t^*)^2}{2} \\ N_{2,3}^*(t^*) &= \frac{-2t^{*2} + 2t^* + 1}{2} \\ N_{3,3}^*(t^*) &= \frac{t^{*2}}{2} \end{aligned}$$

Тогда уравнение (5-87) принимает вид

$$\begin{aligned} 2P_j(t^*) &= (1 - 2t^* + t^{*2})B_j + (-2t^{*2} + 2t^* + 1)B_{j+1} + t^{*2}B_{j+2} = \\ &= \begin{array}{c} t^{*2} \\ + t^* \\ + \end{array} \left(\begin{array}{ccc} B_j & -2B_{j+1} & B_{j+2} \\ -2B_j & 2B_{j+1} & 0 \cdot B_{j+2} \\ B_j & B_{j+1} & 0 \cdot B_{j+2} \end{array} \right) \end{aligned}$$

Перепишем в матричной форме:

$$\begin{aligned} P_j^*(t^*) &= [T^*][N^*][G] = \\ &= \frac{1}{2} [t^{*2} \quad t^* \quad 1] \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} B_j \\ B_{j+1} \\ B_{j+2} \end{bmatrix} \end{aligned} \quad (5-88)$$

Аналогично, для $k = 4$ базисные функции на интервале параметра $0 \leq t^* < 1$ таковы:

$$\begin{aligned} N_{1,4}^*(t^*) &= \frac{-t^{*3} + 3t^{*2} - 3t^* + 1}{6} \\ N_{2,4}^*(t^*) &= \frac{3t^{*3} + 6t^{*2} + 4}{6} \\ N_{3,4}^*(t^*) &= \frac{-t^{*3} + 3t^{*2} + 3t^* + 1}{6} \\ N_{4,4}^*(t^*) &= \frac{t^{*3}}{6} \end{aligned}$$

Матричная форма для $k = 4$ имеет вид

$$\begin{aligned} P_j(t^*) &= [T^*][N^*][G] = \\ &= \frac{1}{6}[t^{*3} \quad t^{*2} \quad t^* \quad 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} B_j \\ B_{j+1} \\ B_{j+2} \\ B_{j+3} \end{bmatrix} \end{aligned} \quad (5-89)$$

Уравнения (5-87) и (5-89) содержат произведение весовой функции и геометрической матрицы (см. уравнения (5-27), (5-44), (5-67); т. е.

$$P_j(t^*) = [T^*][N^*][G] = [F][G] \quad 1 \leq j \leq n - k + 1, \quad 0 \leq t^* < 1 \quad (5-90)$$

где

$$[T^*] = [t^{*k-1} \quad t^{*k-2} \quad \dots \quad t^* \quad 1] \quad 0 \leq t^* < 1$$

$[G]^T = [B_j \quad \dots \quad B_{j+k-1}]$ содержит геометрию — вершины многоугольника, а $[F]$ состоит из соответствующих базисных функций.

В работе [5-19] показано, что для периодических В-сплайнов общий вид $[N]$:

$$[N^*] = [N_{i+1,j+1}^*], \quad (5-91)$$

где

$$\begin{aligned} N_{i+1,j+1}^* &= \frac{1}{(k-1)!} n_{i+1,j+1}^* \\ &= \frac{1}{(k-1)!} \binom{k-1}{i} \sum_{l=j}^{k-1} (k-(l+1))^i (-1)^{l-j} \binom{k}{l-j}, \quad 0 \leq i, \quad j \leq k-1. \end{aligned}$$

Вспомним, что у периодических замкнутых В-сплайнов необходимо повторить некоторые вершины, чтобы ликвидировать разрыв. В матричном виде это очень просто. Замкнутый периодический сплайн имеет вид

$$P_{j+1}(t^*) = \sum_{i=0}^{k-1} N_{i+1,k}^*(t^*) B_{((j+i) \bmod (n+1)) + 1}, \quad 0 \leq j \leq n. \quad (5-92)$$

Или в матричном виде

$$P_{j+1}(t^*) = [T^*][N^*] \begin{bmatrix} B_{j \mod (n+1)+1} \\ B_{((j+1) \mod (n+1))+1} \\ \vdots \\ \vdots \\ B_{((j+1+n-k) \mod (n+1))+1} \end{bmatrix}, \quad (5-93)$$

где $[T^*]$ и $[N^*]$ взяты из уравнения (5-90), а mod — остаток; например, $3 \mod 2 = 1$.

Открытые В-сплайны также можно записать в матричном виде, однако из-за кратных узловых значений на концах это не так удобно, как для периодических. В общем случае матричный вид открытого В-сплайна с целыми узловыми значениями таков:

$$P(t) = [F][G] = [T][N][G], \quad (5-94)$$

где

$$[G]^T = [B_1 \quad \dots \quad B_{n+1}],$$

а элементы $[F]$ и $[N]$ получены по алгоритму Кокса–де Бура (уравнение (5-84)) для каждого ненулевого интервала вектора $[X]$. У В-сплайнов низкого порядка, заданных большим количеством точек, большинство членов $[F]$ равны нулю. Используя это свойство, можно значительно сэкономить вычисления.¹

В работе [5-19] приведена обобщенная формулировка для $[N^*]$ на нормализованном интервале $0 \leq t^* < 1$; однако из-за кратности концевых узловых значений первые и последние $k - 1$ N^* матриц являются частными случаями.

Рассмотрим пример матричного метода.

Пример 5-15 Расчет замкнутого В-сплайна

Найти замкнутый В-сплайн четвертого порядка ($k = 4$), заданный многоугольником, изображенным на рис. 5-46. Записать результат в матричном виде. Вершины многоугольника: $B_1[2 \ 0]$, $B_2[4 \ 0]$, $B_3[4 \ 2]$, $B_4[4 \ 4]$, $B_5[2 \ 4]$, $B_6[0 \ 4]$, $B_7[0 \ 2]$, $B_8[0 \ 0]$, $B_9[2 \ 0]$, $n = 8$.

Для каждого единичного интервала $0 \leq t^* < 1$ из уравнений (5-89) и (5-93) получим

$$P_{j+1}(t^*) = \frac{1}{6}[t^{*3} \quad t^{*2} \quad t^* \quad 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} B_{(j \mod 8)+1} \\ B_{((j+1) \mod 8)+1} \\ B_{((j+2) \mod 8)+1} \\ B_{((j+3) \mod 8)+1} \end{bmatrix}.$$

¹ См. алгоритм из Приложения G.

При $t^* = 1/2$ на первом сегменте ($j = 0$) кривой

$$\begin{aligned} P_1\left(\frac{1}{2}\right) &= \frac{1}{6} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \\ &= \frac{1}{48} [1 \quad 23 \quad 23 \quad 1] \begin{bmatrix} 2 & 0 \\ 4 & 0 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} = \\ &= \frac{1}{48} [190 \quad 50] = [3.958 \quad 1.042]. \end{aligned}$$

При $t^* = 1/2$ на восьмом сегменте ($j = 7$) кривой

$$\begin{aligned} P_8\left(\frac{1}{2}\right) &= \frac{1}{48} [1 \quad 23 \quad 23 \quad 1] \begin{bmatrix} B_8 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix} = \\ &= \frac{1}{48} [1 \quad 23 \quad 23 \quad 1] \begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 4 & 0 \\ 4 & 2 \end{bmatrix} = \\ &= \frac{1}{48} [142 \quad 2] = [2.958 \quad 0.417]. \end{aligned}$$

Результат показан на рис. 5-46б.

Производная в любой точке В-сплайна вычисляется формальным дифференцированием. Из уравнения (5-83)

$$P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t) \quad (5-83)$$

первая производная:

$$P'(t) = \sum_{i=1}^{n+1} B_i N'_{i,k}(t) \quad (5-95)$$

и вторая производная:

$$P''(t) = \sum_{i=1}^{n+1} B_i N''_{i,k}(t). \quad (5-96)$$

Штрихи обозначают дифференцирование по параметру t .

Производные базисных функций также получают формальным дифференцированием. После однократного дифференцирования уравнения (5-84) имеем

$$N'_{i,k}(t) = \frac{N_{i,k-1}(t) + (t - x_i) N'_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t) N'_{i+1,k-1}(t) - N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}. \quad (5-97)$$

Из уравнения (5-84а) следует, что $N'_{i,1}(t) = 0$ для всех t . Следовательно, для $k = 2$ уравнение (5-97) принимает вид

$$N'_{i,2}(t) = \frac{N_{i,1}(t)}{x_{i+k-1} - x_i} - \frac{N_{i+1,1}(t)}{x_{i+k} - x_{i+1}}. \quad (5-98)$$

Вторые производные базисных функций получаем, дифференцируя уравнение (5-97),

$$\begin{aligned} N''_{i,k}(t) = & \frac{2N'_{i,k-1}(t) + (t - x_i)N''_{i,k-1}(t)}{x_{i+k-1} - x_i} + \\ & + \frac{(x_{i+k} - t)N''_{i+1,k-1}(t) - 2N'_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}. \end{aligned} \quad (5-99)$$

Здесь как $N'_{i,1}(t) = 0$, так и $N'_{i,2}(t) = 0$ равны нулю для всех t . Следовательно, для $k = 3$ уравнение (5-97) имеет вид

$$N''_{i,3}(t) = 2 \left(\frac{N'_{i,2}(t)}{x_{i+k-1} - x_i} - \frac{N'_{i+1,2}(t)}{x_{i+k} - x_{i+1}} \right). \quad (5-100)$$

На рис. 5-51 изображены несколько базисных функций В-сплайна и их производные для $k = 4$. Отметим, что при $k = 4$ каждая базисная функция — кусочно кубическая, первая производная — кусочно параболическая, вторая производная — кусочно линейная. Третья производная разрывна и состоит из различных констант.

Рассмотрим пример.

Пример 5-16 Расчет производных В-сплайна

Рассмотрим многоугольник из примера 5-12: $B_1[1 \ 1]$, $B_2[2 \ 3]$, $B_3[4 \ 3]$, $B_4[3 \ 1]$ и соответствующий сплайн. Найти первую производную кривой второго порядка ($k = 2$).

Для $k = 2$ открытый узловой вектор имеет вид:

$$[0 \ 0 \ 1 \ 2 \ 3 \ 3],$$

где $x_1 = 0$, $x_2 = 0, \dots, x_6 = 3$. Диапазон параметра $0 \leq t \leq 3$. Из уравнения (5-95) получаем первую производную для $n = 3$, $k = 2$

$$P'(t) = B_1 N'_{1,2}(t) + B_2 N'_{2,2}(t) + B_3 N'_{3,2}(t) + B_4 N'_{4,2}(t).$$

Из уравнения (5-98) следует, что

$$N'_{i,2}(t) = \frac{N_{i,1}(t)}{x_{i+k-1} - x_i} - \frac{N_{i+1,1}(t)}{x_{i+k} - x_{i+1}}.$$

Пользуясь результатами уравнения (5-12), для $0 \leq t < 1$ получим

$$N_{2,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 2,$$

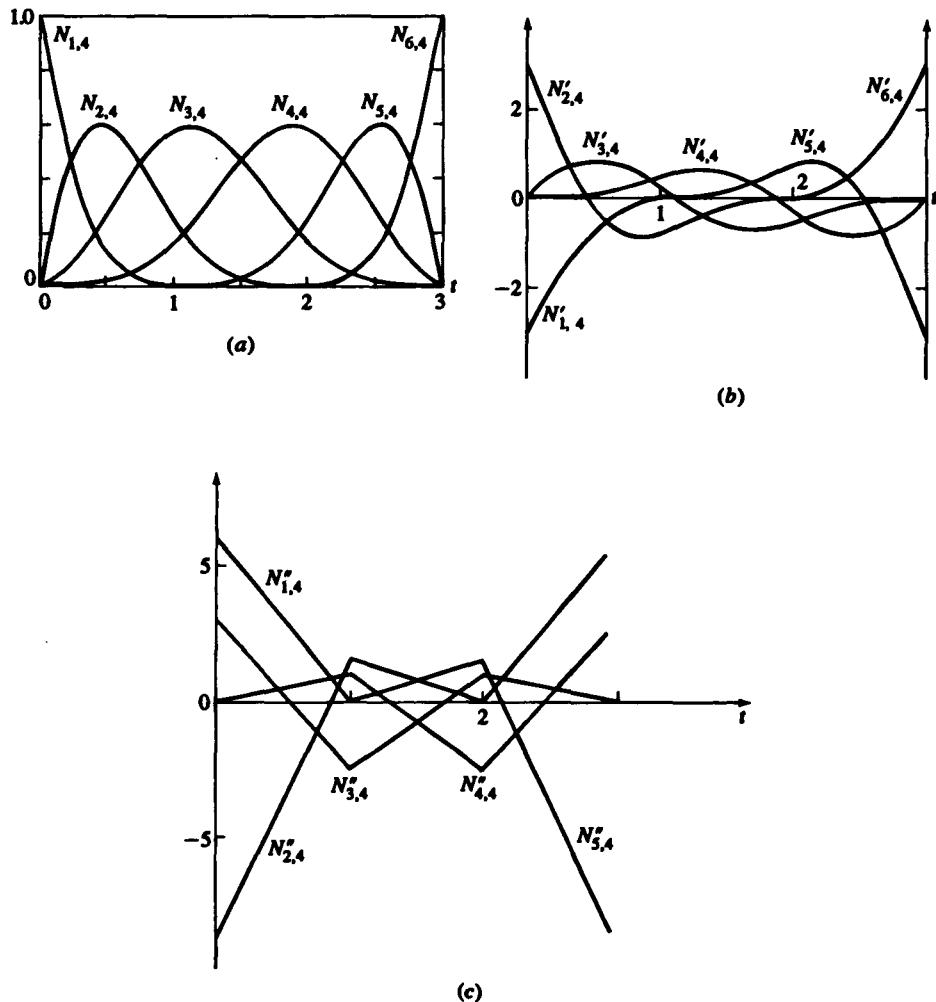


Рис. 5-51 Базисные функции В-сплайна и их первые и вторые производные, $k = 4$, $n = 6$. (a) Базисные функции; (b) первая производная; (c) вторая производная.

и

$$N'_{1,2}(t) = -1; \quad N'_{2,2}(t) = 1; \quad N'_{i,2}(t) = 0, \quad i \neq 2.$$

Итак

$$P'(t) = B_2 - B_1,$$

что является наклоном (касательным вектором) первого ребра многоугольника, как оно и должно быть. Для $1 \leq t < 2$

$$N_{3,1}(t) = 1; \quad N_{i,1}(t) = 0 \quad i \neq 3$$

и

$$N'_{2,2}(t) = -1; \quad N'_{3,2}(t) = 1; \quad N'_{i,2}(t) = 0 \quad i \neq 2, 3$$

т. е.

$$P'(t) = B_3 - B_2,$$

что равно наклону (касательному вектору) второго ребра.

Наконец, для $2 \leq t < 3$

$$N_{4,1}(t) = 1; \quad N_{i,1}(t) = 0 \quad i \neq 4$$

и

$$N'_{3,2}(t) = -1; \quad N'_{4,2}(t) = 1; \quad N'_{i,2}(t) = 0 \quad i \neq 3, 4,$$

т. е.

$$P'(t) = B_4 - B_3$$

— наклон последнего ребра.

При $t = 3$, $N_{i,1}(3) = 0$ для всех i . Следовательно, формально $P'(3) = 0$, но предел $t = 3 - \varepsilon$, при $\varepsilon \rightarrow 0$ показывает, что на самом деле $P'(3) = (B_4 - B_3)$.

5-10 КОНЕЧНЫЕ УСЛОВИЯ ДЛЯ ПЕРИОДИЧЕСКИХ В-СПЛАЙНОВ

Как было показано ранее, конец и начало периодического В-сплайна не совпадают с первой и последней вершинами определяющего многоугольника. Отсюда возникают два вопроса. Во-первых, где же именно начинается и заканчивается В-сплайн, и каковы его граничные условия, т. е. производные в этих точках? Во-вторых, что влияет на конечные точки и производные в них? В работе [5-24] изучается этот вопрос для частного случая кубических ($k = 4$) В-сплайнов. Мы рассмотрим более общий случай.

Произвольный периодический В-сплайн начинается в точке $P_s = P(t = x_k)$ и оканчивается в точке $P_e = P(t = x_{n+1})$. Для целых узловых значений, начинаяющихся с нуля, $P_s = P(t = k)$ и $P_e = P(t = n)$. Из уравнений (5-87) и (5-91), а также из того, что любая точка В-сплайна зависит от k ближайших вершин, следует, что для преобразованного параметра $0 \leq t^* < 1$ начальная точка соответствует $t^* = 0$. Итак

$$P_s = \frac{1}{(k-1)!} (N_{k,1}^* B_1 + N_{k,2}^* B_2 + \cdots + N_{k,k}^* B_k).$$

Заметим, что $N_{k,k}^* = 0$ для всех k

$$P_s = \frac{1}{(k-1)!} (N_{k,1}^* B_1 + N_{k,2}^* B_2 + \cdots + N_{k,k-1}^* B_{k-1}), \quad n \geq k. \quad (5-101)$$

В конце интервала $t^* = 1$. Известно, что $\sum_{i=1}^k N_{i,1}^* = 0$, поэтому конечная точка

$$P_e = \frac{1}{(k-1)!} \left(\sum_{i=1}^k N_{i,2}^* B_{n-k+3} + \sum_{i=1}^k N_{i,3}^* B_{n-k+4} + \cdots + \sum_{i=1}^k N_{i,k}^* B_{n+1} \right) \quad n \geq k. \quad (5-102)$$

Для квадратичного ($k = 3$) периодического сплайна уравнения (5-101) и (5-102) принимают вид

$$\begin{aligned} P_s &= \frac{1}{2}(B_1 + B_2), \\ P_e &= \frac{1}{2}(B_n + B_{n+1}). \end{aligned}$$

Следовательно, квадратичный периодический В-сплайн начинается и кончается, соответственно, в середине первого и последнего ребер многоугольника.

Конечные точки кубического ($k = 4$) периодического сплайна таковы:

$$\begin{aligned} P_s &= \frac{1}{6}(B_1 + 4B_2 + B_3), \\ B_e &= \frac{1}{6}(B_{n-1} + 4B_n + B_{n+1}). \end{aligned}$$

Из уравнения (5-90) первая производная в начальной точке

$$\begin{aligned} P'_s &= [T^{**}]_{t^*=0} [N^*][G] = \\ &= \frac{1}{(k-1)!} (N_{k-1,1}^* B_1 + N_{k-1,2}^* B_2 + \cdots + N_{k-1,k-1}^* B_{k-1}) \end{aligned} \quad (5-103)$$

так как $N_{k-1,k}^* = 0$ для всех k . Первая производная в конечной точке

$$\begin{aligned} P'_e &= [T^{**}]_{t^*=1} [N^*][G] = \\ &= \frac{1}{(k-1)!} \left(\sum_{i=1}^{k-1} (k-i) N_{i,2}^* B_{n-k+3} + \right. \\ &\quad \left. + \sum_{i=1}^{k-1} (k-i) N_{i,3}^* B_{n-k+4} + \cdots + \sum_{i=1}^{k-1} (k-i) N_{i,k}^* B_{n+1} \right), \end{aligned} \quad (5-104)$$

где $[T^{**}] = [(k-1)t^{*(k-2)} \quad (k-2)t^{*(k-3)} \quad \dots \quad 1 \quad 0]$ — производная вектора параметра.

В случае квадратичных ($k = 3$) периодических В-сплайнов эти результаты сводятся к

$$\begin{aligned} P'_s &= \frac{1}{2}(-2B_1 + 2B_2) = B_2 - B_1 \\ P'_e &= \frac{1}{2}(-2B_n + 2B_{n+1}) = B_{n+1} - B_n, \end{aligned}$$

т. е. векторам касательных (наклону) первого и последнего ребер.
Для кубических ($k = 4$) кривых

$$P'_s = \frac{1}{6}(-3B_1 + 3B_3) = \frac{1}{2}(B_3 - B_1)$$

$$P'_e = \frac{1}{6}(-3B_{n-2} + 3B_{n+1}) = \frac{1}{2}(B_{n+1} - B_{n-1}).$$

Здесь касательные векторы (наклоны) зависят от касательных к трем первым и к трем последним сегментам кривой.

Вторые производные в конечных точках:

$$P''_s = [T^{**}]_{t^*=0}[N^*][G] =$$

$$= \frac{2}{(k-1)!}(N_{k-2,1}^*B_1 + N_{k-2,2}^*B_2 + \dots + N_{k-2,k-1}^*B_{k-1}) \quad (5-105)$$

$$P''_e = [T^{**}]_{t^*=1}[N^*][G] =$$

$$= \frac{1}{(k-1)!} \left(\sum_{i=1}^{k-2} (k-i)(k-i-1)N_{i,2}^*B_{n-k+3} + \right.$$

$$+ \sum_{i=1}^{k-2} (k-i)(k-i-1)N_{i,3}^*B_{n-k+4} +$$

$$\left. + \dots + \sum_{i=1}^{k-2} (k-i)(k-i-1)N_{i,k}^*B_{n+1} \right) \quad (5-106)$$

где $[T^{**}] = [(k-1)t^{*(k-2)} \quad (k-2)t^{*(k-3)} \quad \dots \quad 1 \quad 0]$ — вторая производная вектора параметра.

Для кубических ($k = 4$) сплайнов уравнения (5-105) и (5-106) дают

$$P''_s = \frac{1}{6}(6B_1 - 12B_2 + 6B_3) = B_1 - 2B_2 + B_3,$$

$$P''_e = \frac{1}{6}(6B_{n-1} - 12B_n + 6B_{n+1}) = B_{n-1} - 2B_n + B_{n+1}.$$

Существует два метода регулирования положения первой и последней точек, а также конечных условий: кратные вершины и псевдовершины.

Кратные вершины на концах периодического B-сплайна притягивают концы сплайна к соответствующим вершинам. Если определены $k-1$ совпадающая вершина, то конечные точки кривой совпадают с вершинами многоугольника и касательные (наклон) совпадают с направлением соседних ненулевых ребер.

Например, для $k = 3$ с двойными вершинами на концах, т. е. $B_1 = B_2$ и $B_n = B_{n+1}$ из уравнений (5-101) и (5-102) следует

$$P_s = \frac{1}{2}(B_1 + B_2) = \frac{1}{2}(2B_1) = B_1,$$

$$P_e = \frac{1}{2}(B_n + B_{n+1}) = \frac{1}{2}(2B_{n+1}) = B_{n+1}.$$

Для $k = 4$ с двойными вершинами на концах эти же уравнения дают

$$P_s = \frac{1}{6}(5B_1 + B_3),$$

$$P_e = \frac{1}{6}(B_{n-1} + 5B_{n+1}).$$

Отсюда начальная точка находится на расстоянии от одной шестой ребра, соединяющего точки B_1 и B_3 , а конечная точка находится на расстоянии в пять шестых ребра от B_{n-1} до B_{n+1} .

Если в концах располагаются по три кратные вершины, т. е. $B_1 = B_2 = B_3$ и $B_{n-1} = B_n = B_{n+1}$ для $k = 4$, то

$$P_s = \frac{1}{6}(B_1 + 4B_1 + B_1) = B_1,$$

$$P_e = \frac{1}{6}(B_{n+1} + 4B_{n+1} + B_{n+1}) = B_{n+1}.$$

Кривая начинается и кончается в первой и последней вершинах многоугольника. На рис. 5-52 изображено влияние кратных вершин.

С тремя кратными вершинами на концах первый и последний сегменты В-сплайна для $k = 4$ (см. уравнение 5-89) имеют вид:

$$P_1(t^*) = B_1 + \frac{t^{*3}}{6}[B_4 - B_1] = B_3 + \frac{t^{*3}}{6}[B_4 - B_3] \quad 0 \leq t^* < 0$$

и

$$P_n(t^*) = B_{n+1} + \frac{(1-t^{*3})}{6}(B_n - B_{n+1}).$$

При этом первый и последний сегменты линейны. Первый сегмент на одну шестую длины ребра от $B_1 = B_2 = B_3$ до B_4 совпадает с этим ребром. Последний сегмент на одну шестую ребра от B_{n-2} до $B_{n-1} = B_n = B_{n+1}$ совпадает с соответствующим ребром.

Несмотря на то, что участки можно сделать как угодно малыми, в некоторых прикладных областях это может быть неудобным. В таких случаях лучше пользоваться открытыми В-сплайнами.

Псевдовершины в концах периодического В-сплайна, в отличие от кратных вершин, позволяют управлять как положением конечных точек, так и граничными условиями. В общем случае псевдовершины не обозначаются, и пользователь не может контролировать их. Как показано на рис. 5-53, B_0 и B_{n+2} — псевдовершины в начале и конце В-сплайна. В этих обозначениях уравнения (5-101) и (5-102) принимают вид

$$P_s = \frac{1}{(k-1)!}(N_{k,1}^* B_0 + N_{k,2}^* B_1 + \cdots + N_{k,k-1}^* B_{k-2}) \quad n \geq k \quad (5-107)$$

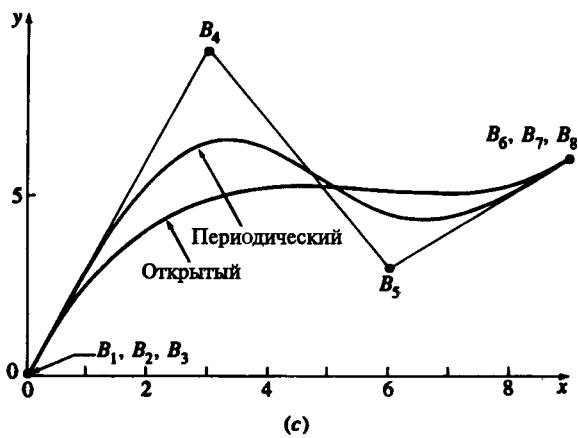
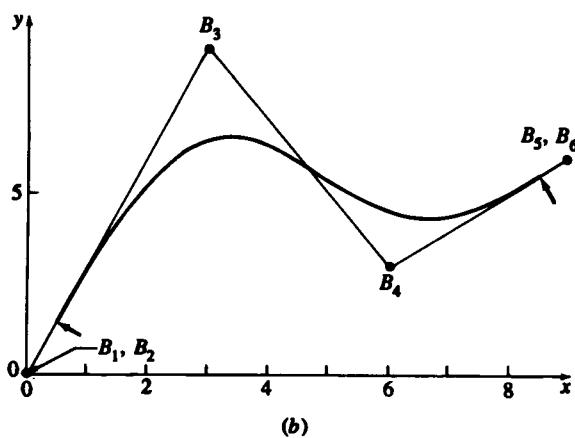
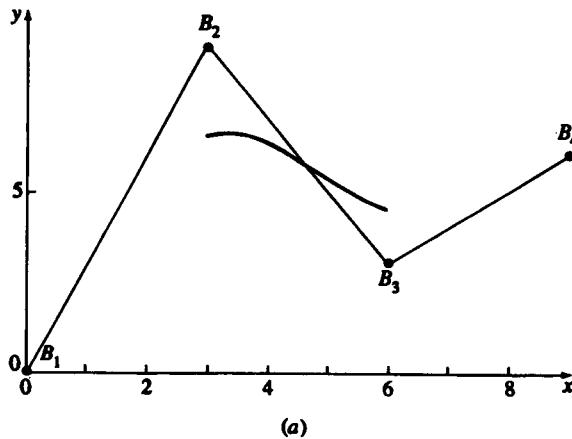


Рис. 5-52 Совпадающие кратные вершины на концах определяющего многоугольника ($k = 4$). (a) Кратных вершин нет; (b) две кратные вершины; (c) три кратные вершины и соответствующие открытые B-сплайны.

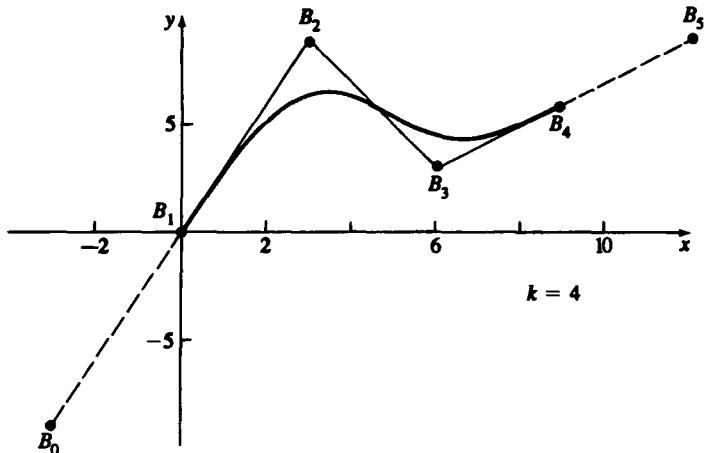


Рис. 5-53 Псевдовершины определяют начальную и конечную точки периодического B-сплайна.

и

$$P_e = \frac{1}{(k-1)!} \left(\sum_{i=1}^k N_{i,2}^* B_{n-k+4} + \sum_{i=1}^k N_{i,3}^* B_{n-k+5} + \cdots + \sum_{i=1}^k N_{i,k}^* B_{n+2} \right) \quad n \geq k \quad (5-108)$$

При $P_s = B_1$ и $P_e = B_{n+1}$ эти уравнения имеют вид

$$B_0 = ((k-1)! - N_{k,2}^*) B_1 = (N_{k,3}^* B_2 + \cdots + N_{k,k-1}^* B_{k-2}) \quad n \geq k \quad (5-109)$$

и

$$\begin{aligned} B_{n+2} &= \left((k-1)! - \sum_{i=1}^k N_{i,k-1}^* \right) B_{n+1} - \\ &\quad - \left(\sum_{i=1}^k N_{i,2}^* B_{n-k+4} + \cdots + \sum_{i=1}^k N_{i,k-2}^* B_n \right) \quad n \geq k \end{aligned} \quad (5-110)$$

где $N_{k,1}^* = 1$ и $\sum_{i=1}^k N_{i,k}^* = 1$.

Для $k = 3$ из уравнений (5-109) и (5-110) следует, что $B_0 = B_1$ и $B_{n+2} = B_{n+1}$, т. е. двукратные вершины на концах!

Для $k = 4$ аналогично

$$B_0 = (6-4)B_1 - B_2 = 2B_1 - B_2$$

и

$$B_{n+2} = (6-4)B_{n+1} - B_n = 2B_{n+1} - B_n.$$

Пример изображен на рис. 5-53.

Первая и вторая производные в концах выражены уравнениями (5-103) и (5-104) с соответствующими изменениями для учета псевдовершин. Например, используя приведенные результаты для B_0 , из уравнений (5-103) и (5-104) для $k = 4$ получаем

$$P'_s = \frac{1}{2}(B_2 - B_0) = \frac{1}{2}\{B_2 - (2B_1 - B_2)\} = B_2 - B_1$$

и

$$P'_e = \frac{1}{2}(B_{n+2} - B_n) = \frac{1}{2}\{2B_{n+1} - B_n - B_n\} = B_{n+1} - B_n.$$

Итак, кривая касательна к первому и последнему ребрам многоугольника.

Аналогично, для $k = 4$ и точек B_0 и B_{n+2} из уравнений (5-105) и (5-106) следует

$$P''_s = B_0 - 2B_1 + B_2 = 2B_1 - B_2 - 2B_1 + B_2 = 0,$$

$$P''_e = B_n + 2B_{n+1} + B_{n+2} = B_n - 2B_{n+1} + 2B_{n+1} - B_n = 0,$$

т. е. «кривизна» в концах нулевая.

Переписав уравнения (5-103) и (5-104), можно определить псевдовершины, порождающие заданные граничные условия. В частности, из уравнения (5-103)

$$B_0 = \frac{1}{N_{k-1,1}^*} \left\{ (k-1)! P'_s - (N_{k-1,2}^* B_1 + \cdots + N_{k-1,k-1}^* B_{k-2}) \right\} \quad n \geq k \quad (5-111)$$

а из уравнения (5-104)

$$B_{n+2} = \frac{1}{\sum_{i=1}^{k-1} (k-1) N_{i,k}^*} \left\{ (k-1)! P'_e - \left(\sum_{i=1}^{k-1} (k-1) N_{i,2}^* B_{n-k+4} + \cdots + \sum_{i=1}^{k-1} (k-1) N_{i,k-1}^* B_{n+1} \right) \right\} \quad n \geq k \quad (5-112)$$

Для $k = 4$ уравнения (5-111) и (5-112) сводятся к

$$B_0 = B_2 - 2P'_s,$$

$$B_{n+2} = 2P'_e + B_n.$$

Типичный пример изображен на рис. 5-54. Конечные точки кривой получены подстановкой этих значений в уравнения (5-107) и (5-108). Из рисунка видно, что первая и последняя точки кривой с заданными касательными на концах для многоугольника $B_0 B_1 B_2 B_3 B_4 B_5$ не совпадают с соответствующими точками кривой для многоугольника $B_1 B_2 B_3 B_4$ (отмечено крестиками).

Для управления второй производной или приблизительной кривизной в концах кривой перепишем уравнения (5-105) и (5-106) в виде

$$B_0 = \frac{1}{N_{k-2,1}^*} \left\{ \frac{(k-1)!}{2} P''_s - (N_{k-2,2}^* B_1 + \cdots + N_{k-2,k-1}^* B_{k-2}) \right\}, \quad n \geq k, \quad (5-113)$$

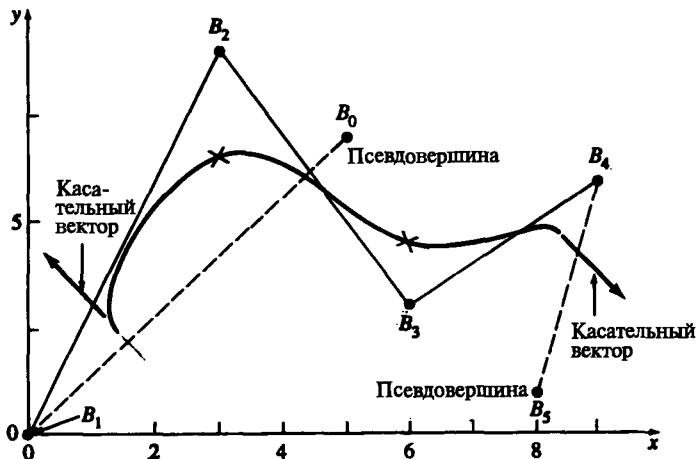


Рис. 5-54 Управление касательным вектором для периодического B-сплайна, $k = 4$.

и

$$B_{n+2} = \frac{1}{\sum_{i=1}^{k-1} (k-1)(k-i-1) N_{i,k}^*} \left\{ (k-1)! P_e'' - \right. \\ \left. - \left(\sum_{i=1}^{k-2} (k-1)(k-i-1) N_{i,2}^* B_{n-k+4} + \right. \right. \\ \left. \left. + \cdots + \sum_{i=1}^{k-2} (k-1)(k-i-1) N_{i,k-1}^* B_{n+1} \right) \right\} \quad n \geq k \quad (5-114)$$

Для $k = 4$ из уравнений (5-113) и (5-114) следует

$$B_0 = B_s'' + 2B_1 - B_2, \\ B_{n+2} = P_e'' + 2B_{n+1} - B_n.$$

Чтобы получить конечные точки, надо подставить эти значения в уравнения (5-107) и (5-108). Касательные векторы получаются аналогично, с помощью уравнений (5-103) и (5-104), переписанных относительно B_0 и B_{n+2} .

5-11 ПОДГОНКА В-СПЛАЙНОВ

В предыдущем разделе рассматривалось построение B-сплайна по определяющему многоугольнику. Сейчас нам нужно найти многоугольник, порождающий B-сплайн, для заданного множества точек кривой, как показано на рис. 5-55.

Точка, лежащая в В-сплайне, должна удовлетворять уравнению (5-83). Запишем уравнение (5-83) для всех j заданных точек

$$\begin{aligned} D_1(t_1) &= N_{1,k}(t_1)B_1 + N_{2,k}(t_1)B_2 + \cdots + N_{n+1,k}(t_1)B_{n+1}, \\ D_2(t_2) &= N_{1,k}(t_2)B_1 + N_{2,k}(t_2)B_2 + \cdots + N_{n+1,k}(t_2)B_{n+1}, \\ &\vdots \\ D_j(t_j) &= N_{1,k}(t_j)B_1 + N_{2,k}(t_j)B_2 + \cdots + N_{n+1,k}(t_j)B_{n+1}, \end{aligned}$$

где $2 \leq k \leq n+1 \leq j$, или в матричной форме

$$[D] = [N][B] \quad (5-115)$$

где

$$\begin{aligned} [D]^T &= [D_1(t_1) \quad D_2(t_2) \quad \cdots \quad D_j(t_j)] \\ [B]^T &= [B_1 \quad B_2 \quad \cdots \quad B_{n+1}] \\ [N] &= \left[\begin{array}{cccc} N_{1,k} & \cdots & \cdots & N_{n+1,k}(t_1) \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ N_{1,k}(t_j) & \cdots & \cdots & N_{n+1,k}(t_j) \end{array} \right] \end{aligned}$$

При $2 \leq k \leq n+1 = j$ матрица $[N]$ квадратная, и определяющий многоугольник вычисляется непосредственно обращением матрицы, т. е.

$$[B] = [N]^{-1}[D] \quad 2 \leq k \leq n+1 = j. \quad (5-116)$$

В этом случае В-сплайн проходит через все заданные точки, т. е. достигается точная подгонка. Кривая на всем протяжении C^{k-2} непрерывна, но при этом она может быть недостаточно гладкой на вид и иметь нежелательные изгибы.

Если указать меньшее число вершин многоугольника, чем количество заданных точек, т. е. $2 \leq k \leq n+1 < j$, кривая будет более гладкой. При этом матрица $[N]$ становится неквадратной, и можно найти только общее решение. Воспользуемся тем, что произведение матрицы на транспонированную всегда дает

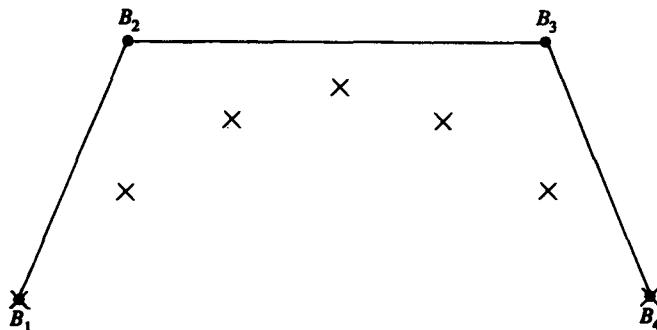


Рис. 5-55 Построение многоугольника В-сплайна для заданного множества точек.

квадратную матрицу (см. разд. 3-31). Вершины определяющего многоугольника, сглаживающего кривую заданы условием

$$\begin{aligned}[D] &= [N][B], \\ [N]^T[D] &= [N]^T[N][B]\end{aligned}$$

и

$$[B] = [[N]^T \quad [N]]^{-1} [N]^T [D] \quad (5-117)$$

Оба метода подразумевают, что матрица $[N]$ известна. Базисные функции $N_{i,k}(t_j)$ и, следовательно, матрицу $[N]$ можно получить, если задан порядок В-сплайна k , количество точек многоугольника $n + 1$, а также величина параметра вдоль кривой. При соблюдении условия $2 \leq k \leq n + 1 \leq j$ порядок и количество вершин произвольны.

Величина параметра t_j для каждой заданной точки — это мера расстояния до точки вдоль В-сплайна. Удобна аппроксимация параметра с помощью длин хорд между заданными точками; в частности, для j точек величина параметра в точке l

$$t_1 = 0$$

$$\frac{t_l}{t_{\max}} = \frac{\sum_{s=2}^l |D_s - D_{s-1}|}{\sum_{s=2}^j |D_s - D_{s-1}|} \quad l \geq 2.$$

Максимальное значение параметра t_{\max} обычно выбирается в качестве максимальной величины узлового вектора. Подобные методы затрагиваются в работах [5-15] и [5-22].

Для открытого равномерного узлового вектора с k кратными вершинами на концах кривая Безье получается при $n = k$.

Рассмотрим пример.

Пример 5-17 Подгонка В-сплайна

Даны пять точек $D_1[0 \ 0]$, $D_2[1.5 \ 2]$, $D_3[3 \ 2.5]$, $D_4[4.5 \ 2]$, $D_5[6 \ 0]$. Найти определяющие многоугольники третьего порядка ($k = 3$) с пятью и четырьмя вершинами для В-сплайна, проходящего через данные точки. Использовать хордовую аппроксимацию параметра.

Сначала найдем длины хорд.

$$D_{21} = |D_2 - D_1| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \sqrt{(1.5)^2 + (2)^2} = \sqrt{6.25} = 2.5,$$

$$D_{32} = |D_3 - D_2| = \sqrt{(1.5)^2 + (.5)^2} = 1.58,$$

$$D_{43} = |D_4 - D_3| = \sqrt{(1.5)^2 + (-.5)^2} = 1.58,$$

$$D_{54} = |D_5 - D_4| = \sqrt{(1.5)^2 + (-2)^2} = 2.5$$

и

$$\sum_{s=2}^5 (D_s - D_{s-1}) = D_{51} = 8.16.$$

Итак

$$\begin{aligned} t_1 &= 0 \\ \frac{t_2}{t_{\max}} &= \frac{D_{21}}{D_{51}} = \frac{2.5}{8.16} = 0.31 \\ \frac{t_3}{t_{\max}} &= \frac{D_{31}}{D_{51}} = \frac{(2.5 + 1.58)}{8.16} = 0.5 \\ \frac{t_4}{t_{\max}} &= \frac{D_{41}}{D_{51}} = \frac{(2.5 + 1.58 + 1.58)}{8.16} = 0.69 \\ \frac{t_5}{t_{\max}} &= \frac{D_{51}}{D_{51}} = 1. \end{aligned}$$

Для пятиугольника максимальное значение узлового вектора у В-сплайна третьего порядка $n - k + 2 = 4 - 3 + 2 = 3$. Здесь n на единицу меньше количества вершин многоугольника. Узловой вектор с кратностью k на концах имеет вид

$$[0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3].$$

Уравнение (5-115) с этими значениями выглядит следующим образом:

$$[D] = [N][B]$$

$$\begin{bmatrix} 0 & 0 \\ 1.5 & 2 \\ 3 & 2.5 \\ 4.5 & 2 \\ 6 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.007 & 0.571 & 0.422 & 0 & 0 \\ 0 & 0.125 & 0.75 & 0.125 & 0 \\ 0 & 0 & 0.422 & 0.571 & 0.007 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} [B].$$

Решим относительно $[B]$

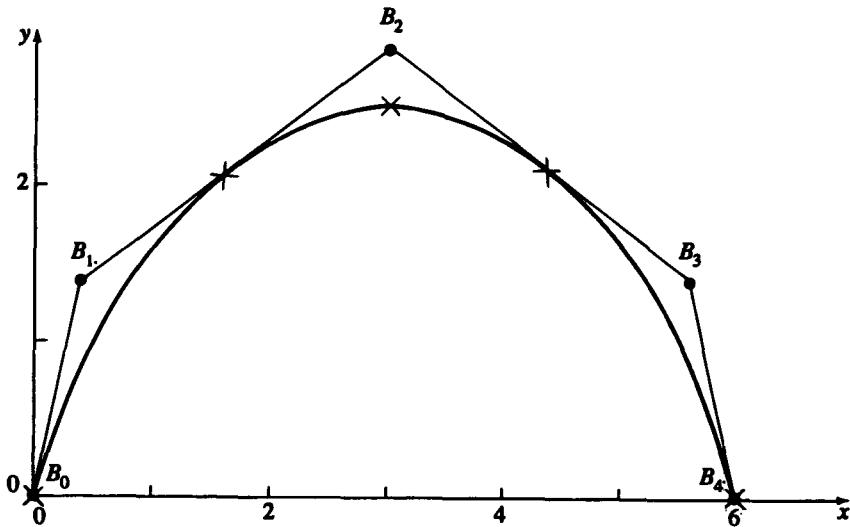
$$\begin{aligned} [B] &= [N]^{-1}[D] \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -0.013 & 2.037 & -307 & 0.286 & -0.002 \\ 0.003 & -0.387 & 1.769 & -0.287 & 0.003 \\ -0.002 & 0.286 & -1.307 & 2.037 & -0.013 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1.5 & 2 \\ 3 & 2.5 \\ 4.5 & 2 \\ 6 & 0 \end{bmatrix} \end{aligned}$$

$$[B] = \begin{bmatrix} 0 & 0 \\ 0.409 & 1.378 \\ 3 & 2.874 \\ 5.591 & 1.377 \\ 6 & 0 \end{bmatrix}.$$

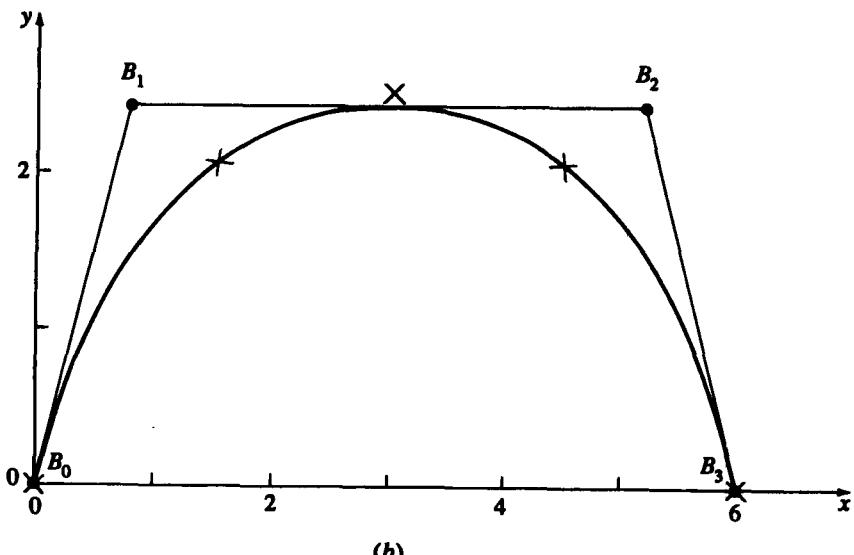
На рис. 5-56а изображены исходные точки, рассчитанные вершины многоугольника и кривая.

Для четырехугольника узловой вектор с кратностью k на концах:

$$[0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2].$$



(a)



(b)

Рис. 5-56 Результаты для примера 5-17. (a) Пятиугольник; (b) четырехугольник.

Матрица $[N]$:

$$[N] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.15 & 0.662 & 0.188 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.188 & 0.662 & 0.15 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Умножим на $[N]^T$ и обратим ее:

$$[[N]^T [N]]^{-1} = \begin{bmatrix} 0.995 & -0.21 & 0.106 & -0.005 \\ -0.21 & 2.684 & -1.855 & 0.106 \\ 0.106 & -1.855 & 2.684 & -0.21 \\ -0.005 & 0.106 & -0.21 & 0.995 \end{bmatrix}.$$

Тогда из уравнения (5-117) получим

$$[B] = [[N]^T [N]]^{-1} [N]^T [D] = \begin{bmatrix} 0 & 0 \\ 0.788 & 2.414 \\ 5.212 & 2.414 \\ 6 & 0 \end{bmatrix}.$$

На рис. 5-56 *b* изображены исходные точки, рассчитанные вершины многоугольника и кривая.

Этот метод допускает произвольное расположение всех точек определяющего многоугольника в трехмерном пространстве, но иногда бывает удобно ограничиться плоскостью, например при подгонке В-сплайна под заданный контур корабля. В работе [5-26] приведен соответствующий метод как для кривых, так и для поверхностей. Его сущность состоит в том, что итеративно вычисляется параметр для фиксированной координаты до тех пор, пока значение на кривой при величине параметра, вычисленной по подогнанному многоугольнику, не попадает в некоторую окрестность фиксированного значения, т. е. $|x_{\text{fixed}} - x_{\text{calc}}| \leq \text{еррор}$. Подгонка получается менее точной, но более удобной для модификаций.

5-12 РАЗБИЕНИЕ В-СПЛАЙНОВ

Для того чтобы увеличить гибкость кривой Безье, необходимо добавить вершину к определяющему многоугольнику, что повышает степень полиномиальной кривой (см. разд. 5-8). Гибкость В-сплайна также растет с увеличением порядка его базиса, т. е. полиномиальных сегментов. В работе [5-27] предлагаются теория и алгоритм увеличения порядка В-сплайна.

Повысить гибкость В-сплайна можно другим способом: вставить дополнительные узловые значения в определяющий узловой вектор. При этом полиномиальный сегмент для некоторого интервала узловых значений (параметрического интервала) локально разбивается на два. Есть два основных метода вставки узловых значений. Первый, так называемый алгоритм Осло, разработанный Коэном и др. ([5-28] и [5-29]), вставляет сразу несколько узловых значений. Второй метод Бема ([5-30] и [5-31]) последовательно вставляет по одному узловому значению. Мы рассмотрим только метод Осло.

Как повышение порядка, так и дополнительные узловые значения увеличивают гибкость базиса кривой (поверхности) и, следовательно, кривой без изменения ее формы. Это возможно потому, что один В-сплайн можно задать бесконечным множеством многоугольников с более чем минимальным количеством вершин. Форма кривой меняется путем передвижения вновь полученных вершин.

Рассмотрим исходную кривую $P(t) = \sum_{i=1}^{n+1} B_i N_{i,k}(t)$ с узловым вектором $[X] = [x_1 \ x_2 \ \dots \ x_{n+k-1}]$. Вставив узел, получаем новую кривую $R(s)$:

$$R(s) = \sum_{j=1}^{m+1} C_j M_{j,k}(s) \quad (5-118)$$

с новым узловым вектором

$$[Y] = [y_1 \ y_2 \ \dots \ y_{m+k+1}],$$

где $m > n$. Надо найти новые вершины определяющего многоугольника C_j , такие, что $P(t) = R(s)$. По алгоритму Осло [5-29]

$$C_j = \sum_{i=1}^{N+1} \alpha_{i,j}^k B_i, \quad 1 \leq i \leq n, \quad 1 \leq j \leq m, \quad (5-119)$$

где $\alpha_{i,j}^k$ заданы рекурсивным отношением

$$\alpha_{i,j}^1 = \begin{cases} 1 & x_i \leq y_j < x_{i+1}, \\ 0 & \text{в противном случае} \end{cases}, \quad (5-120a)$$

$$\alpha_{i,j}^k = \frac{y_{j+k-1} - x_i}{x_{i+k-1} - x_i} \alpha_{i,j}^{k-1} + \frac{x_{i+k} - y_{j+k-1}}{x_{i+k} - x_{i+1}} \alpha_{i+1,j}^{k-1}. \quad (5-120b)$$

Отметим, что $\sum_i^{n+1} \alpha_{i,j}^k = 1$.

На первый взгляд может показаться, что равномерный исходный вектор, не-важно периодический или открытый, после вставки одного или более узловых значений станет неравномерным. Равномерность вектора можно сохранить, добавляя кратные узловые значения в середину каждого ненулевого интервала. Рассмотрим разбиение подробнее на примере.

Пример 5-18 Глобальное разбиение B-сплайна

Пусть дан открытый B-сплайн третьего порядка ($k = 3$), изначально заданный четырьмя ($n + 1 = 4$) вершинами $B_1[0 \ 0]$, $B_2[1 \ 1]$, $B_3[2 \ 1]$, $B_4[3 \ 0]$. Требуется провести разбиение с сохранением открытого равномерного узлового вектора.

Исходный открытый равномерный вектор имеет вид:

$$[X'] = [0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2],$$

где два ненулевых интервала $0 \rightarrow 1$ и $1 \rightarrow 2$ соответствуют двум кусочно параболическим сегментам, составляющим B-сплайн. Предположим, что после разбиения мы должны иметь целочисленный равномерный вектор. Проведем преобразование параметра, умножая каждое узловое значение в $[X']$ на 2, и получим

$$[X] = [0 \ 0 \ 0 \ 2 \ 4 \ 4 \ 4]$$

с $x_1 = 0$, $x_2 = 0, \dots, x_7 = 4$. Кривая при этом не меняется.

Для разбиения кривой с учетом поставленных условий необходимо вставить узловые значения 1 и 3 в интервалы $0 \rightarrow 2$ и $2 \rightarrow 4$, соответственно. Запишем новый узловой вектор:

$$[Y] = [0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 4 \ 4]$$

с $y_1 = 0, y_2 = 0, \dots, y_9 = 4$.

Теперь B-сплайн состоит из четырех параболических сегментов.

Из уравнения (5-119) получим шесть новых вершин C_j . Значения $\alpha_{i,j}^3$ вычисляются с помощью рекурсивных соотношений (5-120). В частности, из уравнения (5-120а) видно, что единственное ненулевое $\alpha_{i,j}^1$ первого порядка ($k = 1$) таково:

$$\alpha_{3,1}^1 = \alpha_{3,2}^1 = \alpha_{3,3}^1 = \alpha_{3,4}^1 = \alpha_{4,5}^1 = \alpha_{4,6}^1 = 1.$$

Из уравнения (5-120б) ненулевые $\alpha_{i,j}^2$ второго порядка ($k = 2$) таковы:

$j = 1$

$$\alpha_{3,1}^2 = \frac{y_2 - x_3}{x_4 - x_3} \alpha_{3,1}^1 = \frac{0 - 0}{2 - 0}(1) = 0,$$

$$\alpha_{2,1}^2 = \frac{x_4 - y_2}{x_4 - x_3} \alpha_{3,1}^1 = \frac{2 - 0}{2 - 0}(1) = 1,$$

$j = 2$

$$\alpha_{2,2}^2 = \frac{x_4 - y_3}{x_4 - x_3} \alpha_{3,1}^1 = \frac{2 - 0}{2 - 0}(1) = 1,$$

$$\alpha_{3,2}^2 = \frac{y_3 - x_3}{x_4 - x_3} \alpha_{3,2}^1 = \frac{0 - 0}{2 - 0}(1) = 0,$$

$j = 3$

$$\alpha_{3,3}^2 = \frac{y_4 - x_3}{x_4 - x_3} \alpha_{3,3}^1 = \frac{1 - 0}{2 - 0}(1) = \frac{1}{2},$$

$$\alpha_{2,3}^2 = \frac{x_4 - y_4}{x_4 - x_3} \alpha_{3,3}^1 = \frac{2 - 1}{2 - 0}(1) = \frac{1}{2},$$

$j = 4$

$$\alpha_{3,4}^2 = \frac{y_5 - x_3}{x_4 - x_3} \alpha_{3,4}^1 = \frac{2 - 0}{2 - 0}(1) = 1,$$

$$\alpha_{2,4}^2 = \frac{x_4 - y_5}{x_4 - x_3} \alpha_{3,4}^1 = \frac{2 - 2}{2 - 0}(1) = 0,$$

$j = 5$

$$\alpha_{4,5}^2 = \frac{y_6 - x_4}{x_5 - x_4} \alpha_{4,5}^1 = \frac{3 - 2}{4 - 2}(1) = \frac{1}{2},$$

$$\alpha_{3,5}^2 = \frac{x_5 - y_6}{x_5 - x_4} \alpha_{4,5}^1 = \frac{4 - 3}{4 - 2}(1) = \frac{1}{2},$$

$j = 6$

$$\alpha_{4,6}^2 = \frac{y_7 - x_4}{x_5 - x_4} \alpha_{4,6}^1 = \frac{4 - 2}{4 - 2}(1) = 1,$$

$$\alpha_{3,6}^2 = \frac{x_5 - y_7}{x_5 - x_4} \alpha_{4,6}^1 = \frac{4 - 4}{4 - 2}(1) = 0$$

третьего порядка ($k = 3$)

$j = 1$

$$\alpha_{2,1}^3 = \frac{y_3 - x_2}{x_4 - x_2} \alpha_{2,1}^2 = \frac{0 - 0}{2 - 0}(1) = 0,$$

$$\alpha_{1,1}^3 = \frac{x_4 - y_3}{x_4 - x_2} \alpha_{2,1}^2 = \frac{2 - 0}{2 - 0}(1) = 1,$$

$j = 2$

$$\alpha_{1,2}^3 = \frac{x_4 - y_4}{x_4 - x_2} \alpha_{2,2}^2 = \frac{2 - 1}{2 - 0}(1) = \frac{1}{2},$$

$$\alpha_{2,2}^3 = \frac{y_4 - x_2}{x_4 - x_2} \alpha_{2,2}^2 = \frac{1 - 0}{2 - 0}(1) = \frac{1}{2},$$

$j = 3$

$$\alpha_{3,3}^3 = \frac{y_5 - x_3}{x_5 - x_3} \alpha_{3,3}^2 = \frac{2 - 0}{4 - 0}\left(\frac{1}{2}\right) = \frac{1}{4},$$

$$\alpha_{2,3}^3 = \frac{y_5 - x_2}{x_4 - x_2} \alpha_{2,3}^2 + \frac{x_5 - y_5}{x_5 - x_3} \alpha_{3,3}^2 = \frac{2 - 0}{2 - 0}\left(\frac{1}{2}\right) + \frac{4 - 2}{4 - 0}\left(\frac{1}{2}\right) = \frac{1}{2} + \frac{1}{4} = \frac{3}{4},$$

$$\alpha_{1,3}^3 = \frac{x_4 - y_5}{x_4 - x_2} \alpha_{2,3}^2 = \frac{2 - 2}{2 - 0}\left(\frac{1}{2}\right) = 0,$$

$j = 4$

$$\alpha_{3,4}^3 = \frac{y_6 - x_3}{x_5 - x_3} \alpha_{3,4}^2 = \frac{3 - 0}{4 - 0}(1) = \frac{3}{4},$$

$$\alpha_{2,4}^3 = \frac{x_5 - y_6}{x_5 - x_3} \alpha_{3,4}^2 = \frac{4 - 3}{4 - 0}(1) = \frac{1}{4},$$

$j = 5$

$$\alpha_{2,5}^3 = \frac{x_5 - y_7}{x_5 - x_3} \alpha_{3,5}^2 = \frac{4 - 4}{4 - 0}\left(\frac{1}{2}\right) = 0,$$

$$\alpha_{3,5}^3 = \frac{y_7 - x_3}{x_5 - x_3} \alpha_{3,5}^2 + \frac{x_6 - y_7}{x_6 - x_4} \alpha_{4,5}^2 = \frac{4 - 0}{4 - 0}\left(\frac{1}{2}\right) + \frac{4 - 4}{4 - 2}\left(\frac{1}{2}\right) = \frac{1}{2},$$

$$\alpha_{4,5}^3 = \frac{y_7 - x_4}{x_6 - x_4} \alpha_{4,5}^2 = \frac{4 - 2}{4 - 2}\left(\frac{1}{2}\right) = \frac{1}{2},$$

$j = 6$

$$\alpha_{4,6}^3 = \frac{y_8 - x_4}{x_6 - x_4} \alpha_{4,6}^2 = \frac{4 - 2}{4 - 2}(1) = 1,$$

$$\alpha_{3,6}^3 = \frac{x_6 - y_8}{x_6 - x_4} \alpha_{4,6}^2 = \frac{4 - 4}{4 - 2}(1) = 0.$$

Новые вершины рассчитываются по уравнению (5-119):

$$\begin{aligned} C_1 &= \sum_{i=1}^{n+1} \alpha_{i,1}^3 B_i = \alpha_{1,1}^3 B_1 + \alpha_{2,1}^3 B_2 + \alpha_{3,1}^3 B_3 + \alpha_{4,1}^3 B_4 = \\ &= \alpha_{1,1}^3 B_1 = B_1 = [0 \quad 0]. \end{aligned}$$

Аналогично

$$C_2 = \frac{1}{2}(B_1 + B_2) = \frac{1}{2}([0 \ 0] + [1 \ 1]) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

$$C_3 = \frac{3}{4}B_2 + \frac{1}{4}B_3 = \frac{3}{4}[1 \ 1] + \frac{1}{4}[2 \ 1] = \begin{bmatrix} \frac{5}{4} & 1 \end{bmatrix},$$

$$C_4 = \frac{1}{4}B_2 + \frac{3}{4}B_3 = \frac{1}{4}[1 \ 1] + \frac{3}{4}[2 \ 1] = \begin{bmatrix} \frac{7}{4} & 1 \end{bmatrix},$$

$$C_5 = \frac{1}{2}(B_3 + B_4) = \frac{1}{2}([2 \ 1] + [3 \ 0]) = \begin{bmatrix} \frac{5}{2} & \frac{1}{2} \end{bmatrix},$$

$$C_6 = B_4 = [3 \ 0].$$

В результате увеличивается гибкость всей кривой.

В следующем примере в исходный открытый равномерный узловый вектор вносится один узел, после чего вектор становится неоднородным.

Пример 5-19 Локальное разбиение B-сплайна

Рассмотрим открытый B-сплайн третьего порядка ($k = 3$) из примера 5-18. Провести разбиение, добавив узловое значение 1 на интервале $0 \rightarrow 1$ вектора: $[X] = [0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2]$, где $x_1 = 0, \dots, x_7 = 2$. Новый узловый вектор имеет вид: $[Y] = [0 \ 0 \ 0 \ 1 \ 1 \ 2 \ 2 \ 2]$, где $y_1 = 0, \dots, y_8 = 0$. Новый многоугольник имеет пять вершин $C_1 \dots C_5$.

Ненулевые $\alpha_{i,j}^k$, необходимые для расчета C_j :

$$k=1$$

$$\alpha_{3,1}^1 = \alpha_{3,2}^1 = \alpha_{3,3}^1 = \alpha_{3,4}^1 = \alpha_{4,5}^1 = 1$$

$$k=2$$

$$\alpha_{2,1}^2 = \alpha_{2,2}^2 = \alpha_{3,3}^2 = \alpha_{3,4}^2 = \alpha_{4,5}^2 = 1$$

$$k=3$$

$$\alpha_{1,1}^3 = \alpha_{2,2}^3 = \alpha_{3,4}^3 = \alpha_{4,5}^3 = 1, \quad \alpha_{3,3}^3 = \alpha_{2,3}^3 = \frac{1}{2}.$$

Новые вершины:

$$C_1 = \alpha_{3,1}^3 B_1 = B_1 = [0 \ 0],$$

$$C_2 = \alpha_{3,2}^3 B_2 = B_2 = [1 \ 1],$$

$$C_3 = \alpha_{2,3}^3 B_2 + \alpha_{3,3}^3 B_3 = \frac{1}{2}(B_2 + B_3) = \frac{1}{2}([1 \ 1] + [2 \ 1]) = \begin{bmatrix} \frac{3}{2} & 1 \end{bmatrix},$$

$$C_4 = \alpha_{3,4}^3 B_3 = B_3 = [2 \ 1],$$

$$C_5 = \alpha_{4,5}^3 B_4 = B_4 = [3 \ 0].$$

Если C_3 сдвинуть до совпадения с C_2 , т. е. $C_2 = C_3 = [1 \ 1]$, то получится как двойная вершина, так и двойной узел для $C_2 = C_3$. Для $k = 3$ B-сплайн имеет острый пик в $C_2 = C_3$ (см. задачу 5-30).

5-13 РАЦИОНАЛЬНЫЕ В-СПЛАЙНЫ

Впервые в машинной графике описание рациональных кривых и поверхностей было предложено в работе [5-32]. В литературе широко известны рациональные формы кубических сплайнов и кривых Безье ([5-33]—[5-37]), а также конических сечений [5-38]. Здесь мы рассмотрим только рациональные В-сплайны, так как они составляют общепринятую основу, а также из-за пространственных ограничений. Рациональные В-сплайны это единственное точное математическое представление, охватывающее все аналитические формы — прямые, плоскости, конические сечения, включающие окружности, кривые произвольной формы, квадрики и трехмерные поверхности, используемые в машинной графике и проектировании.

Первым рациональные В-сплайны изучил Весприлл [5-39]. Основа наших рассуждений приведена в работах [5-40] и [5-41]. Стоит отметить, что неравномерные рациональные В-сплайны (NURBS) с 1983 г. являются стандартом IGES (5-42). IGES — это стандарт обмена проектной информацией между системами машинного проектирования, а также между ними и системами автоматизированного производства. Рациональные В-сплайны применяются в ряде систем геометрического моделирования и реализованы аппаратно (СБИС или микрокод) в некоторых графических рабочих станциях.

Рациональный В-сплайн это проекция нерационального (полиномиального) В-сплайна, определенного в четырехмерном (4D) однородном координатном пространстве, на трехмерное (3D) физическое пространство. В частности

$$P(t) = \sum_{i=1}^{n+1} B_i^h N_{i,k}(t), \quad (5-121)$$

где B_i^h — вершины многоугольника для нерационального 4D В-сплайна в четырехмерном пространстве, $N_{i,k}(t)$ — функция базиса нерационального В-сплайна из уравнения (5-84).

Рациональный В-сплайн получается после проектирования, т. е. деления на однородную координату

$$P(t) = \frac{\sum_{i=1}^{n+1} B_i h_i N_{i,k}(t)}{\sum_{i=1}^{n+1} h_i N_{i,k}(t)} = \sum_{i=1}^{n+1} B_i R_{i,k}(t), \quad (5-122)$$

где B_i — вершины трехмерного многоугольника для рационального В-сплайна, а

$$R_{i,k}(t) = \frac{h_i N_{i,k}(t)}{\sum_{i=1}^{n+1} h_i N_{i,k}(t)} \quad (5-123)$$

— базисные функции рационального В-сплайна. Здесь $h_i \geq 0$ для всех i^1 .

¹ Базисные функции рационального В-сплайна могут быть определены и для $h_i < 0$ (см. [5-39]), но в данном случае это неудобно.

Как видно из уравнений (5-121) – (5-123), рациональные B-сплайны и их базисы это обобщение нерациональных B-сплайнов и базисов. Они наследуют почти все аналитические и геометрические свойства последних. В частности:

Каждая функция рационального базиса положительна или равна нулю для всех значений параметра, т. е. $R_{i,k} \geq 0$.

Для любого значения параметра t сумма базисных функций рационального B-сплайна равна единице, т. е.

$$\sum_{i=1}^{n+1} R_{i,k}(t) \equiv 1. \quad (5-124)$$

Кроме $k = 1$ каждая рациональная базисная функция имеет ровно один максимум.

Рациональный B-сплайн порядка k (степени $k - 1$) везде C^{k-2} непрерывен.

Максимальный порядок рационального B-сплайна равен количеству вершин определяющего многоугольника.

Рациональный B-сплайн обладает свойством уменьшения вариации.

Общая форма рационального B-сплайна повторяет очертания определяющего многоугольника.

Любое *проективное* преобразование рационального B-сплайна производится соответствующим преобразованием вершин определяющего многоугольника; т. е. кривая инварианта относительно *проективного* преобразования. Это более сильное условие, чем для нерационального B-сплайна, который инвариантен только относительно *аффинного* преобразования.

Из уравнений (5-85) и (5-123) видно, что при всех $h_i = 1$, $R_{i,k}(t) = N_{i,k}(t)$, т. е. нерациональные B-сплайны и их базисы являются частным случаем рациональных. Легко показать, что открытый рациональный B-сплайн с порядком, равным количеству вершин определяющего многоугольника, представляет собой рациональную кривую Безье. В случае $h_i = 1$ рациональная кривая Безье сводится к нерациональной. Таким образом, B-сплайны включают как рациональные, так и нерациональные кривые Безье.

Рациональные B-сплайны – это четырехмерное обобщение нерациональных B-сплайнов; поэтому алгоритмы повышения степени (работа [5-27] и пример 6-18), разбиения (разд. 5-12 и работы [5-28] – [5-31]) и подгонки (разд. 5-11) для нерациональных B-сплайнов справедливы и для рациональных при распространении их на четырехмерное пространство.

Рациональные B-сплайны и их базисы можно получить с помощью открытых равномерных, периодических равномерных и неравномерных узловых векторов.

Однородные координаты h_i (также называемые весами) в уравнениях (5-122) и (5-123) предоставляют дополнительные возможности изгиба кривой; $h = 1$ называется аффинным пространством и соответствует физическому пространству. На рис. 5-57 показано влияние однородных координат h на базисные функции рационального B-сплайна, где $[0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3]$ ($n + 1 = 5$, $k = 3$) – открытый равномерный узловой вектор, $h_i = 1$, $i + 3$ – вектор однородных координат. Значе-

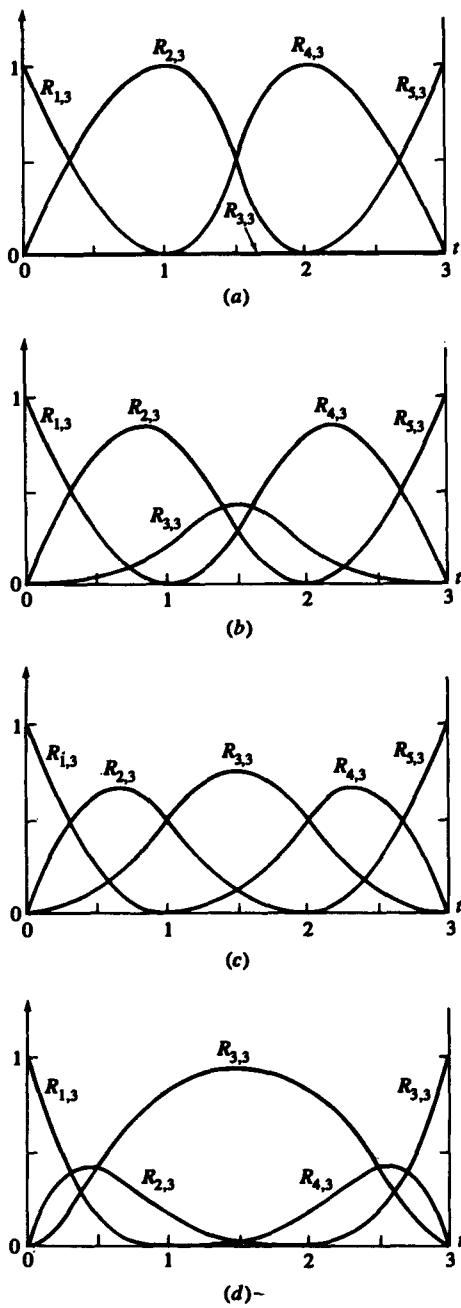


Рис. 5-57 Базисные функции рационального В-сплайна для $n+1 = 5, k = 3$ с открытым узловым вектором $[X] = [0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3]$, $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$. (а) $h_3 = 0$; (б) $h_3 = 0.25$; (в) $h_3 = 1.0$; (г) $h_3 = 5.0$.

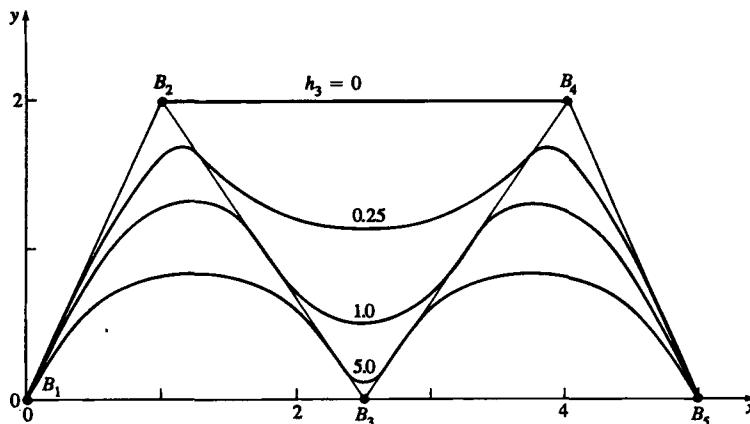


Рис. 5-58 Рациональные B-сплайны для $n+1 = 5, k = 3$ с открытым узловым вектором $[X] = [0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3]$, $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$.

ния h_3 лежат в диапазоне от 0 до 5. Базисные функции рационального B-сплайна на рис. 5.57с при $h = 1$ совпадают с соответствующими функциями нерационального B-сплайна. Рациональный B-сплайн для $h_3 = 1$ на рис. 5.58 также совпадает с соответствующим нерациональным. Отметим, что для $h_3 = 0$ (рис. 5.57а) $R_{3,3}$ везде равно нулю; т. е. соответственная вершина B_3 не оказывает никакого влияния на форму соответствующей кривой. Это показано на рис. 5.58, где вершины определяющего многоугольника B_2 и B_4 соединены прямой. Из рис. 5.57 также видно, что при увеличении h_3 также возрастает $R_{3,3}$ и, вследствие уравнения (5-124), $R_{2,3}$ и $R_{4,3}$ уменьшаются. На рис. 5.58 изображено воздействие на соответствующие рациональные B-сплайны. В частности отметим, что с увеличением h_3 кривая приближается к B_3 . Отсюда, как уже отмечалось, следует, что однородные координаты дают возможность увеличить гибкость кривой. Иллюстрация аналогичных свойств рациональных B-сплайнов и базисов четвертого порядка ($k = 4$) дана на рис. 5.59 и 5.60. Однако для кривых более высокого порядка на рис. 5.60 кривая при $h_3 = 0$ не вырождается в отрезок прямой между B_2 и B_4 .

На рис. 5.61 изображен периодический равномерный базис для $n+1 = 5, k = 3$ и узлового вектора $[X] = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$ и вектора однородных координат $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$ с $0 \leq h_3 \leq 5$. Здесь, как и в случае нерационального базиса, допустимый диапазон параметра $2 \leq t \leq 5$. На рис. 5.61 показан только этот диапазон. Для $h_3 = 1$ базисные функции рационального B-сплайна совпадают с соответствующими нерациональными, но для $h_3 \neq 1$ они становятся непериодическими и не являются параллельным переносом друг друга. На рис. 5.62 приведены соответствующие рациональные B-сплайны. Отметим, что конечные точки всех кривых совпадают.

На рис. 5.63 и 5.64 показаны соответствующие B-сплайны четвертого порядка ($k = 4$) и их базисы. Отметим, что начальные и конечные точки кривых лежат на прямых.

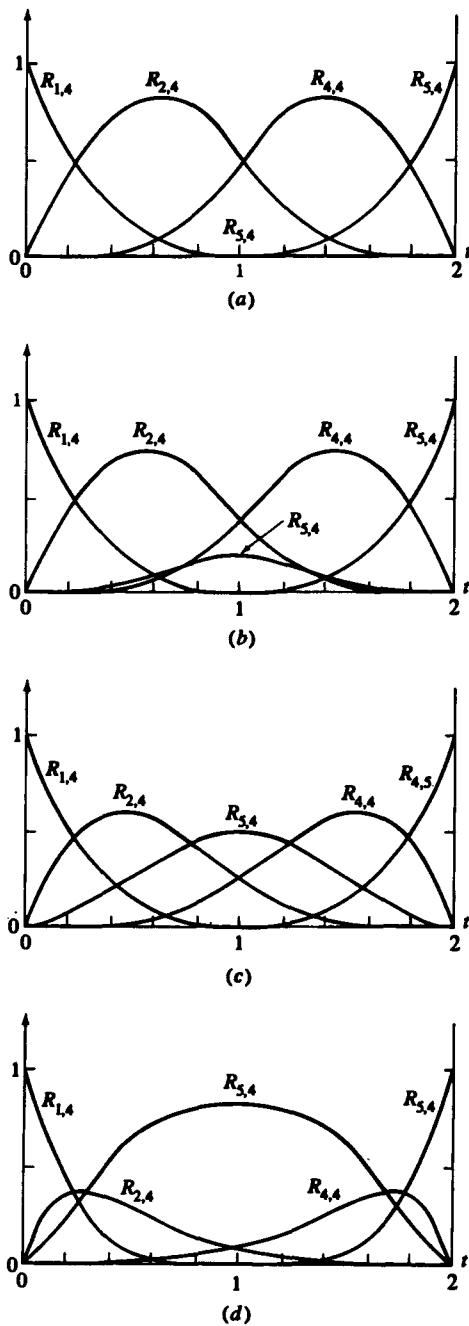


Рис. 5-59 Базисные функции рационального В-сплайна для $n+1 = 5$, $k = 4$ с открытым узловым вектором $[X] = [0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2 \ 2]$, $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$. (а) $h_3 = 0$; (б) $h_3 = 0.25$; (в) $h_3 = 1.0$; (г) $h_3 = 5.0$.

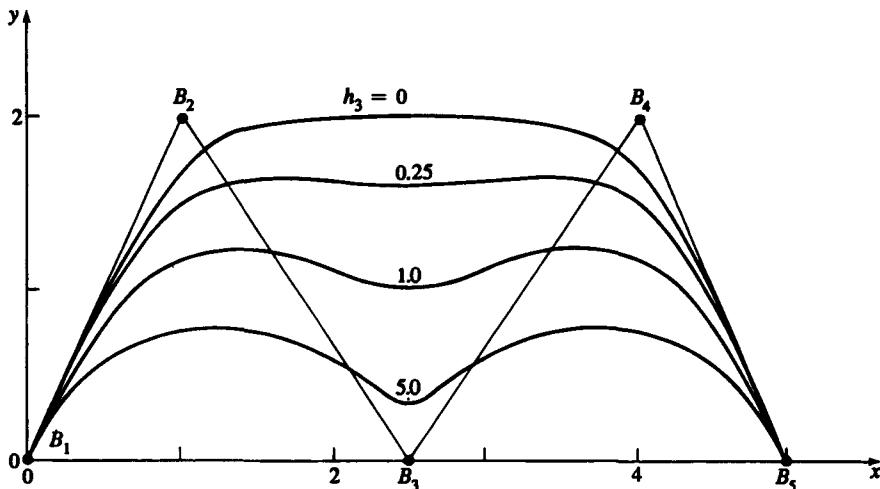


Рис. 5-60 Рациональные B-сплайны для $n+1 = 5$, $k = 4$ с открытым узловым вектором $[X] = [0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2]$, $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$.

Вычисляя уравнения (5-122) и (5-123) для конечных точек кривой с аргументом $(t_{\max} - \varepsilon)_{\varepsilon \rightarrow 0}$ из примера 5-12, получаем, что первая и последняя точки открытого рационального B-сплайна совпадают с конечными точками определяющего многоугольника:

$$P(0) = B_1 \quad \text{и} \quad P(t_{\max}) = P(n - k + 2) = B_{n+1}.$$

Из рис. 5-65 видно, что перемещение одной вершины многоугольника влияет на кривую так же, как и в случае рациональных B-сплайнов. В данном случае $[H] = [1 \ 1 \ 0.25 \ 1 \ 1]$. Если $h_3 = 0$, то передвижение B_3 не оказывает никакого воздействия. С увеличением h_3 растет влияние движения точки B_3 .

На рис. 5-66 показано воздействие кратных вершин в B_3 на рациональный B-сплайн четвертого порядка. Как и для нерациональных кривых, $k - 1$ совпадающая вершина приводит к появлению острого угла или пика. Далее, кратная вершина порождает ребра нулевой длины, поэтому существование угла не зависит от значений $h_3 \geq 0$, соответствующих ей (см. задачу 5-33).

Рассмотрим пример расчета рационального B-сплайна.

Пример 5-20 Расчет открытых рациональных B-сплайнов

Пусть дан многоугольник с вершинами $B_1[0 \ 1]$, $B_2[1 \ 2]$, $B_3[2.5 \ 0]$, $B_4[4 \ 2]$, $B_5[5 \ 0]$. Найти точку при $t = 3/2$ на открытом рациональном B-сплайне третьего порядка ($k = 3$) с однородными векторами, заданными $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$, $h_3 = 0, 1/4, 1, 5$.

Узловой вектор — $[0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3]$. Диапазон параметра t : $0 \leq t \leq 3$. Кривые состоят из трех кусочно рациональных квадратичных кривых, по одной на каждом внутреннем интервале узлового вектора.

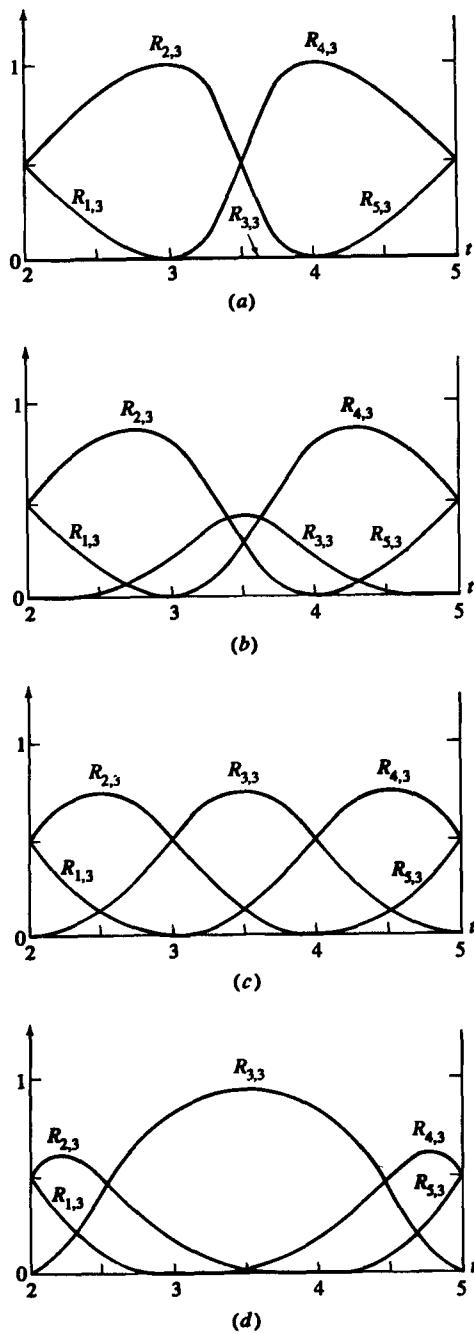


Рис. 5-61 Базисные функции рационального В-сплайна для $n+1 = 5$, $k = 3$ сperi одицеским узловым вектором $[X] = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$, $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$
 (a) $h_3 = 0$; (b) $h_3 = 0.25$; (c) $h_3 = 1.0$; (d) $h_3 = 5.0$.

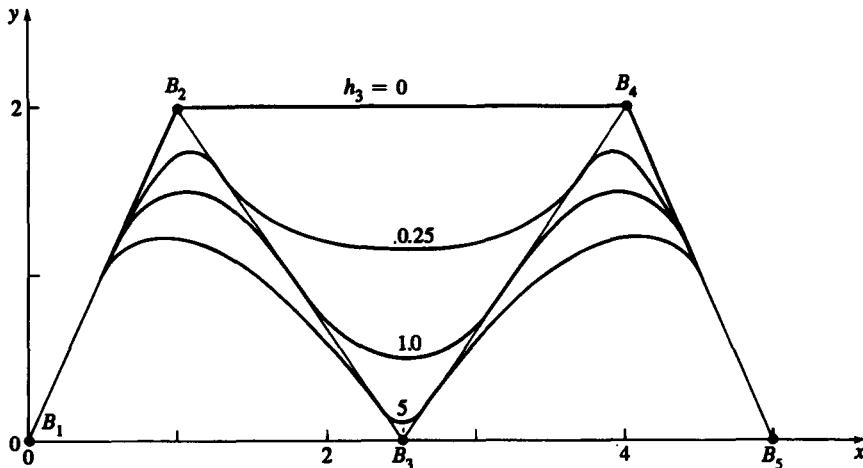


Рис. 5-62 Рациональные B-сплайны для $n + 1 = 5$, $k = 4$ с периодическим узловым вектором $[X] = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$, $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$.

Из уравнения (5-84) на интервале $1 \leq t \leq 2$ получим базисные функции нерационального B-сплайна:

$$1 \leq t < 2$$

$$N_{4,1}(t) = 1; \quad N_{i,1}(t) = 0, \quad i \neq 4$$

$$N_{3,2}(t) = (2 - t); \quad N_{4,2}(t) = (t - 1); \quad N_{i,2}(t) = 0, \quad i \neq 3, 4$$

$$N_{2,3}(t) = \frac{(2 - t)^2}{2}; \quad N_{3,3}(t) = \frac{t(2 - t)}{2} + \frac{(3 - t)(t - 1)}{2};$$

$$N_{4,3}(t) = \frac{(t - 1)^2}{2}; \quad N_{i,3}(t) = 0, \quad i \neq 2, 3, 4$$

Отсюда и из уравнения (5-123) после определения знаменателя

$$\begin{aligned} S &= \sum_{i=1}^{n+1} h_i N_{i,k}(t) = h_1 N_{1,3}(t) + h_2 N_{2,3}(t) + h_4 N_{4,3}(t) + h_5 N_{5,3}(t) = \\ &= h_2 N_{2,3}(t) + h_4 N_{4,3}(t) = \\ &= \frac{(2 - t)^2}{2} + \frac{(t - 1)^2}{2} = \frac{2t^2 - 6t + 5}{2} \end{aligned}$$

получим базис рационального B-сплайна:

$$1 \leq t < 2$$

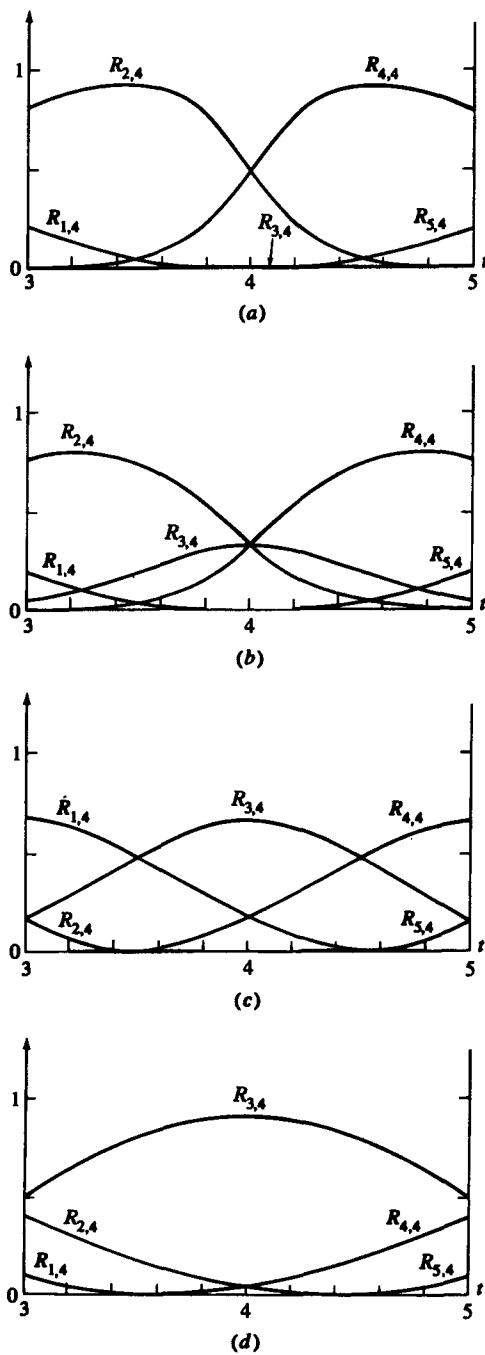


Рис. 5-63 Базисные функции рационального В-сплайна для $n+1 = 5$, $k = 4$ с периодическим узловым вектором $[X] = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8]$, $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$.
 (a) $h_3 = 0$; (b) $h_3 = 0.25$; (c) $h_3 = 1.0$; (d) $h_3 = 5.0$.

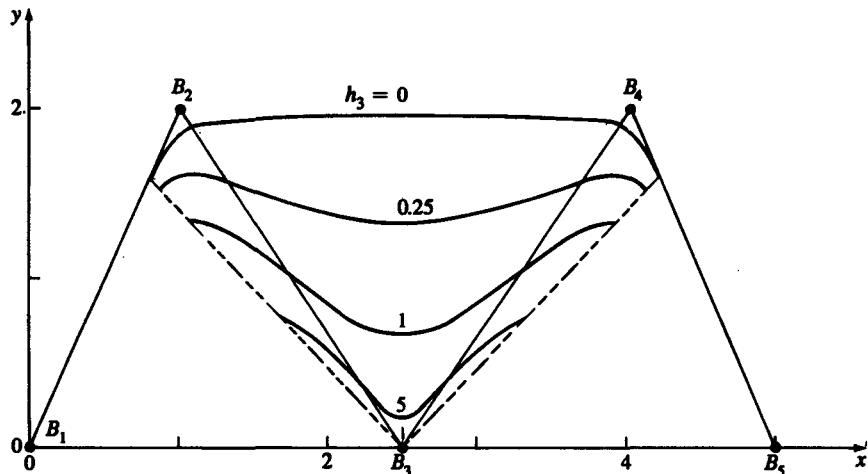


Рис. 5-64 Рациональные B-сплайны для $n + 1 = 5$, $k = 4$ с периодическим узловым вектором $[X] = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8]$, $[H] = [1 \ 1 \ h_3 \ 1 \ 1]$.

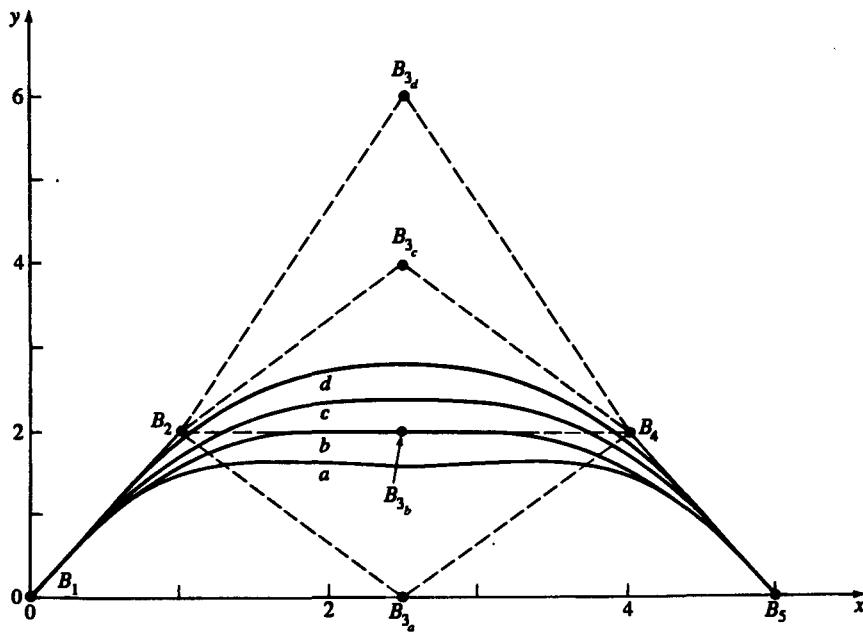


Рис. 5-65 Влияние перемещения одной вершины многоугольника на рациональный B-сплайн $n + 1 = 5$, $k = 4$, $[H] = [1 \ 1 \ 0.25 \ 1 \ 1]$.

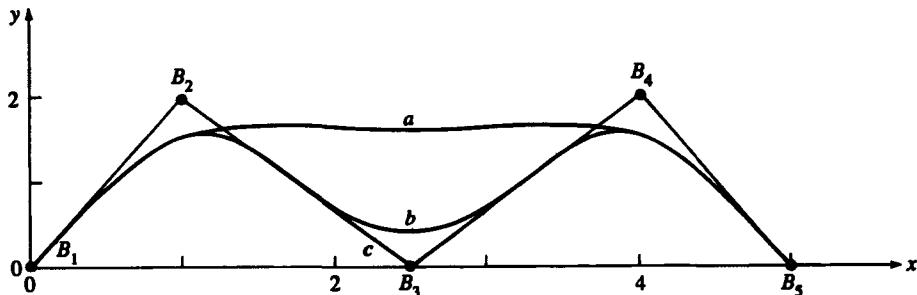


Рис. 5-66 Влияние кратной вершины многоугольника B_3 на рациональный B-сплайн $n+1=5, k=4$. (а) Одна вершина $[H] = [1 \ 1 \ 0.25 \ 1 \ 1]$; (б) двойная вершина $[H] = [1 \ 1 \ 0.25 \ 0.25 \ 1 \ 1]$; (с) тройная вершина $[H] = [1 \ 1 \ 0.25 \ 0.25 \ 0.25 \ 1 \ 1]$.

$$h_3 = 0$$

$$R_{1,3}(t) = 0$$

$$R_{2,3}(t) = \frac{h_2 N_{2,3}(t)}{S} = \frac{(2-t)^2}{2t^2 - 6t + 5}$$

$$R_{3,3}(t) = 0$$

$$R_{4,3}(t) = \frac{h_4 N_{4,3}(t)}{S} = \frac{(t-1)^2}{2t^2 - 6t + 5}$$

$$R_{5,3}(t) = 0$$

$$h_3 = 1/4$$

$$S = h_2 N_{2,3}(t) + h_3 N_{3,3}(t) + h_4 N_{4,3}(t)$$

$$\begin{aligned} &= \frac{(2-t)^2}{2} + \frac{t(2-t)}{8} + \frac{(3-t)(t-1)}{8} + \frac{(t-1)^2}{2} \\ &= \frac{6t^2 - 18t + 17}{8} \end{aligned}$$

$$R_{1,3}(t) = 0$$

$$R_{2,3}(t) = \frac{4(2-t)^2}{6t^2 - 18t + 17}$$

$$R_{3,3}(t) = \frac{t(2-t) + (3-t)(t-1)}{6t^2 - 18t + 17} = \frac{-2t^2 + 6t - 3}{6t^2 - 18t + 17}$$

$$R_{4,3}(t) = \frac{4(t-1)^2}{6t^2 - 18t + 17}$$

$$R_{5,3}(t) = 0$$

$$h_3 = 1$$

$$S = 1$$

$$R_{1,3}(t) = 0$$

$$R_{2,3}(t) = N_{2,3}(t) = \frac{(2-t)^2}{2}$$

$$R_{3,3}(t) = N_{3,3}(t) = \frac{t(2-t)}{2} + \frac{(3-t)(t-1)}{2}$$

$$R_{4,3}(t) = N_{4,3}(t) = \frac{(t-1)^2}{2}$$

$$R_{5,4}(t) = 0$$

$h_3 = 5$

$$S = \frac{(2-t)^2}{2} + \frac{5t(2-t)}{2} + \frac{5(3-t)(t-1)}{2} + \frac{(t-1)^2}{2} = -4t^2 + 12t - 5$$

$$R_{1,3}(t) = 0$$

$$R_{2,3}(t) = \frac{(2-t)^2}{2(-4t^2 + 12t - 5)}$$

$$R_{3,3}(t) = \frac{5t(2-t) + 5(3-t)(t-1)}{2(-4t^2 + 12t - 5)} = \frac{5(-2t^2 + 6t - 3)}{2(-4t^2 + 12t - 5)}$$

$$R_{4,3}(t) = \frac{(t-1)^2}{2(-4t^2 + 12t - 5)}$$

$$R_{5,3}(t) = 0$$

Полностью результаты приведены на рис. 5-57.

При $t = 3/2$

$$h_3 = 0 : \quad R_{1,3}(3/2) = 0; \quad R_{2,3}(3/2) = \frac{1}{2}; \quad R_{3,3}(3/2) = 0;$$

$$R_{4,3}(3/2) = \frac{1}{2}; \quad R_{5,3}(3/2) = 0$$

$$h_3 = \frac{1}{4} : \quad R_{1,3}(3/2) = 0; \quad R_{2,3}(3/2) = \frac{2}{7}; \quad R_{3,3}(3/2) = \frac{3}{7};$$

$$R_{4,3}(3/2) = \frac{2}{7}; \quad R_{5,3}(3/2) = 0$$

$$h_3 = 1 : \quad R_{1,3}(3/2) = 0; \quad R_{2,3}(3/2) = \frac{1}{8}; \quad R_{3,3}(3/2) = \frac{3}{4};$$

$$R_{4,3}(3/2) = \frac{1}{8}; \quad R_{5,3}(3/2) = 0$$

$$h_3 = 5 : \quad R_{1,3}(3/2) = 0; \quad R_{2,3}(3/2) = \frac{1}{32}; \quad R_{3,3}(3/2) = \frac{15}{16};$$

$$R_{4,3}(3/2) = \frac{1}{32}; \quad R_{5,3}(3/2) = 0$$

Соответственные точки на рациональных В-сплайнах:

$$h_3 = 0 : \quad P(3/2) = \frac{1}{2}[1 \quad 2] + \frac{1}{2}[4 \quad 2] = \begin{bmatrix} \frac{5}{2} & 2 \end{bmatrix}$$

$$h_3 = \frac{1}{4} : \quad P(3/2) = \frac{2}{7}[1 \quad 2] + \frac{3}{7}\begin{bmatrix} \frac{5}{2} & 0 \end{bmatrix} + \frac{2}{7}[4 \quad 2] = \begin{bmatrix} \frac{5}{2} & \frac{8}{7} \end{bmatrix}$$

$$h_3 = 1 : \quad P(3/2) = \frac{1}{8}[1 \quad 2] + \frac{3}{4}\begin{bmatrix} \frac{5}{2} & 0 \end{bmatrix} + \frac{1}{8}[4 \quad 2] = \begin{bmatrix} \frac{5}{2} & \frac{1}{2} \end{bmatrix}$$

$$h_3 = 5 : \quad P(3/2) = \frac{1}{32}[1 \quad 2] + \frac{15}{16}\begin{bmatrix} \frac{5}{2} & 0 \end{bmatrix} + \frac{1}{32}[4 \quad 2] = \begin{bmatrix} \frac{5}{2} & \frac{1}{8} \end{bmatrix}$$

Полностью результаты приведены на рис. 5-58.

Формально дифференцируя уравнения (5-122) и (5-123), получаем производные рациональных В-сплайнов, а именно

$$P'(t) = \sum_{i=1}^{n+1} B_i R'_{i,k}(t) \quad (5-125)$$

где

$$R'_{i,k}(t) = \frac{h_i N'_{i,k}(t)}{\sum_{i=1}^{n+1} h_i N_{i,k}} - \frac{h_i N_{i,k} \sum_{i=1}^{n+1} h_i N'_{i,k}}{\left(\sum_{i=1}^{n+1} h_i N_{i,k}\right)^2}. \quad (5-126)$$

Подставим $t = 0$ и $t = n - k + 2$:

$$P'(0) = (k-1) \frac{h_2}{h_1} (B_2 - B_1), \quad (5-127)$$

$$P'(n-k+2) = (k-1) \frac{h_n}{h_{n+1}} (B_{n+1} - B_n), \quad (5-128)$$

т.е. наклон кривой в концах совпадает с наклоном первого и последнего ребер многоугольника, соответственно.

Аналогично рассчитываются производные более высокого порядка (см. задачи 5-35 и 5-36).

Рассмотрим простой пример.

Пример 5-21 Производные открытых рациональных сплайнов

Пусть дан определяющий многоугольник из примера 5-16 с вершинами $B_1[1 \ 1], B_2[2 \ 3], B_3[4 \ 3], B_4[3 \ 1]$. Найти первую производную рационального B-сплайна второго порядка ($k = 2$) с $[H] = [1 \ 1/2 \ 1 \ 1]$.

Узловой вектор $[X] = [0 \ 0 \ 1 \ 2 \ 3 \ 3]$. Параметр изменяется в диапазоне $0 \leq t \leq 3$. Из уравнения (5-125) первая производная

$$P'(t) = B_1 R'_{1,2}(t) + B_2 R'_{2,2}(t) + B_3 R'_{3,2}(t) + B_4 R'_{4,3}(t)$$

Из уравнений (5-12) и (5-16) получаем нерациональные базисные функции и их производные:

$$0 \leq t < 1$$

$$N_{1,2}(t) = 1 - t; \quad N_{2,2}(t) = t; \quad N_{i,2}(t) = 0, \quad i \neq 1, 2$$

$$N'_{1,2}(t) = -1; \quad N'_{2,2}(t) = 1; \quad N'_{i,2}(t) = 0, \quad i \neq 1, 2$$

Используя уравнение (5-126), найдем рациональный базис и его производные

$$\begin{aligned} \sum_{i=1}^{n+1} h_i N_{i,2} &= \frac{2-t}{2}; \quad \sum_{i=1}^{n+1} h_i N'_{i,2} = -\frac{1}{2}; \\ R'_{1,2}(t) &= \frac{2}{2-t} \left\{ (-1) - (1-t) \frac{(-1)}{(2-t)} \right\} = \frac{-2}{(2-t)^2} \\ R'_{2,2}(t) &= \left(\frac{1}{2} \right) \left(\frac{2}{2-t} \right) \left\{ 1 - t \frac{(-1)}{(2-t)} \right\} = \frac{2}{(2-t)^2} \\ R'_{i,2}(t) &= 0, \quad i \neq 1, 2 \end{aligned}$$

Итак

$$P'(t) = \frac{2}{(2-t)^2} (B_2 - B_1)$$

Отметим, что в отличие от нерационального B-сплайна из примера 5-16, при сохранении наклона равным наклону первого ребра значение производной теперь изменяется вдоль кривой. При $t = 0$, т. е. в начале кривой

$$P'(t) = \frac{1}{2}(B_2 - B_1)$$

величина производной в два раза меньше, чем у нерационального B-сплайна из примера 5-16.

На интервале

$$1 \leq t < 2$$

$$\begin{aligned} N'_{2,2}(t) &= 2 - t; & N_{3,2}(t) &= t - 1; & N_{i,2}(t) &= 0, & i &\neq 2, 3 \\ N'_{2,2}(t) &= -1; & N'_{3,2}(t) &= 1; & N'_{i,2}(t) &= 0, & i &\neq 2, 3 \end{aligned}$$

Здесь

$$\begin{aligned} \sum_{i=1}^{n+1} h_i N_{i,2} &= \frac{t}{2}; & \sum_{i=1}^{n+1} h_i N'_{i,2} &= \frac{1}{2}; \\ R'_{2,2}(t) &= \left(\frac{1}{2}\right) \left(\frac{2}{t}\right) \left\{ (-1) - (2-t) \left(\frac{1}{t}\right) \right\} &= -\frac{2}{t^2} \\ R'_{3,2}(t) &= \frac{2}{t} \left\{ (1) - (t-1) \left(\frac{1}{t}\right) \right\} &= \frac{2}{t^2} \\ R'_{i,2}(t) &= 0, & i &\neq 2, 3 \end{aligned}$$

Итак

$$P'(t) = \frac{2}{t^2}(B_3 - B_2)$$

На интервале

$$2 \leq t < 3$$

$$\begin{aligned} N_{3,2}(t) &= 3 - t; & N_{4,2}(t) &= t - 2; & N_{i,2}(t) &= 0, & i &\neq 3, 4 \\ N'_{3,2}(t) &= -1; & N'_{4,2}(t) &= 1; & N'_{i,2}(t) &= 0, & i &\neq 3, 4 \end{aligned}$$

При этом

$$\begin{aligned} \sum_{i=1}^{n+1} h_i N_{i,2} &= 1; & \sum_{i=1}^{n+1} h_i N'_{i,2} &= 0; \\ R'_{3,2}(t) &= N'_{3,2}(t) = -1; & R'_{4,2}(t) &= N'_{4,2}(t) = 1; \\ R'_{i,2}(t) &= 0, & i &\neq 3, 4 \end{aligned}$$

То есть

$$P'(t) = B_4 - B_3$$

При $t = (3 - \varepsilon)$, $\varepsilon \rightarrow 0$, получаем тот же результат, что и при $t = 3$.

Как уже отмечалось, с помощью рациональных В-сплайнов можно представить все конические сечения. Кроме того, это единственное математическое описание, способное задать произвольную кривую, состоящую из конических сечений. Коническим сечениям соответствуют квадратичные уравнения, поэтому удобно сначала рассмотреть квадратичный рациональный В-сплайн ($k = 3$), заданный тремя вершинами ($n + 1 = 3$) с узловым вектором $[X] = [0 \quad 0 \quad 1 \quad 1 \quad 1]$:

$$P(t) = \frac{h_1 N_{1,3}(t) B_1 + h_2 N_{2,3}(t) B_2 + h_3 N_{3,3}(t) B_3}{h_1 N_{1,3}(t) + h_2 N_{2,3}(t) + h_3 N_{3,3}(t)}, \quad (5-129)$$

что фактически представляет собой рациональную кривую Безье третьего порядка (см. рис. 5-67). Предположим, что $h_1 = h_3 = 1$. Тогда уравнение (5-129) принимает вид

$$P(t) = \frac{N_{1,3}(t) B_1 + h_2 N_{2,3}(t) B_2 + N_{3,3}(t) B_3}{N_{1,3}(t) + h_2 N_{2,3}(t) + N_{3,3}(t)}. \quad (5-130)$$

При $h_2 = 0$ точки B_1 и B_3 соединены прямой линией. При $h_2 \rightarrow \infty$ повторяется контур определяющего многоугольника. Если $h_2 = 0$ и $t = 1/2$, достигается серединная точка отрезка $B_1 B_2$, обозначенная на рис. 5-67 буквой M . Аналогично при $h_2 \rightarrow \infty$, $t = 1/2$, достигается точка B_2 . Для $0 < h_2 < \infty$ точка S , соответствующая точке кривой $P(t)$ при $t = 1/2$, движется по прямой линии между M и B_2 . S называется плечевой точкой. Значение h_2 определяет тип конического сечения. В работе [5-43] показано, что результат при

- $h_2 = 0$ — прямая линия,
- $0 < h_2 < 1$ — эллиптический сегмент,
- $h_2 = 1$ — параболический сегмент,
- $h_2 > 1$ — гиперболический сегмент.

Воспользуемся уравнением (5-84) и подставим $t = 1/2$ в уравнение (5-130):

$$P(t) = \frac{(1-t)^2 B_1 + 2h_2 t(1-t) B_2 + t^2 B_3}{(1-t)^2 + 2h_2 t(1-t) + t^2}.$$

При $t = 1/2$, $P(t) = S$, что дает

$$S = \frac{1}{1+h_2} \frac{B_1 + B_3}{2} + \frac{h_2}{1+h_2} B_2 = \frac{M}{1+h_2} + \frac{h_2}{1+h_2} B_2. \quad (5-131)$$

Запишем параметрическое уравнение прямой между M и B_2

$$S = (1-s)M + sB_2, \quad (5-132)$$

где s — параметр. Приравнивая коэффициенты уравнений (5-131) и (5-132), получаем

$$s = \frac{h_2}{1+h_2} \quad \text{и} \quad h_2 = \frac{s}{1-s} = \frac{M-S}{S-B_2}. \quad (5-133)$$

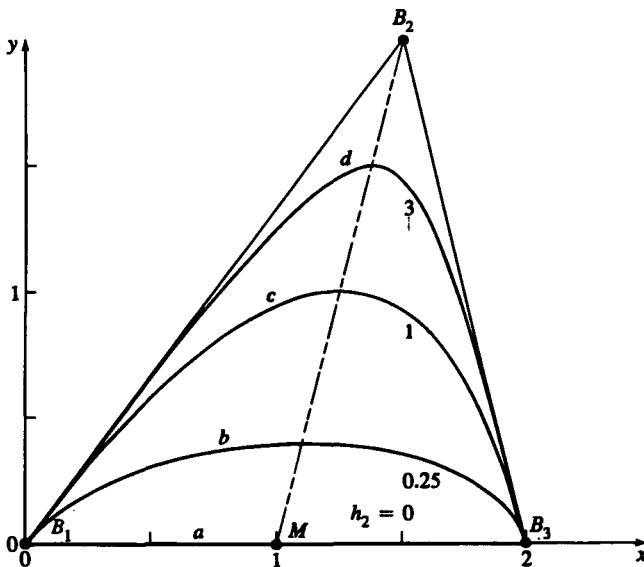


Рис. 5-67 Конические сечения, заданные рациональными B-сплайнами (кривыми Безье); (а) прямая линия, $h_2 = 0$; (б) эллипс, $h_2 = 0.25$; (в) парабола, $h_2 = 1$; (г) гипербола, $h_2 = 3$.

Параметр s определяет форму кривой и тип конического сечения, т. е. это удобное средство разработки.

Так как окружность является частным случаем эллипса, можно подобрать такое значение h_2 , что уравнение (5-130) будет уравнением дуги окружности. Из-за симметрии B_1 , B_2 и B_3 для дуги окружности формируют равнобедренный треугольник, как показано на рис. 5-68. Требуемое значение h_2 , определяется из рис. 5-68.

Треугольник $B_1B_2B_3$ равнобедренный, поэтому S — максимум кривой и касательная в точке S параллельна линии B_1B_3 . Треугольник B_1qS также равнобедренный с равными углами $\angle SB_1q$ и $\angle B_1Sq$ у основания, обозначенными $\theta/2$ на рис. 5-68. Из параллельности касательной S линии B_1B_3 следует, что углы $\angle qSB_1$ и $\angle SB_1M$ равны. Поэтому угол $\angle SB_1M = e/2$ равен половине угла у основания треугольника B_1 , B_2 и B_3 .

Отсюда и из уравнения (5-133) получаем

$$h_2 = \frac{M - S}{S - B_2} = \frac{e \operatorname{tg}(\frac{\theta}{2})}{f \sin \theta - e \operatorname{tg}(\frac{\theta}{2})}.$$

Подставляя тождество $(\theta/2) = \sin \theta / (1 + \cos \theta)$, получаем

$$h_2 = \frac{\frac{e \sin \theta}{1 + \cos \theta}}{f \sin \theta - \frac{e \sin \theta}{1 + \cos \theta}} = \frac{e}{f(1 + \cos \theta) - e} = \frac{e}{f} = \cos \theta \quad (5-134)$$

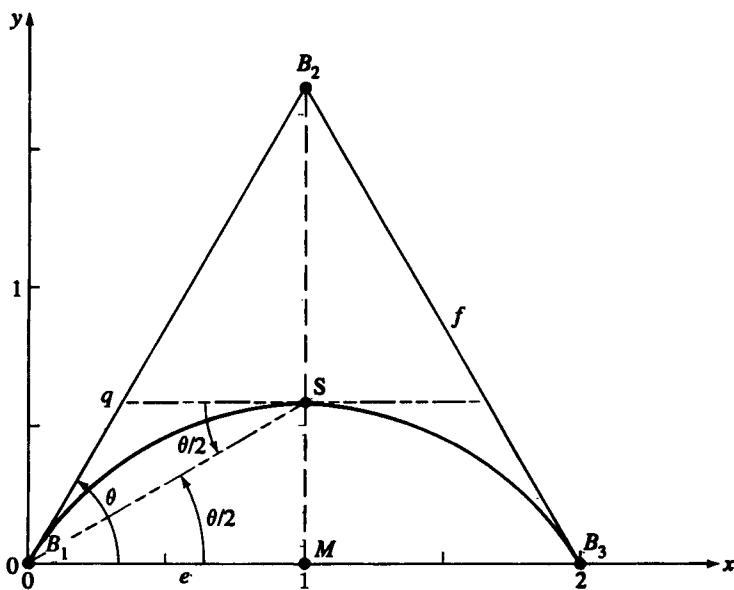


Рис. 5-68 Дуга окружности, построенная как рациональный В-сплайн.

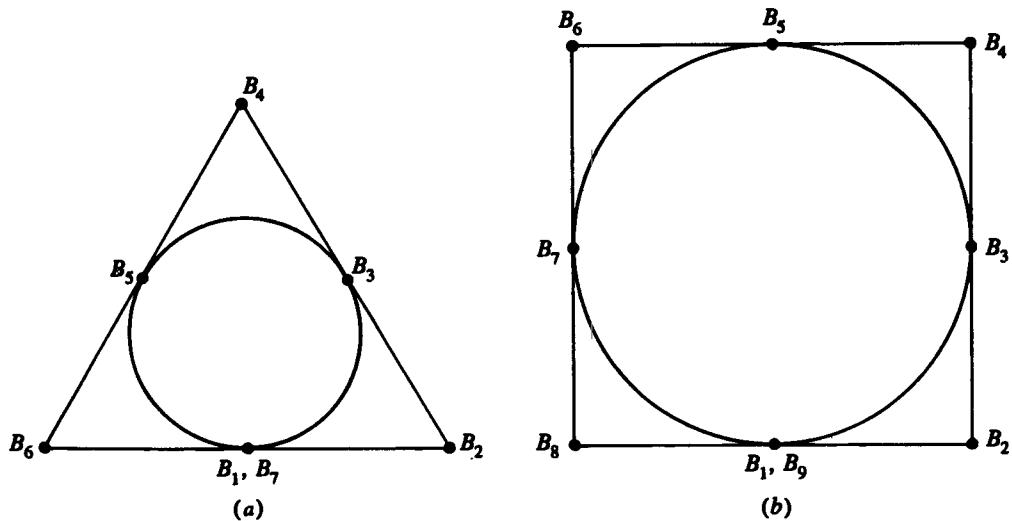


Рис. 5-69 Рациональные В-сплайны — окружности. (а) Три сегмента по 120° ; (б) четырех сегмента по 90° .

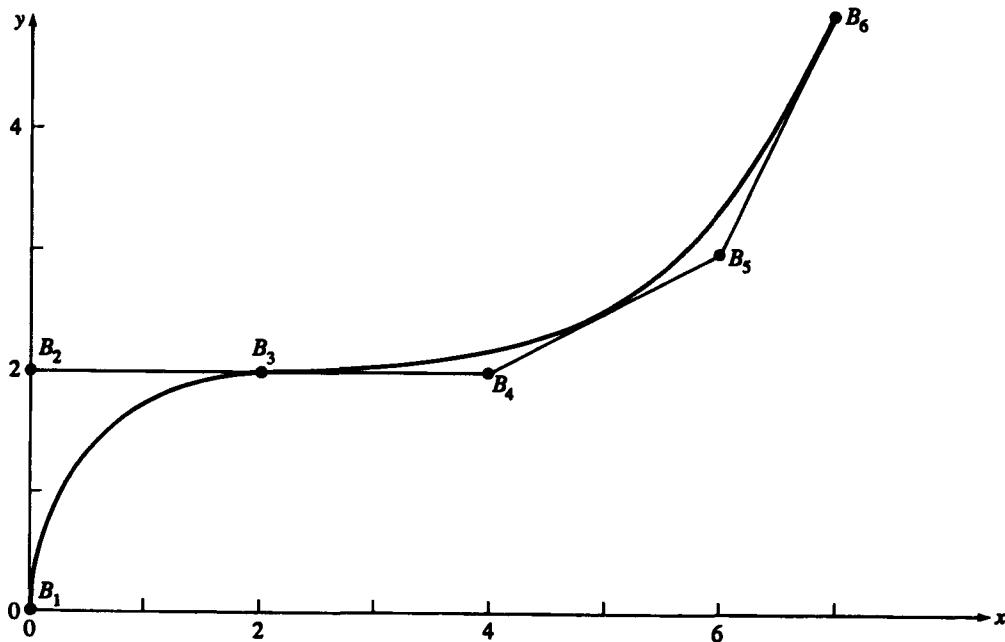


Рис. 5-70 Сопряжение рациональных В-сплайнов.

Дуга покрывает часть окружности для удвоенного угла θ . Для дуги 120° угол $\theta = 60^\circ$ и $h_2 = 1/2$. Для данного случая радиус окружности равен $2(S - M)$.

Полная окружность получается соединением нескольких сегментов. В частности, окружность можно составить из трех сегментов рационального квадратичного В-сплайна, каждый из которых покрывает 120° . Определяющие точки составляют равносторонний треугольник, как показано на рис. 5-69а. Неравномерный узловой вектор и вектор однородных координат:

$$\begin{aligned}[X] &= [0 \ 0 \ 0 \ 1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 3] \\ [H] &= [1 \ 1/2 \ 1 \ 1/2 \ 1 \ 1/2 \ 1]\end{aligned}$$

Аналогично, полная окружность также задается четырьмя рациональными квадратичными сегментами В-сплайна, каждый по 90° , где определяющие вершины составляют квадрат, рис. 5-69б. Неравномерный узловой вектор и вектор однородных координат:

$$\begin{aligned}[X] &= [0 \ 0 \ 0 \ 1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 4 \ 4 \ 4] \\ [H] &= [1 \ \sqrt{2}/2 \ 1 \ \sqrt{2}/2 \ 1 \ \sqrt{2}/2 \ 1 \ \sqrt{2}/2 \ 1]\end{aligned}$$

Рассмотрим пример построения конических сечений с помощью В-сплайна и объединения их в произвольные кривые.

Пример 5-22 Конические сечения, использующие рациональные В-сплайны

Построить единственный рациональный В-сплайн третьего порядка, со- прягающий 90° дугу окружности, заданную квадратичным рациональным В-сплайном с определяющими вершинами $B_1[0 \ 0]$, $B_2[0 \ 2]$, $B_3[2 \ 2]$ с квадратическим рациональным В-сплайном третьего порядка, заданным точка- ми $B_3[2 \ 2]$, $B_4[4 \ 2]$, $B_5[6 \ 3]$, $B_6[7 \ 5]$ с $h_i = 1$, $4 \leq i \leq 7$.

90° дуга окружности имеет узловой вектор $[0 \ 0 \ 0 \ 1 \ 1 \ 1]$ и век- тор однородных координат $[1 \ \sqrt{2}/2 \ 1]$. Соответствующие векторы рацио- нального В-сплайна, заданного точками $B_3B_4B_5B_6$: $[0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2]$ и $[1 \ 1 \ 1 \ 1]$.

Неравномерный узловой вектор комбинированной прямой: $[X] = [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 2 \ 3 \ 3 \ 3]$; вектор однородных координат: $[1 \ \sqrt{2}/2 \ 1 \ 1 \ 1 \ 1 \ 1]$. Ре- зультат изображен на рис. 5-70.

5-14 ЛИТЕРАТУРА

- 5-1 Böhm, W., Farin, G., and Kahmann, J. «A Survey of Curve and Surface Methods in CAGD», *Comp. Aid. Geom. Des.*, Vol. 1, pp. 1–60, 1984.
- 5-2 Higdon, A., Olisen, E., Stiles, W., and Weese, J. *Mechanics of Materials*, 2d ed., John Wiley & Sons, New York, 1967.
- 5-3 South, N.E., and Kelly, J.P., «Analytic Surface Methods», Ford Motor Company N/C Development Unit, Product Engineering Office, December 1965.
- 5-4 Nutbourne, A.W., «A Cubic Spline Package Part 2 — The Mathematics», *Comp. Aid. Des.*, Vol. 5, No. 1, January 1973.
- 5-5 Adams, J. A., «A Comparison of Methods for Cubic Spline Curve Fitting», *Comp. Aid. Des.*, Vol. 6, pp. 1–9, 1974.
- 5-6 Schweikert, D. C., «An Interpolation Curve Using a Spline in Tension», *J. Math. Phys.*, Vol. 45, pp. 312–317, 1966.
- 5-7 Cline, A. K., «Curve Fitting Using Splines Under Tension», *Atmos. Tech.*, No. 3, pp. 60–65, 1973.
- 5-8 Nielson, G., «Some Piecewise Polynomial Alternatives to Splines Under Tension», *Computer Aided Geometric Design*, Barnhill, R. E., and Riesenfeld, R. F. (eds.), pp. 209–235, Academis Press, New York, 1974.
- 5-9 Overhauser, A. W., «Analytic Definition of Curves and Surfaces by Parabolic Blending», Tech. Rep. No. SL68-40, Ford Motor Company Scientific Laboratory, May 8, 1968.
- 5-10 Brewer, J. A., and Anderson, D. C., «Visual Interaction with Overhauser Curves and Surfaces», *Comp. Graph.*, Vol. 11, pp. 132–137, 1977 (SIGGRAPH 77).
- 5-11 Bézier, P. E., «How Renault Uses Numerical Control for Car Body Design and Tooling», SAE paper 680010, Society of Automotive Engineers' Congress, Detroit, MI, 1968.
- 5-12 Bézier P. E., «Example of an Existing System in the Motor Industry: The Unisurf System», *Proc. Roy. Soc. (London)*, Vol. A321, pp. 207–218, 1971.
- 5-13 Bézier, P. E., *Emploi des Machines à Commande Numérique*, Masson et Cie, Paris, 1970. Translated by Forrest, A. R., and Pankhurst, A. F., as Bézier, P. E., *Numerical Control Mathematics and Applications*, John Wiley & Sons, London, 1972.

- 5-14 Forrest, A. R., «Interactive Interpolation and Approximation by Bézier Polynomials», *Comp. J.*, Vol. 15, pp. 71–79, 1972.
- 5-15 Gordon, W. J., and Riesenfeld, R. F., «Bernstein-Bézier Methods for the Computer Aided Design of Free-from Curves and Surfaces», *J. ACM*, Vol. 21, pp. 293–310, 1974.
- 5-16 Bartels, R. H., Beatty, J. C., and Barsky, B. A., *Splines for Use in Computer Graphics & Geometric Modeling*, Morgan Kaufman, Los Altos, CA, 1987.
- 5-17 Barsky, B. A., «Arbitrary Subdivision of Bézier Curves», TR UCB/CSD 85/265, Comp. Asi. Div., Univ. of Cal., Berkeley, November 1985.
- 5-18 Clark, J. H., «A Fast Scanline Algorithm for Rendering Parametric Surfaces», *Comp. Graph.*, Vol. 14, pp. 7–12, 1979 [addendum to SIGGRAPH 79 Conf. Proc. — «papers to be published in the CACM»].
- 5-19 Cohen, E., and Riesenfeld, R. F., «General Matrix Representations for Bézier and B-Spline Curves», *Comp. in Indus.*, Vol. 3, pp. 9–15, 1982.
- 5-20 Schoenberg, I. J., «Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions», *Q. Appl. Math.*, Vol. 4, pp. 45–99; pp. 112–141, 1946.
- 5-21 Cox, M. G., «The Numerical Evaluation of B-Csplines», National Physical Laboratory DNAC 4, August 1971.
- 5-22 de Boor, C., «On Calculation with B-splines», *J. Approx. Theory*, Vol. 6, pp. 50–62, 1972.
- 5-23 Riesenfeld, R. F., «Application of B-Spline Approximation to Geometric Problems of Computer Aided Design», PhD dissertation, Syracuse Univ., Syracuse, NY, 1972. Also available as U. of Utah, UTEC-CSc-73-126, March 1973.
- 5-24 Barsky, B. A., «End Conditions and Boundary Conditions for Uniform B-Spline Curve and Surface Representations», *Comp. in Indus.*, Vol. 3, pp. 17–29, 1982.
- 5-25 Hartley, P. J., and Judd, C. J., «Parameterization of Bézier Type B-spline Curves and Surfaces», *Comp. Aid. Des.*, Vol. 10, pp. 130–134, 1978.
- 5-26 Rogers, D. F., and Fog, N. G., «Constrained B-spline Curve and Surface Fitting», *CADJ*, Vol. 21, pp. 641–648, 1989.
- 5-27 Cohen, E., Lyche, T., and Schumacher, L. L., «Zlgorithms for Degree-Raising of Splines», *ACM Trans. on Graph.*, Vol. 4, pp. 171–181, 1985.
- 5-28 Cohen, E., Lyche, T., and Riesenfeld, R. F., «Discrete B-Splines and Subdivision Techniques in Computer Aided Geometric Design and Computer Graphics», *Comp. Graph. Imag. Proc.*, Vol. 14, pp. 87–111, 1980.
- 5-29 Prautzsch, H., «A Short Proof of the Oslo Algorithm», *Comp. Aid. Geom. Des.*, Vol. 1, pp. 95–96, 1984.
- 5-30 Böhm, W., «Inserting New Knots into B-spline Curves», *Comp. Aid. Des.*, Vol. 12, pp. 199–201, 1980.
- 5-31 Böhm, W., and Prautzsch, H., «The Insertion Algorithm», *Comp. Aid. Des.*, Vol. 17, pp. 58–59, 1985.
- 5-32 Coons, S. A., «Surfaces for Computer Aided Design of Space Forms», MIT Project MAC-TR-41, 1967.
- 5-33 Farouki, R., and Hinds, J., «A Hierarchy of Geometric Forms», *IEEE Comp. Graph. & Appl.*, Vol. 5, No. 5, pp. 51–78, 1985.
- 5-34 Piegl, L., «A Geometric Investigation of the Rational Bézier Scheme of Computer Aided Design», *Comp. in Indust.*, Vol. 7, pp. 401–410, 1986.
- 5-35 Forrest, A. R., «The Twisted Cubic Curve: A Computer Aided Geometric Design Approach», *Comp. Aid. Des.*, Vol. 12, pp. 165–172, 1980.
- 5-36 Böhm, W., «On Cubics: A Survey», *Comp. Graph. Imag. Proc.*, Vol. 19, pp. 201–226, 1982.

- 5-37 Forrest, A. R., «Curves and Surfaces for Computer-Aided Design», PhD Dissertation, Cambridge Univ., 1968.
- 5-38 Ball, A. A., «Consurf I-III», *Comp. Aid. Des.*, Vol. 6, pp. 243–249, 1974; Vol. 7, pp. 237–242, 1975; Vol. 9, pp. 9–12, 1977.
- 5-39 Versprille, K. J., «Computer-Aided Design Applications of the Rational B-spline Approximation Form», PhD dissertation, Syracuse Univ., Syracuse, NY, February 1975.
- 5-40 Tiller, W., «Rational B-splines for Curve and Surface Representation», *IEEE Comp. Graph. & Appl.*, Vol. 3, No. 6, pp. 61–69, September 1983.
- 5-41 Piegl, L., and Tiller, W., «Curve and Surface Constructions Using Rational B-splines», *Comp. Aid. Des.*, Vol. 19, pp. 485–498, 1987.
- 5-42 IGES ‘Initial Graphics Exchange Specifications, Version 3.0’, Doc. No. NB-SIR 86-3359 Nat. Bur. of Stds., Gaithersburg, MD, USA, 1986.
- 5-43 Lee, E., «Rational Bézier Representation for Conics», in *Geometric Modeling*, Farin, G. (ed.), SIAM, pp. 3–27, 1986.

Глава 6

ПОВЕРХНОСТИ

6-1 ВВЕДЕНИЕ

Поверхности и их описание играют важную роль в конструировании и производстве. Очевидными примерами этого являются разработка и производство автомобильных кузовов, корабельных корпусов, авиационных фюзеляжей и крыльев; пропеллеров, турбин, компрессоров и лопастей вентиляторов; посуды, мебели и обуви. В этом случае сущность конструирования либо по функциональным, либо по эстетическим причинам составляет форма или геометрия поверхности. Описание поверхности также играет важную роль в представлении данных, полученных в медицине, геологии, физике и других естественных науках.

Традиционным способом представления поверхности является использование нескольких ортогональных проекций. По существу, поверхность задается сеткой ортогональных плоских кривых, лежащих на секущих плоскостях, и несколькими ортогональными проекциями определенных «характерных» пространственных линий (см. рис. 5-1). Эти кривые первоначально могут быть созданы на бумаге либо же взяты (оцифрованы) из трехмерной модели, например в автомобильной промышленности дизайнерами традиционно используется глиняная модель.

В машинной графике и автоматизированном проектировании выгодно разрабатывать «настоящую» трехмерную математическую модель поверхности. Такая модель позволяет на ранних стадиях и относительно легко провести анализ характеристик поверхности, например кривизны, или физических количественных характеристик, зависящих от поверхности, например объема, площади поверхности, момента инерции и т. д. Упрощается визуализация поверхности (см. [6-1]), применяемая для разработки или контроля за ходом разработки. Далее, по сравнению с традиционным методом, использующим сетку линий, также существенно упрощается генерация необходимой для изготовления поверхности информации, например управляемых программ для станка с числовым программным управлением. В ранних работах Безье [6-2], Сабина [6-3], Петерса [6-4] и других авторов продемонстрирована осуществимость этого подхода. Разработанные в последнее время методы описания поверхностей достигли такой стадии развития, что позволяют «почти» исключить традиционное описание поверхности с помощью сетки линий.

Существуют две основные идеи, лежащие в основе методов описания поверхностей. В первой, связанной в основном с именем Кунса, математическую поверхность стараются создать по заранее известным данным. Во второй, связанной в основном с именем Безье, математическую поверхность стараются создать *ab initio* (с самого начала). В первое время отрасли, связанные с числовыми параметрами, например конструирование, тяготели к первому подходу, тогда как отрасли, учитывающие визуальные, осознательные или эстетические факторы, например дизайнеры и художники, тяготели ко второму. Последние работы Роджерса ([6-5] — [6-7]) об интерактивных системах разработки корпусов кораблей и Коэна ([6-8]) о разработке поверхностей показали, что эти два подхода совместимы.

В следующих разделах представлены основы математического описания параметрических поверхностей, используемых в машинной графике и автоматизированном проектировании.

6-2 ПОВЕРХНОСТИ ВРАЩЕНИЯ

Возможно самым простым способом создания трехмерной поверхности является вращение двумерного объекта, например прямой или плоской кривой вокруг оси в пространстве. Такие поверхности называются поверхностями вращения. Сначала для простоты предположим, что ось вращения совпадает с осью x и положительно направлена. Предположим также, что объекты вращения — отрезок, прямая или плоская кривая — лежат на плоскости xy . Позднее мы рассмотрим метод, позволяющий избавиться от этих ограничений.

Самый простой объект, который можно вращать вокруг оси, — это точка. При условии что точка не лежит на оси, вращение на угол 2π (360°) породит окружность. Поворот на меньший угол даст дугу окружности.

Следующим по сложности является отрезок, параллельный, но не совпадающий с осью вращения. Вращение на угол 2π (360°) породит в этом случае круговой цилиндр. Радиусом этого цилиндра является длина перпендикуляра, опущенного с отрезка на ось вращения. Длина цилиндра равна длине отрезка. Пример изображен на рис. 6-1.

Если отрезок и ось вращения компланарны и отрезок не параллелен оси вращения, то в результате вращения вокруг оси на угол 2π (360°) мы получим усеченный круговой конус. Радиусы оснований усеченного конуса — длины перпендикуляров, опущенных с концов отрезка на ось вращения. Высота конуса — это длина спроектированного на ось вращения отрезка. Пример изображен на рис. 6-2.

И снова, если отрезок и ось вращения компланарны и отрезок перпендикулярен оси вращения, то в результате вращения на угол 2π (360°) мы получим плоский диск. Если отрезок пересекает (или касается) ось вращения, то получится сплошной диск, в противном случае диск будет иметь круглое отверстие. Примеры изображены на рис. 6-3.

И наконец, если отрезок наклонен к оси вращения, т. е. некомпланарен, то вращение на угол 2π (360°) породит однополостный гиперболоид (см. разд. 6-4 и 6-7).

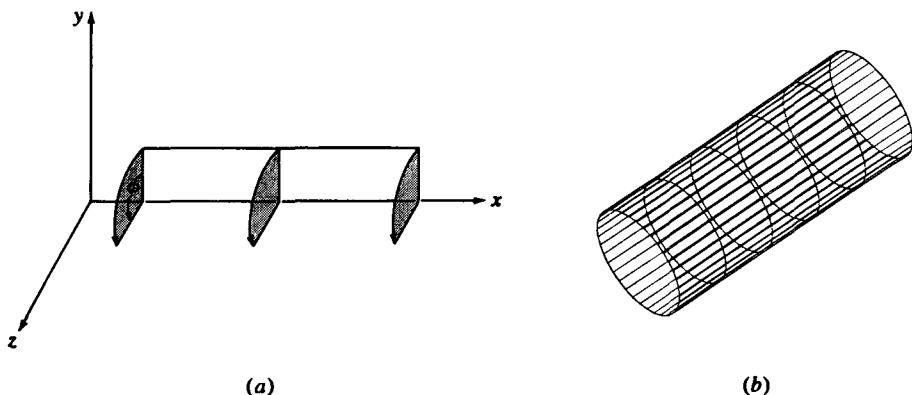


Рис. 6-1 Цилиндрическая поверхность вращения. (а) Схема построения; (б) результат.

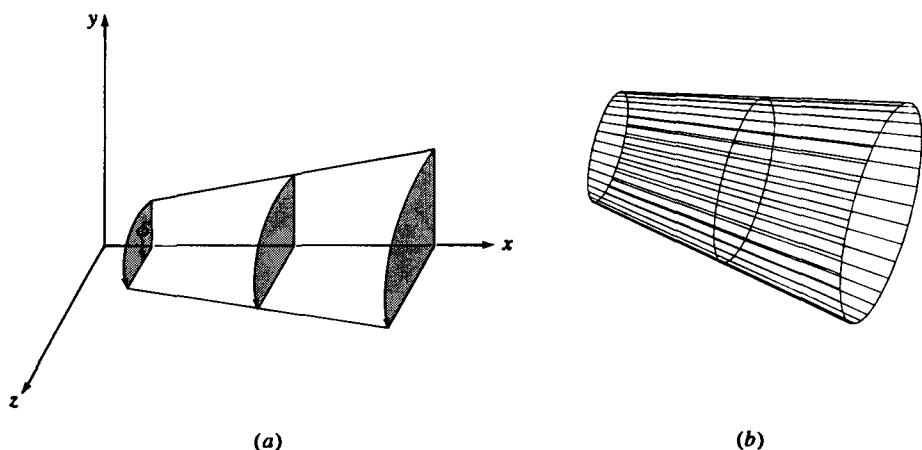


Рис. 6-2 Коническая поверхность вращения. (а) Схема построения; (б) результат.

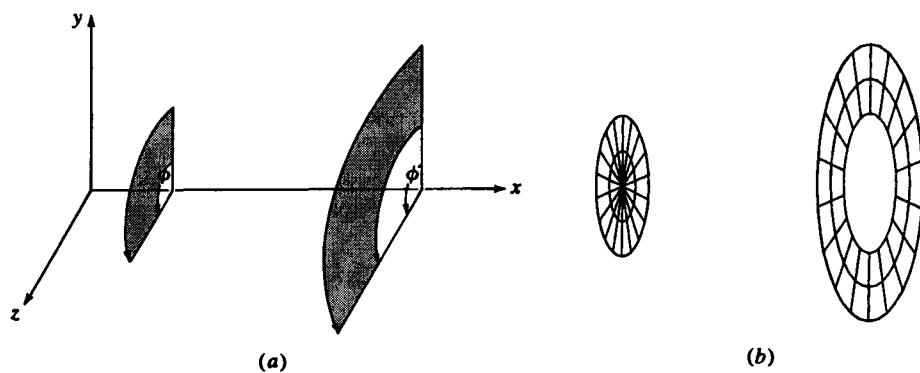


Рис. 6-3 Диск в качестве поверхности вращения. (а) Схема построения; (б) результат.

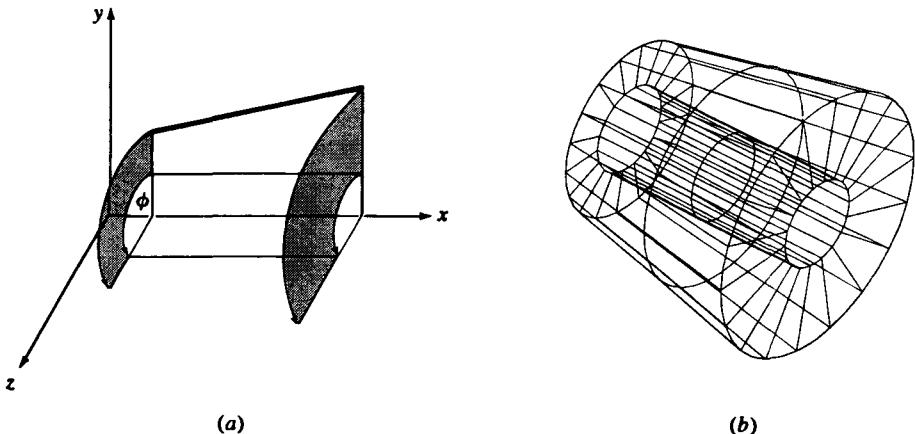


Рис. 6-4 Поверхность вращения из замкнутой ломаной. (а) Схема построения; (б) результат.

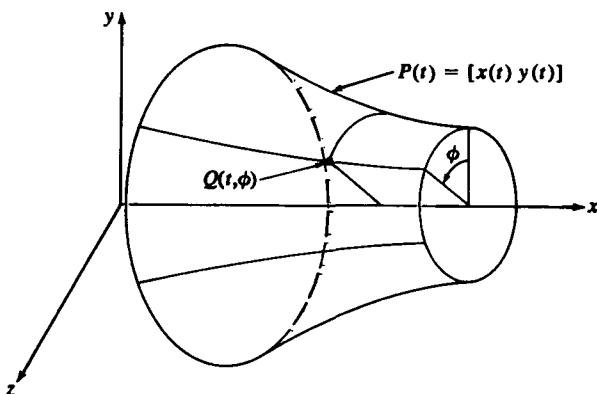


Рис. 6-5 Бипараметрическая поверхность вращения.

Для создания поверхностей вращения могут быть также использованы замкнутые и незамкнутые ломаные. На рис. 6-4 представлен конус с цилиндрическим отверстием.

Параметрическое уравнение точки на поверхности вращения можно получить, если вспомнить, что параметрическое уравнение вращаемого объекта, например

$$P(t) = [x(t) \quad y(t) \quad z(t)] \quad 0 \leq t \leq t_{\max},$$

есть функция одного параметра t . Вращение вокруг оси приводит к тому, что координаты зависят также от угла поворота. Таким образом, точка на поверхности вращения определяется двумя параметрами t и ϕ . Как показано на рис. 6-5, это бипараметрическая функция.

Для рассматриваемого частного случая, т. е. вращения вокруг оси x объекта, расположенного в плоскости xy , уравнение поверхности записывается¹

$$Q(t, \phi) = [x(t) \quad y(t) \cos \phi \quad y(t) \sin \phi] \quad (6-1)$$

Заметим, что здесь координата x не меняется. В качестве иллюстрации приведем пример.

Пример 6-1 Простая поверхность вращения

Рассмотрим отрезок с концами $P_1[1 \ 1 \ 0]$ и $P_2[6 \ 2 \ 0]$, лежащий в плоскости xy . Вращение отрезка вокруг оси x породит коническую поверхность. Определим на поверхности координаты точки с параметрами $t = 0.5$, $\phi = \pi/3$ (60°).

Параметрическое уравнение отрезка, соединяющего P_1 и P_2 , имеет вид

$$P(t) = [x(t) \quad y(t) \quad z(t)] = P_1 + (P_2 - P_1)t \quad 0 \leq t \leq 1$$

с декартовыми координатами

$$\begin{aligned} x(t) &= x_1 + (x_2 - x_1)t = 1 + 5t, \\ y(t) &= y_1 + (y_2 - y_1)t = 1 + t, \\ z(t) &= z_1 + (z_2 - z_1)t = 0. \end{aligned}$$

Используя уравнение (6-1), получим точку $Q(1/2, \pi/3)$ на поверхности вращения

$$\begin{aligned} Q(1/2, \pi/3) &= [1 + 5t \quad (1 + t) \cos \phi \quad (1 + t) \sin \phi] \\ &= \left[\frac{7}{2} \quad \frac{3}{2} \cos\left(\frac{\pi}{3}\right) \quad \frac{3}{2} \sin\left(\frac{\pi}{3}\right) \right] \\ &= \left[\frac{7}{2} \quad \frac{3}{4} \quad \frac{3\sqrt{3}}{4} \right] = [3.5 \quad 0.75 \quad 1.3]. \end{aligned}$$

Вращение плоских кривых также порождает поверхности вращения. Как показано на рис. 6-6а, сфера получается в результате вращения вокруг оси x расположенной в плоскости xy полуокружности, центрированной относительно начала координат. Вспомнив параметрическое уравнение окружности (см. разд. 4-5)

$$x = r \cos \theta \quad 0 \leq \theta \leq \pi \quad (4-4)$$

$$y = r \sin \theta$$

получим параметрическое уравнение сферы

$$\begin{aligned} Q(\theta, \phi) &= [x(\theta) \quad y(\theta) \cos \phi \quad y(\theta) \sin \phi] \\ &= [r \cos \theta \quad r \sin \theta \cos \phi \quad r \sin \theta \sin \phi] \quad 0 \leq \theta \leq \pi, \\ &\quad 0 \leq \phi \leq 2\pi \end{aligned} \quad (6-2)$$

¹ Заметим, что $Q(t, \phi)$ является вектор-функцией, т. е. в векторной форме

$$Q(t, \phi) = x(t)\mathbf{i} + y(t) \cos \phi \mathbf{j} + y(t) \sin \phi \mathbf{k},$$

где $\mathbf{i}, \mathbf{j}, \mathbf{k}$ – единичные векторы в x, y, z направлениях, соответственно.

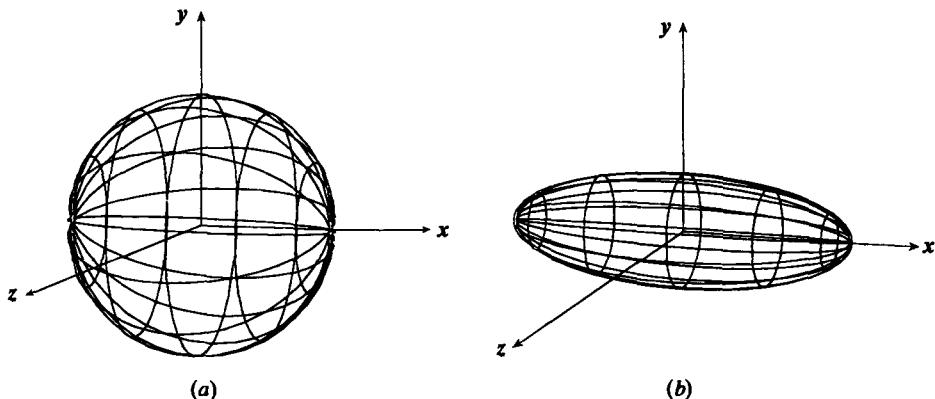


Рис. 6-6 Поверхности вращения. (a) Сфера; (b) эллипсоид.

Если вместо окружности подставить параметрическое уравнение центрированного полуэллипса, расположенного в плоскости xy , получится эллипсоид вращения. Напомнив параметрическое уравнение полуэллипса (см. разд. 4-6)

$$\begin{aligned} x &= a \cos \theta & 0 \leq \theta \leq \pi \\ y &= b \sin \theta \end{aligned} \quad (4-6)$$

получим для любой точки эллипсоида следующее параметрическое уравнение:

$$\begin{aligned} Q(\theta, \phi) = [a \cos \theta & b \sin \theta \cos \phi & b \sin \theta \sin \phi] & 0 \leq \theta \leq \pi, \\ & 0 \leq \phi \leq 2\pi. \end{aligned} \quad (6-3)$$

При $a = b = r$ уравнение (6-3) превращается в уравнение (6-2) для сферы. Эллипсоид вращения показан на рис. 6-6б.

Если ось вращения не проходит через центр окружности или эллипса, то в результате вращения получается тор с сечением в виде окружности или эллипса, соответственно. Параметрическое уравнение эллипса на плоскости xy с центром, не совпадающим с началом координат, выглядит так

$$\begin{aligned} x &= h + a \cos \theta & 0 \leq \theta \leq 2\pi, \\ y &= k + b \sin \theta, \end{aligned}$$

где (h, k) — это x, y — координаты центра эллипса, тогда параметрическое уравнение для любой точки тора имеет вид:

$$Q(\theta, \phi) = [h + a \cos \theta & (k + b \sin \theta) \cos \phi & (k + b \sin \theta) \sin \phi], \quad (6-4)$$

где $0 \leq \theta \leq 2\pi, 0 \leq \phi \leq 2\pi$. Если $a = b = r$, то уравнение (6-4) задает тор с сечением в виде окружности. Если $a \neq b$, то получится тор с сечением в виде эллипса. На рис. 6-7 представлены оба типа торов.

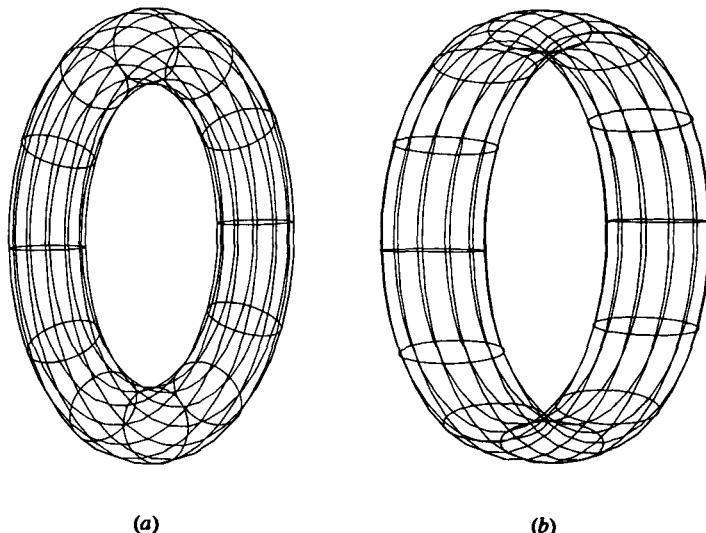


Рис. 6-7 Торы. (a) С сечением в виде окружности; (b) с сечением в виде эллипса.

Параболоид вращения получается при вращении параметрической параболы (см. разд. 4-7)

$$\begin{aligned} x &= a\theta^2 & 0 \leq \theta \leq \theta_{\max} \\ y &= 2a\theta \end{aligned} \quad (4-9)$$

вокруг оси x . Параметрическая поверхность задается уравнением

$$\begin{aligned} Q(\theta, \phi) &= [a\theta^2 \quad 2a\theta \cos \phi \quad 2a\theta \sin \phi] & 0 \leq \theta \leq \theta_{\max}, \\ & & 0 \leq \phi \leq 2\pi. \end{aligned} \quad (6-5)$$

Гиперболоид вращения получается при вращении параметрической гиперболы

$$\begin{aligned} x &= a \sec \theta & 0 \leq \theta \leq \theta_{\max} \\ y &= b \operatorname{tg} \theta \end{aligned} \quad (4-14)$$

вокруг оси x . Параметрическая поверхность задается уравнением

$$\begin{aligned} Q(\theta, \phi) &= [a \sec \theta \quad b \operatorname{tg} \theta \cos \phi \quad b \operatorname{tg} \theta \sin \phi], & 0 \leq \theta \leq \theta_{\max}, \\ & & 0 \leq \phi \leq 2\pi. \end{aligned} \quad (6-6)$$

Примеры показаны на рис. 6-8.

Для создания поверхности вращения можно использовать любую параметрическую кривую, например кубический сплайн, параболический сплайн, кривую Безье и В-сплайн. На рис. 6-9 изображена поверхность вращения, созданная из относительно простого параболического сплайна. На рис. 6-10 изображен бокал,

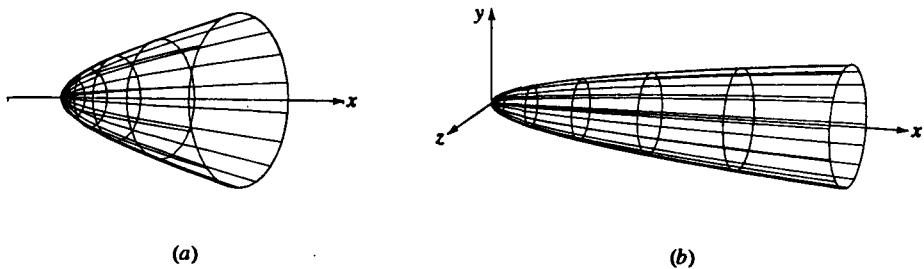


Рис. 6-8 Поверхности вращения. (a) Параболоид; (b) гиперболоид.

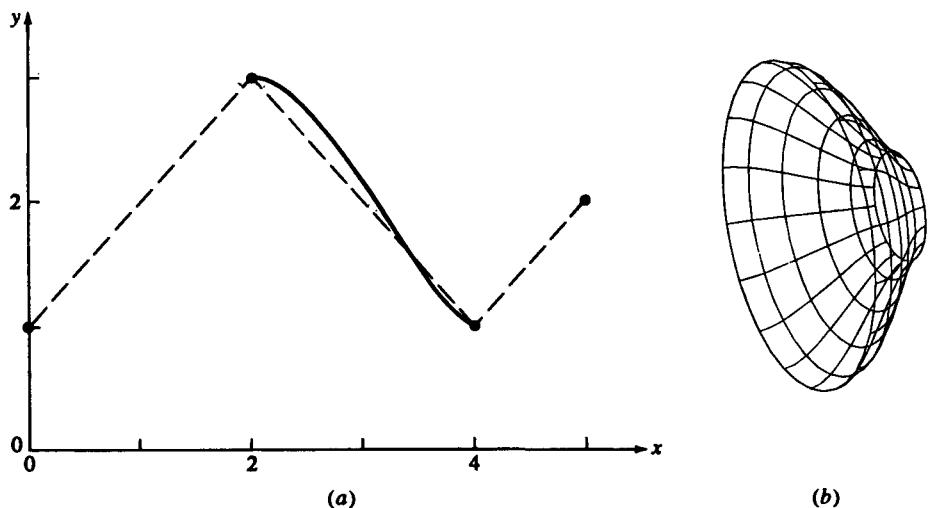


Рис. 6-9 Поверхность вращения из параболически интерполированной кривой. (a) Создание кривой; (b) поверхность.

созданный как поверхность вращения с помощью незамкнутого В-сплайна. Заметим, что бокал имеет как внутреннюю, так и внешнюю стороны. Вращение производится относительно оси y ¹.

¹ Метод, использованный при создании В-сплайна для бокала на рис. 6-10, может представлять определенный интерес. Сначала набросок бокала был сделан на миллиметровке. Затем тридцать четыре точки с этого рисунка были оцифрованы. С помощью описанного в разд. 5-11 итерационного процесса был получен исходный В-сплайн ($k = 4$, $n + 1 = 21$). После этого для окончательной разработки в интерактивную систему (см. [6-7]) были переданы определяющие точки кривой. Для изображенной поверхности вращения был использован В-сплайн, полученный из окончательных 25 вершин. Определяющие точки кривой: [0, 0.684], [0.302 0.684], [0.302 0.684], [1.598 -0.288], [1.088 0.405], [0.374 0.773], [0.848 0.993], [1.232 1.446], [1.451 1.875], [1.502 2.631], [1.448 3.308], [1.226 4.076], [1.397 4.449], [1.398 4.592], [1.268 4.572], [1.241 4.485], [1.109 4.170], [1.214 3.669], [1.313 2.889], [1.310 2.280], [1.172 1.575], [0.629 1.119], [0.281 0.882], [0.278 0.891], [0.897].

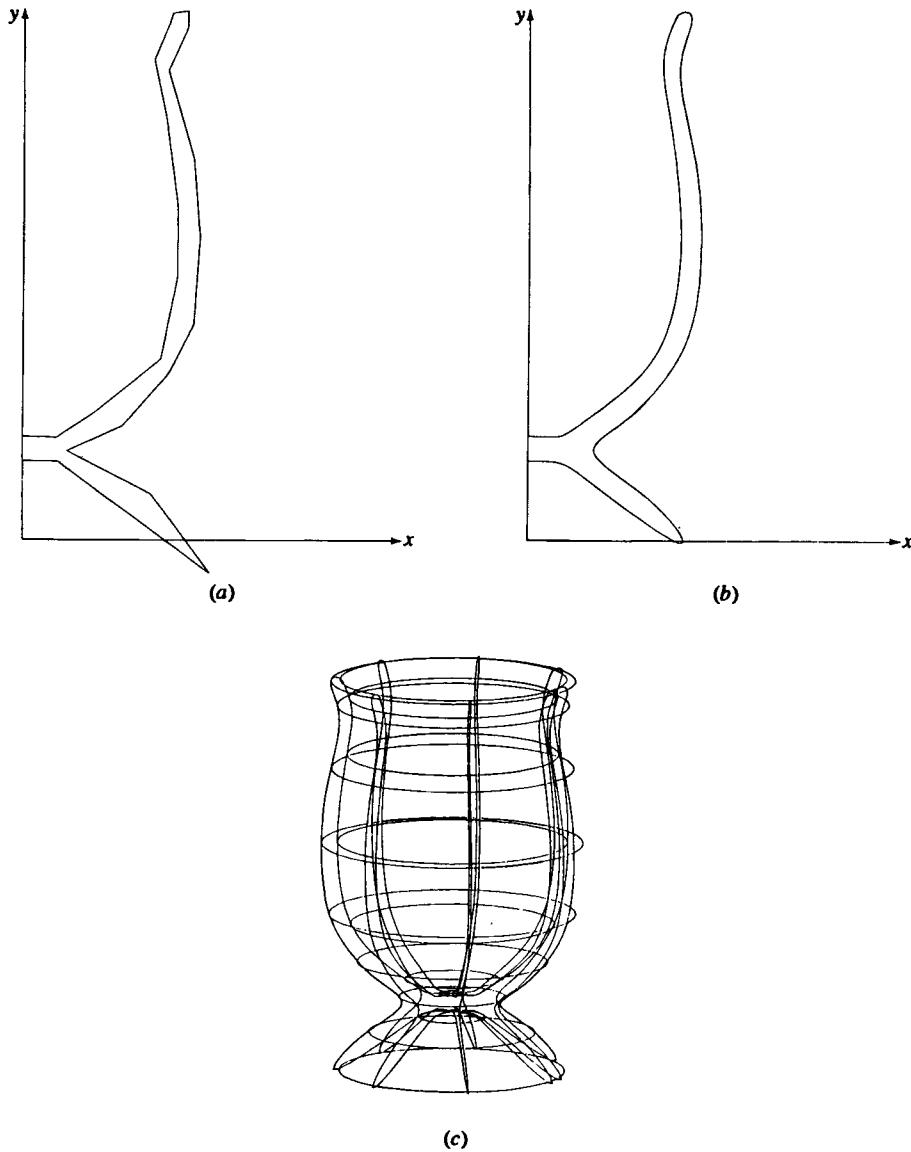


Рис. 6-10 В-сплайн поверхность вращения. (а) Вершины ломаной; (б) В-сплайн; (с) поверхность.

Напомним, что в матричной форме параметрическая пространственная кривая (см. уравнения (5-27), (5-44), (5-67) и (5-94)) задается следующим образом:

$$P(t) = [T][N][G],$$

где $[T]$, $[N]$ и $[G]$ – соответственно матрица параметров, матрица функций смещивания и геометрическая матрица. Таким образом, в общей форме матричное уравнение поверхности вращения записывается в виде:

$$Q(t, \phi) = [T][N][G][S], \quad (6-7)$$

где $[S]$ представляет вклад вращения вокруг оси на угол ϕ . Для частного случая вращения вокруг оси x имеем:

$$[S] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & \sin \phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6-8)$$

Эти методы иллюстрируются в следующем примере.

Пример 6-2 Поверхность вращения, созданная по параболической кривой

Рассмотрим параболическую кривую, заданную точками $P_1[0 \ 1 \ 0]$, $P_2[2 \ 3 \ 0]$, $P_3[4 \ 1 \ 0]$, $P_4[5 \ 2 \ 0]$. Будем вращать эту кривую вокруг оси x на угол 2π , чтобы получить поверхность вращения. Найдем на поверхности точку с параметрами $t = 0.5$, $\phi = \pi/3$ (60°).

Из уравнений (6-7) и (6-8) получим параметрическое уравнение поверхности вращения

$$Q(t, \phi) = [T][A][G][S],$$

где $[S]$, $[T]$, $[A]$ и $[G]$ задаются уравнениями (5-44), (5-52) и (5-53) соответственно.

Конкретнее,

$$Q(t, \phi) = \left(\frac{1}{2}\right) [t^3 \ t^2 \ t \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 1 \\ 2 & 3 & 0 & 1 \\ 4 & 1 & 0 & 1 \\ 5 & 2 & 0 & 1 \end{bmatrix} \times$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & \sin \phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q(t, \phi) = \left(\frac{1}{2}\right) [t^3 \ t^2 \ t \ 1] \begin{bmatrix} -1 & 7 & 0 & 1 \\ 1 & -11 & 0 & 1 \\ 4 & 0 & 0 & 1 \\ 4 & 6 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & \sin \phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Для $t = 0.5$ и $\phi = \pi/3$ (60°)

$$Q(0.5, \pi/3) = \frac{1}{2} \begin{bmatrix} \frac{1}{8} & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} -1 & 7 & 0 & 1 \\ 1 & -11 & 0 & 1 \\ 4 & 0 & 0 & 1 \\ 4 & 6 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & \sqrt{3}/2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{49}{16} & \frac{33}{32} & \frac{33\sqrt{3}}{32} & 1 \end{bmatrix} = [3.0625 \ 1.03125 \ 1.786181 \ 1].$$

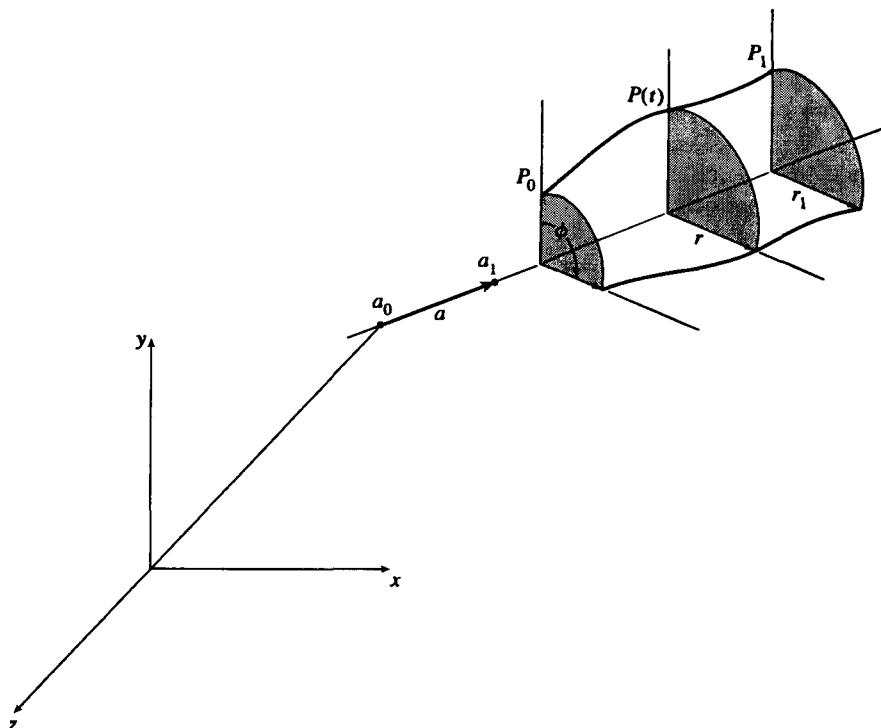


Рис. 6-11 Поверхность вращения вокруг произвольной оси.

Результаты изображены на рис. 6-9. Такая поверхность может быть результатом разработки кубка или даже газового канала двигателя или ракетного сопла.

Предыдущие результаты были получены путем вращения точки, отрезка, ломаной или кривой вокруг координатной оси, а именно вокруг оси x . К более общему случаю поворота вокруг произвольной оси в пространстве поверхность вращения, полученную в более удобной локальной системе координат, можно свести с помощью переносов и поворотов, приводящих поверхность в нужное положение.

На рис. 6-11 показана параметрическая кривая $P(t)$, повернутая вокруг произвольной оси в пространстве, проходящей через точки a_0 и a_1 и направленной от a_0 к a_1 . После того как поверхность создана в удобной системе координат для приведения поверхности вращения в нужное положение, нужно совершить следующие действия:

1. Перенести точку a_0 в начало координат.
2. Выполнить повороты, необходимые для совмещения осей a и $+z$ (см. разд. 5-9).
3. Повернуть вокруг оси y на угол $+90^\circ$ для совмещения осей a и x .¹

¹ Этот шаг необходим только для согласования с предыдущими рассуждениями. Поверхность вращения может быть легко построена для поворотов вокруг оси z .

Эти три шага необходимы только для того, чтобы найти обратное преобразование, размещающее поверхность вращения в нужном месте в трехмерном пространстве. Получив поверхность вращения вокруг оси x , приведем ее в нужное положение в пространстве:

1. Сдвинуть по оси x , чтобы переместить центр поверхности вращения в нужное положение на оси a .
2. Применить к поверхности вращения преобразование, обратное к суммарному преобразованию поворотов.
3. Применить к поверхности вращения обратный перенос точки a_0 .

Точка на поверхности вращения тогда задается уравнением:

$$Q(t, \phi) = [\bar{Q}][Tr][\bar{R}_y]^{-1}[R_y]^{-1}[R_x]^{-1}[Tr]^{-1}, \quad (6-9)$$

где $[Tr]$, $[R_x]$, $[R_y]$ задаются уравнениями (3-22) – (3-24). $[\bar{R}_y]^{-1}$ задается уравнением (3-8), и матрица $[\bar{Q}]$ задается в форме уравнения (6-7) с геометрической матрицей $[G]$, представленной в однородных координатах. $[S]$ теперь является матрицей 4×4 , заданной в виде

$$[S] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & \sin \phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6-10)$$

Данный метод иллюстрируется на следующем примере.

Пример 6-3 Поверхность вращения вокруг произвольной оси

Найдем координаты точки с параметрами $t = \pi/2$, $\phi = \pi/6$ на поверхности вращения, образованной вращением эллипса с главной осью, наклоненной относительно оси вращения. Ось вращения проходит через центр эллипса и лежит в плоскости эллипса. Угол наклона $i = \pi/4$. Полуоси эллипса $a = 5$, $b = 1$. Ось проходит через точки $a_0[0 \ 10 \ 10]$ и $a_1[10 \ 10 \ 0]$. Центр эллипса находится в точке a_1 .

Для начала найдем направляющие косинусы оси вращения (см. уравнение 3-26)

$$[c_x \ c_y \ c_z] = [1/\sqrt{2} \ 0 \ -1/\sqrt{2}]$$

и (см. уравнение 3-18)

$$d = \sqrt{c_y^2 + c_z^2} = 1/\sqrt{2}.$$

Таким образом, используя уравнения (3-22) – (3-24)

$$[M_1] = [Tr][R_x][R_y] =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -10 & -10 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 1 & 0 & 0 \\ -1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & -1 & 0 & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ -10/\sqrt{2} & 10 & 10/\sqrt{2} & 1 \end{bmatrix}$$

совместим ось вращения с осью z . Поворот вокруг оси y на угол 90° приводит к

$$[M_2] = [Tr][R_x][R_y][\bar{R}_y] =$$

$$= \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & \sqrt{2} & -1 & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ -10/\sqrt{2} & 10 & 10/\sqrt{2} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ -1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ 10/\sqrt{2} & 10 & 10/\sqrt{2} & 1 \end{bmatrix}.$$

Используя преобразование $[M_2]$ и однородные координаты, преобразуем центр эллипса, находящийся первоначально в a :

$$[10 \quad 10 \quad 10 \quad 1] \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 \\ 0 & \sqrt{2} & -1 & 0 \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} & 0 \\ -10/\sqrt{2} & 10 & 10/\sqrt{2} & 1 \end{bmatrix} = [20/\sqrt{2} \quad 0 \quad 0 \quad 1]$$

т.е. в $h = 20/\sqrt{2}$ на оси x .

Вспомнив пример 4-4 из разд. 4-6, получим параметрические уравнения центрированного относительно Начала координат эллипса, повернутого на угол i :

$$x = a \cos t \cos i - b \sin t \sin i \quad 0 \leq t \leq 2\pi$$

$$y = a \cos t \sin i + b \sin t \cos i$$

которые могут быть записаны в форме уравнения (6-7)

$$[\cos t \quad \sin t \quad 0 \quad 1] \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\cos i & \sin i & 0 & 0 \\ -\sin i & \cos i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Тогда поверхность вращения имеет вид:

$$[Q] = [T][N][G][S] =$$

$$= [\cos t \quad \sin t \quad 0 \quad 1] \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\cos i & \sin i & 0 & 0 \\ -\sin i & \cos i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & \sin \phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Для $a = 5$, $b = 1$, $i = \pi/4$

$$[\bar{Q}] = [\cos t \quad \sin t \quad 0 \quad 1] \begin{bmatrix} 5\sqrt{2}/2 & 5\sqrt{2}/2 & 0 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & \sin \phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Эта центрированная относительно начала координат поверхность вращения также показана на рис. 6-10.

Перенеся начало в (h, n) на оси x и заметив, что

$$[M_2]^{-1} = [\bar{R}_y]^{-1}[R_y]^{-1}[R_x]^{-1}[Tr]^{-1} =$$

$$= \begin{bmatrix} \sqrt{2}/2 & 0 & -\sqrt{2}/2 & 0 \\ 0 & -1 & 0 & 0 \\ -\sqrt{2}/2 & 0 & -\sqrt{2}/2 & 0 \\ 0 & 10 & 10 & 1 \end{bmatrix}$$

получим

$$[Q] = [T][N][G][S][Tr_x][M_2]^{-1}$$

Точка на поверхности $Q(t, \phi)$ имеет координаты

$$Q(t, \phi) = [\cos t \quad \sin t \quad 0 \quad 1] \begin{bmatrix} 5\sqrt{2}/2 & 5\sqrt{2}/2 & 0 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times$$

$$\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & \sin \phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 20/\sqrt{2} & 0 & 0 & 1 \end{bmatrix} \times$$

$$\times \begin{bmatrix} \sqrt{2}/2 & 0 & -\sqrt{2}/2 & 0 \\ 0 & -1 & 0 & 0 \\ -\sqrt{2}/2 & 0 & -\sqrt{2}/2 & 0 \\ 0 & 10 & 10 & 1 \end{bmatrix}$$

или

$$Q(t, \phi) = \left(\frac{1}{2}\right) [\cos t \quad \sin t \quad 0 \quad 1] \times$$

$$\times \begin{bmatrix} 5(1 - \sin \phi) & -5\sqrt{2} \cos \phi & -5(1 + \sin \phi) & 0 \\ -(1 + \sin \phi) & -\sqrt{2} \cos \phi & (1 - \sin \phi) & 0 \\ 0 & 0 & 0 & 0 \\ 20 & 20 & 0 & 0 \end{bmatrix}$$

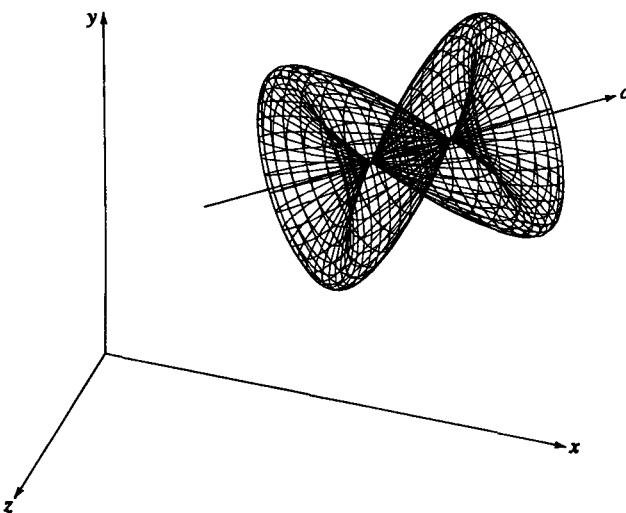


Рис. 6-12 Сложная эллиптическая поверхность вращения из примера 6-3.

Для $t = \pi/2, \phi = \pi/6$

$$\begin{aligned} Q(\pi/2, \pi/6) &= \left(\frac{1}{2}\right) [0 \ 1 \ 0 \ 1] \begin{bmatrix} 5/2 & -5\sqrt{6}/2 & -15/2 & 0 \\ -3/2 & -\sqrt{6}/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 \\ 20 & 20 & 0 & 1 \end{bmatrix} = \\ &= [37/4 \ 10 - \sqrt{6}/4 \ 1/4 \ 1] = \\ &= [9.25 \ 9.388 \ 0.25 \ 1] \end{aligned}$$

Получившаяся в результате поверхность изображена на рис. 6-12. Заметим, что эта поверхность самопересекается и имеет сложную форму.

Формальное дифференцирование уравнения (6-7) дает параметрические производные для поверхности вращения. А именно, производная в осевом направлении равна

$$Q_t(t, \phi) = [T'][N][G][S], \quad (6-11)$$

а производная в радиальном направлении

$$Q_\phi(t, \phi) = [T][N][G][S'], \quad (6-12)$$

где штрих обозначает соответствующее дифференцирование.

Нормаль к поверхности задается векторным произведением параметрических производных, т. е.

$$n = Q_t \times Q_\phi. \quad (6-13)$$

6-3 ЗАМЕТАЮЩИЕ ПОВЕРХНОСТИ

Пространственную поверхность также можно получить путем перемещения объекта, например отрезка, ломаной или кривой вдоль некоторой кривой, в пространстве. Получающиеся в результате поверхности называются заметающими. Генерация заметающих поверхностей часто используется в геометрическом моделировании. Самый простой заметающий объект — точка. Результатом заметания точки является, конечно, не поверхность, а пространственная кривая. Тем не менее на этом примере хорошо иллюстрируется основная идея.

Рассмотрим координатный вектор $P[x \ y \ z \ 1]$, перемещаемый вдоль кривой, заданной заметающим преобразованием $[T(s)]$. Координатный вектор $Q(s)$, представляющий получающуюся в результате кривую, задается выражением

$$Q(s) = P[T(s)] \quad s_1 \leq s \leq s_2 \quad (6-14)$$

Преобразование $[T(s)]$ определяет форму кривой. Например, если направляющая есть отрезок длины n , параллельный оси z , то (см. уравнение 3-14)

$$[T(s)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & ns & 1 \end{bmatrix} \quad 0 \leq s \leq 1.$$

Если направляющая есть окружность с центром в начале координат, лежащая в плоскости $z = \text{const}$, то (см. уравнение 3-8)

$$[T(s)] = \begin{bmatrix} \left(\frac{r}{x}\right) \cos\{2\pi(s + s_i)\} & 0 & 0 & 0 \\ 0 & \left(\frac{r}{y}\right) \sin\{2\pi(s + s_i)\} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 0 \leq s \leq 1,$$

где $s_i = (1/2\pi) \arctg(y_i/x_i)$ и для $P[x \ y \ z \ 1]$, $r = \sqrt{x^2 + y^2}$. Здесь подстрочный индекс i используется для обозначения начальной, или стартовой точки.

Сложные направляющие можно сконструировать с помощью объединения более простых. Например, после объединения двух предыдущих преобразований получим один оборот спирали вдоль оси z , т. е.

$$[T(s)] = \begin{bmatrix} \left(\frac{r}{x}\right) \cos\{2\pi(s + s_i)\} & 0 & 0 & 0 \\ 0 & \left(\frac{r}{y}\right) \sin\{2\pi(s + s_i)\} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & ns & 1 \end{bmatrix} \quad 0 \leq s \leq 1.$$

Самая простая заметающая поверхность получается в результате перемещения отрезка вдоль направляющей. Напомним, что параметрическое уравнение отрезка

$$P(t) = P_1 + (P_2 - P_1)t \quad 0 \leq t \leq 1. \quad (6-15)$$

Соответствующая заметающая поверхность задается уравнением

$$Q(t, s) = P(t)[T(s)], \quad 0 \leq t \leq 1, \quad s_1 \leq s \leq s_2, \quad (6-16)$$

где $[T(s)]$ снова обозначает заметающее преобразование. Если оно состоит только из переносов и/или локальных или общих масштабирований, то в результате возникает плоская поверхность. Если заметающее преобразование включает повороты, то поверхность не будет плоской. На рис. 6-13 изображена спиральная заметающая поверхность, полученная из отрезка, в начальный момент параллельного оси y , с помощью одновременного переноса вдоль оси x и вращения вокруг нее. Более подробно этот метод объясняется в следующем примере.

Пример 6-4 Линейчатая заметающая поверхность

Рассмотрим отрезок, лежащий в плоскости xy , параллельный оси y и заданный концевыми точками $P_1[0 \ 0 \ 0]$ и $P_2[0 \ 3 \ 0]$. Найти точку с параметрами $t = 0.5$, $s = 0.5$ на заметающей поверхности, образованной одновременным переносом отрезка на 10 единиц вдоль оси x и его поворотом на 2π вокруг оси x .

В данном случае матрица заметающего преобразования — переноса и последующего поворота, задается матрицей¹

$$[T(s)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\pi s) & \sin(2\pi s) & 0 \\ 0 & -\sin(2\pi s) & \cos(2\pi s) & 0 \\ ls & 0 & 0 & 1 \end{bmatrix}$$

Параметрическое уравнение отрезка

$$\begin{aligned} P(t) &= P_1 + (P_2 - P_1)t = [0 \ 0 \ 0 \ 1] + [0 - 0 \ 3 - 0 \ 0 - 0 \ 1 - 1]t \\ &= [0 \ 3t \ 0 \ 1] \end{aligned}$$

Из уравнения (6-15) заметающая поверхность определяется следующим образом:

$$\begin{aligned} Q(t, s) &= [P(t)][T(s)] \\ &= [0 \ 3t \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\pi s) & \sin(2\pi s) & 0 \\ 0 & -\sin(2\pi s) & \cos(2\pi s) & 0 \\ ls & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

и

$$\begin{aligned} Q(0.5, 0.5) &= [0 \ 1.5 \ 0 \ 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 5 & 0 & 0 & 1 \end{bmatrix} \\ &= [5 \ -1.5 \ 0 \ 1]. \end{aligned}$$

Результат изображен на рис. 6-13.

Для создания заметающих поверхностей также могут быть использованы параметрические кривые, например кубические сплайны, параболические сплайны,

¹ Та же самая матрица получается для поворота и последующего переноса.

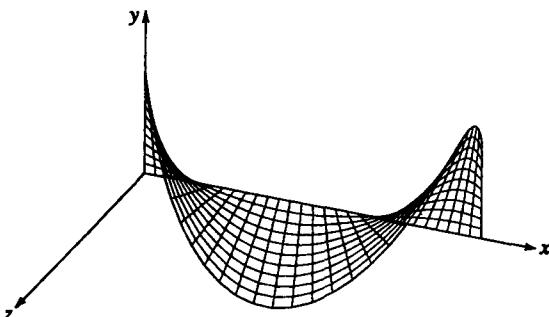


Рис. 6-13 Спиральная заметающая поверхность.

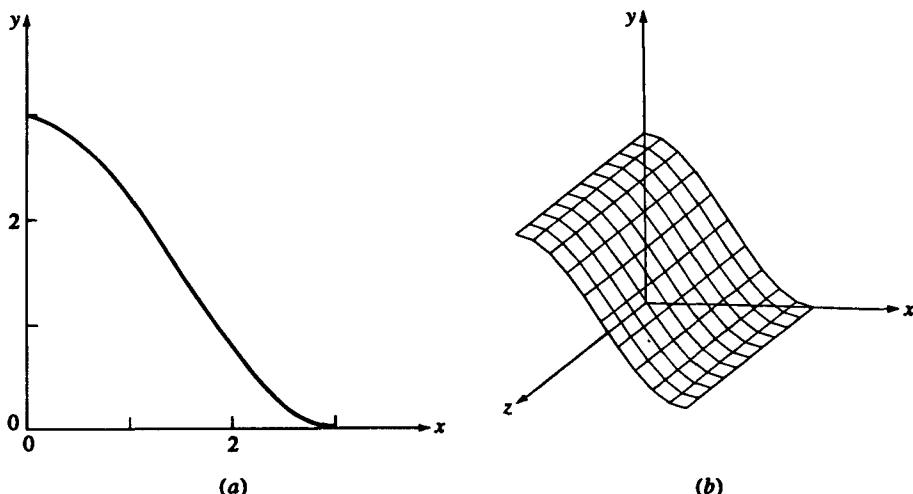


Рис. 6-14 Заметающая поверхность на основе кубического сплайна. (a) Кривая; (b) поверхность.

кривые Безье и В-сплайны. В этом случае уравнение поверхности идентично уравнению (6-15), где $P(t)$ теперь представляет параметрическую кривую. На рис. 6-14 показана заметающая поверхность, созданная из одного сегмента кубического сплайна, перенесенного вдоль оси z . Этот метод иллюстрируется примером.

Пример 6-5 Заметающая поверхность с кубическим сплайном в качестве образующей

Рассмотрим поверхность, образованную в результате переноса на 10 единиц вдоль оси z кубического сплайна, определяемого следующими данными: $P_1[0 \ 3 \ 0 \ 1]$, $P_2[3 \ 0 \ 0 \ 1]$, $P'_1[3 \ 0 \ 0 \ 0]$, $P'_2[3 \ 0 \ 0 \ 0]$.¹

¹ В однородных координатах касательный вектор имеет нулевой однородный координатный множитель.

Заметающая поверхность задается формулами:

$$Q(t, s) = [C(t)][T(s)] \quad 0 \leq t \leq 1, \quad 0 \leq s \leq 1.$$

Нормализованный сегмент кубического сплайна задается (см. уравнение 5-27)

$$[C(t)] = [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 3 & 0 & 1 \\ 3 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 \end{bmatrix}$$

Сегмент кривой изображен на рис. 6-14a.

Заметающее преобразование (см. уравнение 3-14) имеет вид:

$$[T(s)] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & ns & 1 \end{bmatrix}$$

Следовательно,

$$Q(t, s) = [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} 0 & 6 & 0 & 0 \\ 6 & -9 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 10s & 1 \end{bmatrix}$$

Для $t = 0.5, s = 0.5$

$$Q(0.5, 0.5) = [0.125 \quad 0.25 \quad 0.5 \quad 1] \begin{bmatrix} 0 & 6 & 0 & 0 \\ 6 & -9 & 0 & 0 \\ 3 & 0 & 0 & 0 \\ 0 & 3 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 5 & 1 \end{bmatrix}$$

$$= [3 \quad 1.5 \quad 5 \quad 1].$$

Результат изображен на рис. 6-14b.

Следует позаботиться о том, чтобы избежать вырождения поверхностей или их частей при генерации заметающих поверхностей из отрезков и кривых. Пример этого приведен на рис. 6-15. Здесь s -образную кривую, расположенную в плоскости xy , перемещают параллельно оси x . Заметим, что «хвосты» с левой и правой сторон являются вырожденными частями поверхности (т. е. отрезками) с нулевой площадью. Такие вырожденные части поверхности могут породить проблемы в геометрических моделирующих системах.

Кроме незамкнутых кривых для создания заметающих поверхностей используются замкнутые ломаные и кривые. При добавлении концевых поверхностей заметающая поверхность ограничивает конечный объем в пространстве. Подобным способом объемные примитивы создаются во многих геометрических моделирующих системах. Перемещаемый вдоль прямой направляющей квадрат или прямоугольник порождает прямоугольный параллелепипед. Окружность, перемещаемая вдоль прямой направляющей, порождает цилиндр. Окружность с уменьшающимся радиусом, перемещаемая вдоль прямой направляющей, порождает конус. Также возможно вращение вокруг направляющей оси. На рис. 6-16 показана

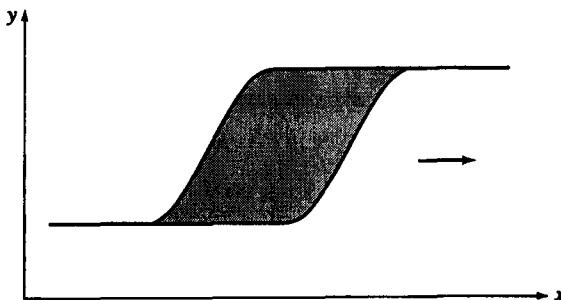


Рис. 6-15 Заметающая Поверхность с вырожденными участками.

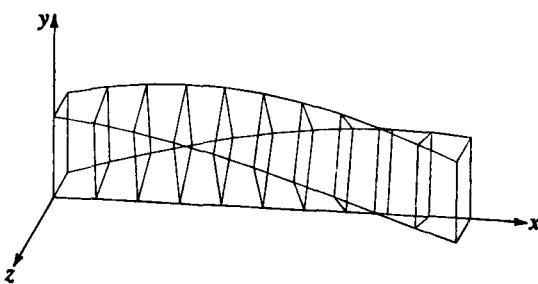


Рис. 6-16 Заметающая Поверхность, образованная квадратом, Перемещаемым вдоль оси x и одновременно вращаемым вокруг нее.

заметающая поверхность, образованная из плоского квадрата, перпендикулярного оси x и перемещаемого вдоль оси x с одновременным вращением на 90° вокруг этой же оси.

При перемещении плоского многоугольника или замкнутой кривой вдоль произвольной направляющей кривой следует отметить два важных момента. Во-первых, какая точка многоугольника постоянно лежит на направляющей? В общем случае это может быть любая точка многоугольника или замкнутой кривой. Для различных точек порожденные поверхности различаются.

Во-вторых, каково направление нормали многоугольника или замкнутой кривой во время перемещения вдоль направляющей? В этом случае обычно применяется два подхода. Нормаль может быть направлена как касательная к направляющей кривой. Либо может задаваться независимо от направляющей. Последняя альтернатива очень гибка. Два примера изображены на рис. 6-17, где показаны заметающие поверхности, образованные квадратом, центрированным на оси x и передвигаемым вдоль направляющей $x = ls$, $y = \cos(\pi s) - 1$. На рис. 6-17a нормаль направлена как ось x . На рис. 6-17b нормаль в каждой точке направлена так же, как и касательная к направляющей. Отметим разницу между этими двумя заметающими поверхностями. Детальный пример дополнительно иллюстрирует эту идею.

Пример 6-6 Сложная заметающая поверхность

Поверхность создается перемещением плоского квадрата, заданного вершинами $P_1[0 \ -1 \ 1]$, $P_2[0 \ -1 \ -1]$, $P_3[0 \ 1 \ -1]$, $P_4[0 \ 1 \ 1]$, вдоль направляющей кривой $x = 10s$, $y = \cos(\pi s) - 1$. При этом направление нормали к многоугольнику совпадает с касательной к образующей.

Направление касательной к образующей равно $[10 \ -\pi \sin(\pi s) \ 0]$. Таким образом, угол Поворота вокруг оси z , необходимого для сополавленности нормали многоугольника и касательной, равен

$$\psi = \operatorname{arctg} \left(\frac{-\pi \sin(\pi s)}{10} \right).$$

Заметающее преобразование равно, таким образом,

$$[T(s)] = \begin{bmatrix} \cos \psi & \sin \psi & 0 & 0 \\ -\sin \psi & \cos \psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 10s & \cos(\pi s) - 1 & 0 & 1 \end{bmatrix}.$$

При $s = 0.5$ угол поворота равен

$$\psi = \operatorname{arctg} \left(\frac{-\pi \sin(\pi/2)}{10} \right) = \operatorname{arctg}(-\pi/10) = -17.44^\circ$$

Тогда при $s = 0.5$ квадрат задается матрицей

$$Q = \begin{bmatrix} 0 & -1 & 1 & 1 \\ 0 & -1 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.954 & -0.3 & 0 & 0 \\ 0.3 & 0.954 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 5 & -1 & 0 & 1 \end{bmatrix} = \\ = \begin{bmatrix} 4.7 & -0.954 & 1 & 1 \\ 4.7 & -0.954 & -1 & 1 \\ 5.3 & -0.046 & -1 & 1 \\ 5.3 & -0.046 & 1 & 1 \end{bmatrix}$$

Результат изображен на рис. 6-17b

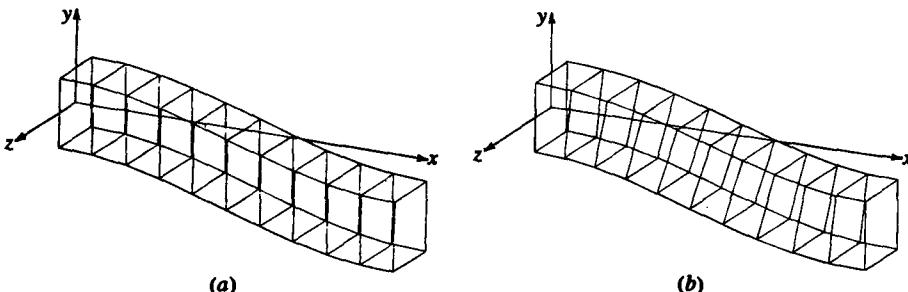


Рис. 6-17 Прямоугольник перемещается вдоль направляющей кривой. (a) Нормаль направлена как ось x ; (b) нормаль направлена так же, как касательная к направляющей кривой.

Дополнительную информацию о заметающих поверхностях можно найти в [6-9] и [6-10], а также в приведенной в них библиографии.

6-4 КВАДРАТИЧНЫЕ ПОВЕРХНОСТИ

Обычные квадратичные поверхности, такие как сфера, конус, цилиндр, эллипсоид, параболоид вращения и гиперболоид вращения являются также и поверхностями вращения (см. разд. 6-2). Эти квадратичные поверхности, особенно сфера, конус и цилиндр, играют важную роль при изготовлении деталей механизмов, а также в описании обрабатываемых поверхностей. Например, баскетбольные и футбольные мячи имеют сферическую форму, воронки — коническую, банки для пива — цилиндрическую, а спутниковые антенны — параболическую. Квадратичные поверхности также важны и при описании более сложных поверхностей. Например, фюзеляж истребителя North American P-51 времен второй мировой войны был описан с помощью сегментов конических поверхностей [6-11]. Этот метод называется методом построения плазов с помощью конических сечений.

Существует два метода представления квадратичных поверхностей — алгебраический (включающий параметрические представления) и геометрический. Геометрические описания обычно применяются для естественных квадратичных поверхностей. Алгебраически квадратичная поверхность задается в общем виде уравнением

$$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fxz + Gx + Hy + Jz + K = 0, \quad (6-17)$$

где $A, B, C, D, E, F, G, H, J, K$ — константы. Уравнение (6-17) является обобщением уравнения конических сечений (см. уравнение 4-31) на три измерения. В матричной форме обобщенное квадратичное уравнение можно записать как

$$[X][S][X]^T = 0, \quad (6-18)$$

где $[X] = [x \ y \ z \ 1]$, а

$$[S] = \begin{bmatrix} A & D/2 & F/2 & G/2 \\ D/2 & B & E/2 & H/2 \\ F/2 & E/2 & C & J/2 \\ G/2 & H/2 & J/2 & K \end{bmatrix} = \left(\frac{1}{2}\right) \begin{bmatrix} 2A & D & F & G \\ D & 2B & E & H \\ F & E & 2C & J \\ G & H & J & 2K \end{bmatrix}$$

Как и в случае конических сечений, квадратичные поверхности бывают либо центральными, либо нецентральными. Центральные квадрики — это эллипсоид и гиперболоид¹. Параболоиды — это нецентральные поверхности. Если квадрика центральна, то ее центр можно перенести в начало координат. Если это невозможно, то квадрика нецентральная, т. е. является параболоидом. Перенос центра в начало координат приводит к исчезновению линейных членов уравнения (6-17).

После переноса уравнение (6-18) превращается в

$$[X][Tr][S][Tr]^T[X]^T = [X][S'][X]^T = 0, \quad (6-19)$$

¹ Сфера есть частный случай эллипса. Цилиндр является предельным случаем для эллипса либо гиперболоида.

где матрица переноса имеет вид:

$$[Tr] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ l & m & n & 1 \end{bmatrix}.$$

Преобразованная матрица $[S']$ равна

$$[S'] = \left(\frac{1}{2}\right) \begin{bmatrix} 2A' & D' & F' & G' \\ D' & 2B' & E' & H' \\ F' & E' & 2C' & J' \\ G' & H' & J' & 2K' \end{bmatrix} \quad (6-20)$$

с $A' = A, B' = B, C' = C, D' = D, E' = E, F' = F$ и

$$\begin{aligned} G' &= 2Al + Dm + Fn + G, \\ H' &= Dl + 2Bm + En + H, \\ J' &= Fl + Em + 2Cn + J, \\ 2K' &= lG' + mH' + nJ' + (Gl + Hm + Jn + 2K). \end{aligned}$$

Исключение линейных членов, т. е. случай $C' = H' = J' = 0$, приводит к

$$\begin{aligned} 2Al + Dm + Fn + G &= 0, \\ Dl + 2Bm + En + H &= 0, \\ Fl + Em + 2Cn + J &= 0 \end{aligned}$$

или

$$[l \ m \ n] \begin{bmatrix} 2A & D & F \\ D & 2B & E \\ F & E & 2C \end{bmatrix} = \begin{bmatrix} -G \\ -H \\ -J \end{bmatrix}, \quad (6-21)$$

что можно записать в следующем виде:

$$[M][L] = [G].$$

Если матрица $[L]$ обратима, то решение существует и квадрика центральна, т. е. является эллипсоидом или гиперболоидом. Если же $[L]$ особенная (вырожденная) матрица, то решения не существует и квадрика является параболоидом. Приравнивание детерминанта $[L]$ к нулю дает условие для параболоида. Таким образом,

$$\det[L] = \begin{bmatrix} 2F & D & F \\ D & 2B & E \\ F & E & 2C \end{bmatrix} = 0 \quad (6-22)$$

$A(E^2 - BC) + B(F^2 - AC) + C(D^2 - AB) - (ABC + DEF) = 0$ является условием для параболоида. Далее, если $\det[L] > 0$, то мы получаем гиперболоид, и если $\det[L] < 0$ — эллипсоид.

Как и для конических сечений (см. разд. 4-10), независимо от того, обратима $[L]$ или нет, оси квадрики с помощью поворотов можно сделать параллельными координатным осям. Здесь требуется три поворота: два для того, чтобы сделать одну из осей квадрики параллельной координатной оси (см. разд. 3-9) и последний поворот вокруг этой же оси для того, чтобы сделать две другие оси квадрики параллельными другим координатным осям. Для квадрики в обобщенном виде решение получающихся уравнений для необходимых углов поворотов является нетривиальной задачей и здесь не приводится.

Аналогичным образом, если квадрика нецентральна, т. е. является параболоидом, то как было ранее упомянуто, нельзя исключить из уравнения все линейные члены. Тем не менее, два из трех линейных членов могут быть исключены. И снова, решение получающихся при этом уравнений является нетривиальной задачей и здесь не приводится.

Результатом вышеперечисленных операций должно было быть приведение квадрики к стандартному виду. Для центральной квадрики в стандартном виде центр находится в начале координат, а оси направлены в соответствии с координатными осями. Запишем получившееся стандартное уравнение в матричном виде

$$[X] \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & -\kappa \end{bmatrix} [X]^T = 0 \quad (6-23)$$

или

$$\alpha x^2 + \beta y^2 + \gamma z^2 = \kappa. \quad (6-24)$$

Если $\alpha \neq \beta \neq \gamma > 0$, $\kappa > 0$, то мы получим обобщенный эллипсоид (см. рис. 6-18a). Наибольшее из значений α , β , γ определяет главную ось. Если две из трех констант равны, например, $\alpha = \beta \neq \gamma > 0$, то мы получим эллипсоид вращения. Ось вращения связана с константой, отличной от двух остальных. Если $\alpha = \beta = \gamma > 0$, то мы получим сферу радиуса $\sqrt{\kappa/\alpha}$.

Переписав уравнение (6-24) в виде

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$$

и устремив одну из констант к бесконечности, получим цилиндр как предельный случай эллипсоида. «Ось» этого эллипса, заданная отсутствующей координатой, «бесконечно» длинна. Если две оставшиеся константы равны, например $\alpha = \beta$, то мы получим круговой цилиндр. Если они не равны, например $\alpha \neq \beta$, то получим эллиптический цилиндр. В этом случае уравнение (6-24) превращается в

$$\alpha x^2 + \beta y^2 = \kappa.$$

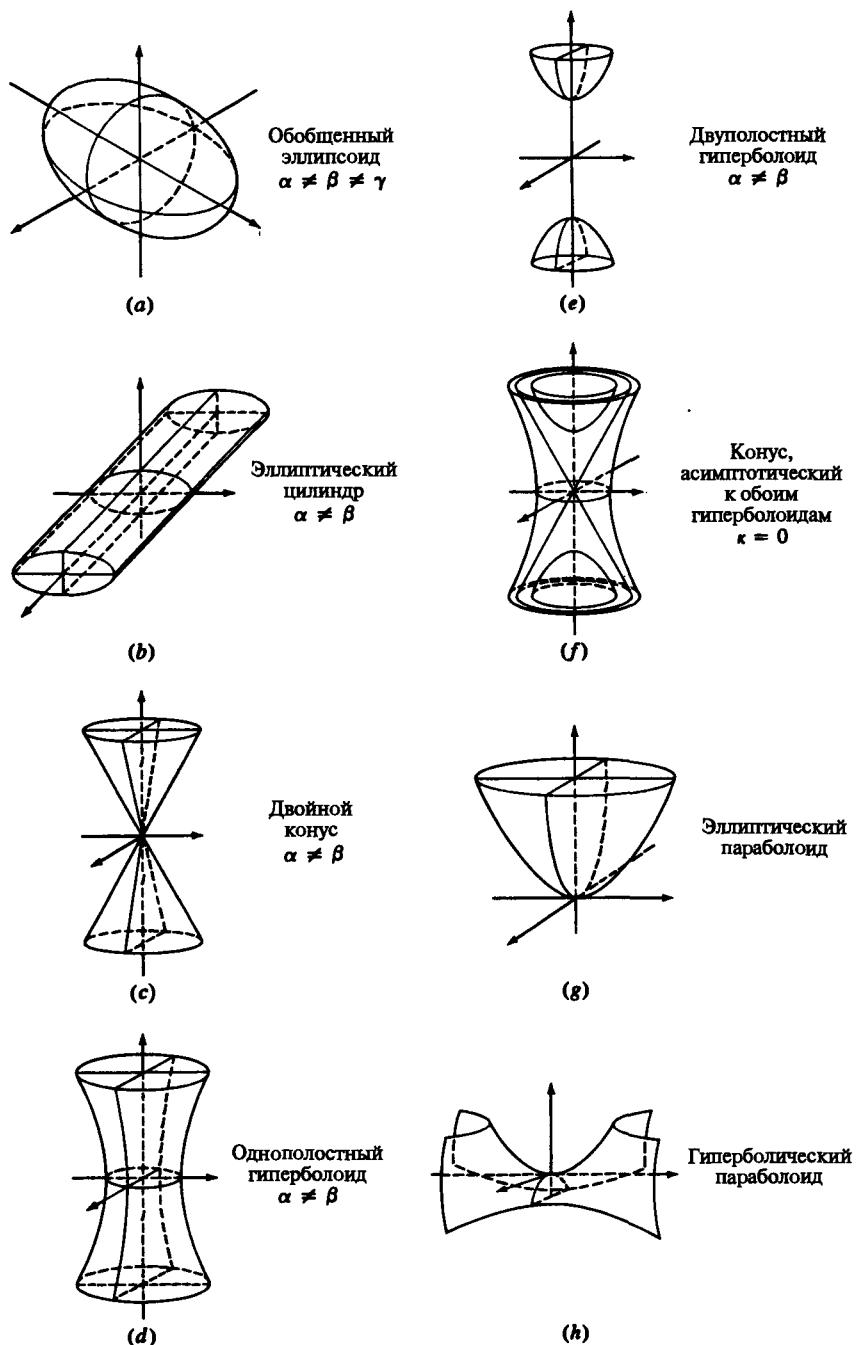


Рис. 6-18 Квадратичные поверхности. (а) Обобщенный эллипсоид $\alpha \neq \beta \neq \gamma$; (б) эллиптический цилиндр $\alpha \neq \beta$; (в) двойной конус $\alpha \neq \beta$; (г) однополостный гиперболоид $\alpha \neq \beta$; (е) двуполостный гиперболоид $\alpha \neq \beta$; (ж) конус, асимптотический к обоим гиперболоидам $\kappa = 0$; (з) эллиптический параболоид; (и) гиперболический параболоид.

Посмотрим пример, приведенный на рис. 6-18b. Если $\kappa = 0$, то мы получим мнимый цилиндр, потому что сумма квадратов двух вещественных чисел не может быть равной нулю¹.

Если $\kappa = 0$, одна из констант отрицательна, а две другие положительны, например $\alpha, \beta > 0, \gamma < 0$, то мы получим двойной конус. Ось конуса связана с координатой отрицательной константы. Если $\alpha = \beta$, то получится правильный круговой конус. Если $\alpha \neq \beta$, получится эллиптический конус. В этом случае уравнение (6-24) превращается в

$$\alpha x^2 + \beta y^2 = \gamma z^2.$$

Пример изображен на рис. 6-18c.

Если $\kappa = 0, \alpha, \beta, \gamma \neq 0$, то конус мнимый, потому что сумма квадратов трех вещественных чисел не может быть равной нулю.

Если $\alpha, \beta, \gamma > 0$, а $\kappa < 0$, получится мнимый эллипсоид, потому что опять сумма квадратов трех вещественных чисел не может быть отрицательной.

Если одна из констант в уравнении (6-24) отрицательна, например $\gamma < 0$, а другие положительны, в результате квадратичная поверхность будет однополостным гиперболоидом. Ось гиперболоида связана с отрицательной константой, например, если $\gamma < 0$, то осью гиперболоида будет координатная ось z . На рис. 6-18d приведен пример. Если две положительные константы равны между собой, например $\alpha = \beta$, то получается гиперболоид вращения. В противном случае — эллиптический гиперболоид. Гиперболоид называется однополостным, потому что можно соединить любые две точки поверхности, не выходя за ее границы. В пределе, при приближении отрицательной константы к нулю, получится либо круговой, либо эллиптический цилиндр. Однополостный гиперболоид является двулинейчатой поверхностью (см. разд. 6-8). Двуполостный гиперболоид в некотором смысле противоположен однополостному. Здесь отрицательны две из трех констант α, β, γ , например $\alpha, \beta < 0$, а третья положительна. Ось гиперболоида связана с положительной константой. Пример приведен на рис. 6-18e. Соединить две любые точки двуполостного гиперболоида, не покидая поверхность, невозможно. Для $\alpha, \beta < 0, \gamma > 0$ вершины двух полостей гиперболоида находятся в диапазоне

$$\pm \left(\frac{\alpha \beta \kappa}{\gamma^2} \right)^{1/6}$$

на оси z^2 . Если отрицательные константы равны, то получается гиперболоид вращения. В противном случае получается эллиптический гиперболоид. Заметим, что число отрицательных констант равно числу полостей гиперболоида.

Если $\kappa = 0$, то оба гиперболоида вырождаются в конус, асимптотический к обоим из них, как это показано на рис. 6-18f.

¹ Кроме случая $x = y = 0$, т. е. прямой, совпадающей с осью z . — Прим. перев.

² $\pm \left(\frac{\alpha \beta \kappa}{\gamma^2} \right)^{1/6} = \pm c$, когда уравнение двуполостного гиперболоида записано в менее общем виде $\frac{z^2}{c^2} - \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$.

Теперь рассмотрим нецентральные квадратичные поверхности, а именно параболоиды. В стандартном виде, где z является осью параболоида, имеем уравнение

$$[X] \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma & 0 \end{bmatrix} [X]^T = 0 \quad (6-25)$$

или

$$\alpha x^2 + \beta y^2 + \gamma z = 0. \quad (6-26)$$

Если $\alpha, \beta > 0$, то, как показано на рис. 6-18g, получается эллиптический параболоид. Если $\alpha = \beta > 0$, то получим параболоид вращения. Вершина параболоида находится в начале координат. При $\gamma < 0$ поверхность «открыта» в положительную сторону оси z . Аналогичным образом, при $\gamma > 0$ поверхность открыта в отрицательную сторону оси z . Если $\gamma = 0$, поверхность превращается в эллипс¹. Если либо α , либо $\beta < 0$, то, как показано на рис. 6-18h, получается гиперболический параболоид. Если $\alpha > 0, \beta < 0$, то фокальной осью является ось y . Аналогично, если $\alpha < 0, \beta > 0$, фокальная — ось x . Если $\gamma < 0$, то седло направлено вверх, как это показано на рис. 6-18h. Если $\gamma > 0$, то поверхность отражена относительно плоскости $z = 0$ и седло направлено вниз. Если $\gamma = 0$, то поверхность вырождается в гиперболу². Гиперболический параболоид иногда называют седловой поверхностью. Заметим, что даже при $\alpha = \beta$ гиперболический параболоид не является поверхностью вращения. Тем не менее это двулинейчатая поверхность (см. разд. 6-8).

И наконец, если либо α , либо $\beta = 0$, то получаем параболический цилиндр. Дальнейшую информацию о квадратических поверхностях и их свойствах можно найти в [6-12] — [6-14].

Хотя для нахождения свойств квадратичных поверхностей часто бывает полезна явная форма их задания, но, аналогично случаю конических сечений, параметрическое задание поверхностей позволяет получить более привлекательные и доходчивые графические изображения. Для квадратичных поверхностей со стандартной ориентацией параметрические представления задаются следующим образом:

Эллипсоид:

$$\begin{aligned} x &= a \cos \theta \sin \phi, & 0 \leq \theta \leq 2\pi, \\ y &= b \sin \theta \sin \phi, & 0 \leq \phi \leq 2\pi, \\ z &= c \cos \phi. \end{aligned} \quad (6-27)$$

¹ Так как знаки α и β одинаковые, то это прямая $x = y = 0$, т. е. совпадающая с осью z . — *Прим. перев.*

² Так как у α, β разные знаки, то получается пара скрещивающихся плоскостей. — *Прим. перев.*

Однополостный гиперболоид:

$$\begin{aligned}x &= a \cos \theta \operatorname{ch} \phi, & 0 \leq \theta \leq 2\pi, \\y &= b \sin \theta \operatorname{sh} \phi, & -\pi \leq \phi \leq \pi, \\z &= c \operatorname{sh} \phi.\end{aligned}\tag{6-28}$$

Двуполостный гиперболоид:

$$\begin{aligned}x &= \pm a \operatorname{ch} \phi, & 0 \leq \theta \leq 2\pi, \\y &= b \sin \theta \operatorname{sh} \phi, & -\pi \leq \phi \leq \pi, \\z &= c \cos \theta \operatorname{sh} \phi.\end{aligned}\tag{6-29}$$

Эллиптический параболоид:

$$\begin{aligned}x &= a\phi \cos \theta, & 0 \leq \theta \leq 2\pi, \\y &= b\phi \sin \theta, & 0 \leq \phi \leq \phi_{\max}, \\z &= \phi^2.\end{aligned}\tag{6-30}$$

Гиперболический параболоид:

$$\begin{aligned}x &= a\phi \operatorname{ch} \theta, & -\pi \leq \theta \leq \pi, \\y &= b\phi \operatorname{sh} \theta, & \phi_{\min} \leq \phi \leq \phi_{\max}, \\z &= \phi^2.\end{aligned}\tag{6-31}$$

Эллиптический конус:

$$\begin{aligned}x &= a\phi \cos \theta, & 0 \leq \theta \leq 2\pi, \\y &= b\phi \sin \theta, & \phi_{\min} \leq \phi \leq \phi_{\max}, \\z &= c\phi.\end{aligned}\tag{6-32}$$

Эллиптический цилиндр:

$$\begin{aligned}x &= a \cos \theta, & 0 \leq \theta \leq 2\pi, \\y &= b \sin \theta, & \phi_{\min} \leq \phi \leq \phi_{\max}, \\z &= \phi.\end{aligned}\tag{6-33}$$

Параболический цилиндр:

$$\begin{aligned}x &= a\theta^2, & 0 \leq \theta \leq \theta_{\max}, \\y &= 2a\theta, & \phi_{\min} \leq \phi \leq \phi_{\max}, \\z &= \phi.\end{aligned}\tag{6-34}$$

Параметрически заданные эллипсоид и часть гиперболического параболоида показаны на рис. 6-19.

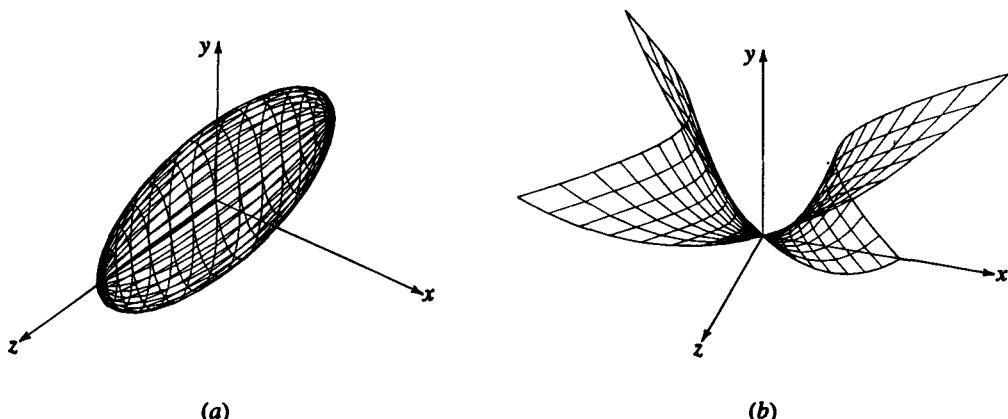


Рис. 6-19 Параметрические квадратичные поверхности. (а) Эллипсоид; (б) гиперболический параболоид.

Хотя представлять все квадратичные поверхности удобно с помощью обобщенного уравнения второй степени (см. уравнение (6-17)), а генерировать — с помощью параметрического задания, описанного выше, но при компьютерном моделировании квадратичные поверхности наиболее точно представляются геометрически (см. [6-15] и [6-17]). Геометрически любая квадратичная поверхность представляется точкой, двумя ортогональными (единичными) векторами и тремя скалярами. Точка — либо центр, либо вершина, фиксирует местоположение в пространстве. Вектора и их векторное произведение определяют оси поверхности или ориентацию. Скаляры определяют ее измерения или размер. Например, сфера определяется своим центром и радиусом, правильный цилиндр — вектором, задающим его ось, точкой на оси и радиусом, эллипсоид определяется центром, двумя векторами, представляющими две из его трех ортогональных осей и тремя скалярами, представляющими длины вдоль этих осей, и т. д. В табл. 6-1 перечислены геометрические описания квадратичных поверхностей.

При работе с геометрически заданными квадратичными поверхностями необходимо преобразовывать только определяющие точки и векторы. Скалярные величины остаются неизменными и, таким образом, не накапливают при вычислениях ошибок, возникающих из-за использования в компьютере арифметики с ограниченной разрядной сеткой.

Например, геометрически заданный радиус сферы всегда равен R . А радиус сферы, полученный путем преобразования алгебраического представления (см. уравнение (6-17)) равен $R \pm \epsilon$, где ϵ — небольшая числовая ошибка. Числовая стабильность определяющих скаляров очень важна, например при определении характера кривой пересечения двух квадратичных поверхностей или при определении идентичности двух поверхностей (см. [6-15]).

Таблица 6-1 Геометрические описания квадратичных поверхностей

Поверхность	Скаляры	Точка	Векторы
Плоскость	Нет	Любая точка на плоскости	Единичная нормаль
Сфера	Радиус	Центр	Нет
Правильный круговой конус	Половинный угол	Центр (вершина)	Единичный вектор, параллельный оси
Правильный круговой цилиндр	Радиус	Любая точка на оси	Единичный вектор, параллельный оси
Правильный эллиптический цилиндр	Длины осей эллипса (2)	Любая точка на оси	—
Правильный параболический цилиндр	Фокусное расстояние	Вершина	Единичный вектор оси
Эллиптический конус	Длины осей эллипса (2)	Центр	Единичный вектор, параллельный оси
Эллипсоид	Длины осей (3)	Центр	Два единичных вектора, параллельных двум из трех осей
Эллиптический параболоид	Длины осей эллипса	Вершина	Единичный вектор, параллельный оси
Гиперболический параболоид	Гиперболический и параболический фокусы (3)	Вершина	Единичный вектор, параллельный оси
Однополостный гиперболоид	Длины осей эллипса (2) Гиперболический фокус вдоль главной оси	Центр	Единичный вектор, параллельный оси
Двуполостный гиперболоид	Длины осей (2) Местоположение вершины на оси симметрии (1)	Центр	Единичный вектор, параллельный оси

6-5 КУСОЧНОЕ ПРЕДСТАВЛЕНИЕ ПОВЕРХНОСТЕЙ

В предыдущих разделах обсуждались создание и характеристики поверхностей, для которых существуют известные аналитические описания. Однако есть много поверхностей, для которых таких описаний сделать невозможно. Типичными примерами являются автомобильные кузова, фюзеляжи и крылья самолетов, корпуса кораблей, скульптуры, бутылки, башмаки и т. д. Эти поверхности представляются в кусочном виде, т. е. похожем на лоскутное одеяло. В рассуждениях используется векторное параметрическое представление, потому что оно не зависит от осей, устраняет бесконечные значения углов наклона относительно некоторой произвольной системы координат, устраняет неоднозначность представления многозначных поверхностей и пространственных функций, облегчает представле-

ние поверхностей в однородных координатах, и такое представление совместимо с использованием преобразований трехмерных однородных координат, обсуждавшихся в гл. 3.

Остаток этой главы будет посвящен обсуждению способов и методов математического описания кусков поверхностей.

Наша цель заключается в том, чтобы для создания полной поверхности объединить вместе отдельные куски вдоль их границ. Для кусочного представления кривых мы будем придерживаться принятого в гл. 5 изложения. Мы начнем наше обсуждение с достаточно интересного элемента или куска аналитической поверхности, а именно сферы.

На примере сферической поверхности мы продемонстрируем некоторые свойства кривых, полезные при описании поверхности. Определенные кривые на поверхности задаются плоскостями, пересекающими сферу. В качестве примера рассмотрим пересечение единичной сферы и плоскости, определяемое уравнением поверхности $z = \cos \phi_1 = a_1 = \text{constant}$ и показанное на рис. 6-20a. Получившаяся кривая является параллелью широты. Уравнение этой кривой получается путем решения системы двух уравнений поверхностей. Непараметрическое уравнение единичной сферы записывается в виде:

$$x^2 + y^2 + z^2 = 1.0.$$

Таким образом,

$$x^2 + y^2 = 1 - a_1^2$$

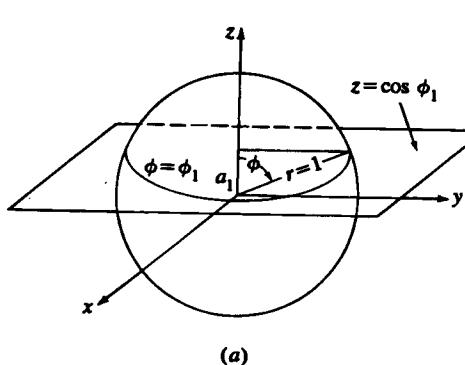
определяет пересечение плоскости и сферы.

На рис. 6-20b плоскость $\theta = \theta_0 = \text{constant}$ задается уравнением

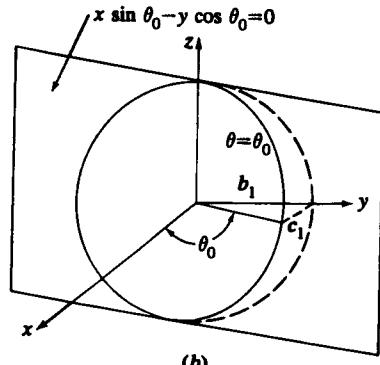
$$x \sin \theta_0 - y \cos \theta_0 = 0$$

или

$$c_1 x - b_1 y = 0.$$



(a)



(b)

Рис. 6-20 Пересечение плоскости и сферы. (a) Параллель широты; (b) меридиан долготы.

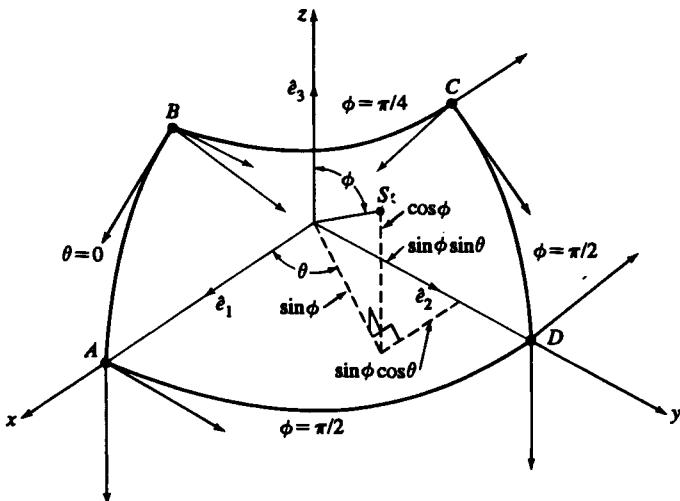


Рис. 6-21 Кусок сферической поверхности.

В результате пересечения этой плоскости и сферы получается меридиан долготы. Решение системы уравнений дает уравнение кривой пересечения; т. е.

$$y^2 \left[\left(\frac{b_1}{c_1} \right)^2 + 1 \right] + z^2 = 1.0.$$

Как показано на рис. 6-21, границы куска сферической поверхности на единичной сфере могут быть сформированы четырьмя плоскостями, двумя параллелями и двумя меридианами, пересекающими сферу. Векторное параметрическое уравнение для куска поверхности $Q(\theta, \phi)$

$$Q(\theta, \phi) = [\cos \theta \sin \phi, \quad \sin \theta \sin \phi \quad \cos \phi] \quad \theta_1 \leq \theta \leq \theta_2, \quad \phi_1 \leq \phi \leq \phi_2. \quad (6-35)$$

Кусок поверхности — это все возможные положения точки в трехмерном пространстве, движущейся с двумя степенями свободы, управляемой двумя параметрическими переменными θ и ϕ , т. е. это бипараметрическая функция.

Параметрическое представление единичной сферической поверхности (см. уравнение (6-27) с $a = b = c = 1$) таково:

$$\begin{aligned} x &= \cos \theta \sin \phi, \\ y &= \sin \theta \sin \phi, \\ z &= \cos \phi. \end{aligned}$$

Показанный на рис. 6-21 кусок поверхности задан для диапазонов изменения параметров $0 \leq \theta \leq \pi/2$ и $\pi/4 \leq \phi \leq \pi/2$. Границы или ребра его задаются кривыми AB, BC, CD и DA . Для сферического куска на рис. 6-21 эти кривые являются дугами окружностей. Каждая кривая может быть задана двумя конечными точками

и касательными векторами в концах. Следовательно, четыре граничных кривых куска задаются четырьмя координатными векторами в углах и восемью касательными векторами, по два в каждом углу. Для сферического куска на рис. 6-21 касательные векторы задаются параметрическими производными $Q(\theta, \phi)$, т. е.

$$Q_\theta(\theta, \phi) = \frac{\partial Q}{\partial \theta}(\theta, \phi) = [-\sin \theta \sin \phi \quad \cos \theta \sin \phi \quad 0] \quad (6-36)$$

и

$$Q_\phi(\theta, \phi) = [\cos \theta \cos \phi \quad \sin \theta \cos \phi \quad -\sin \phi]. \quad (6-37)$$

Касательные векторы в каждом углу показаны на рис. 6-21.

Форма внутренней части поверхности около каждого угла управляетяется вектором кручения или смешанной производной в этом углу. Для куска сферической поверхности на рис. 6-21 смешанная производная или вектор кручения

$$Q_{\theta,\phi}(\theta, \phi) = \frac{\partial^2 Q}{\partial \theta \partial \phi} = \frac{\partial^2 Q}{\partial \phi \partial \theta} = [-\sin \theta \cos \phi \quad \cos \theta \cos \phi \quad 0]. \quad (6-38)$$

Подстановка значений параметров в углах куска поверхности дает векторы кручения в них. Внутренняя форма куска задается уравнением (6-35). Это уравнение можно считать сферической смешивающей функцией. Следовательно, четырехугольный кусок поверхности может быть полностью описан 4 координатными векторами в углах, 8 касательными векторами (по два в каждом углу), 4 векторами кручения в углах и смешивающей функцией, заданной в уравнении (6-35).

Нормаль в любой точке куска поверхности определяется векторным произведением производных по параметрам. Например, для сферической поверхности

$$\begin{aligned} Q_\theta \times Q_\phi &= \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -\sin \theta \sin \phi & \cos \theta \sin \phi & 0 \\ \cos \theta \cos \phi & \sin \theta \cos \phi & -\sin \phi \end{vmatrix} \\ &= [-\cos \theta \sin^2 \phi \quad \sin \theta \sin^2 \phi \quad -\sin \phi \cos \phi]. \end{aligned} \quad (6-39)$$

На куске поверхности изопараметрические линии, т. е. линии с постоянным значением параметра, ортогональны. Следовательно, скалярное произведение производных по параметрам равно нулю. Например, для сферической поверхности

$$\begin{aligned} Q_\theta \cdot Q_\phi &= [-\sin \theta \sin \phi \quad \cos \theta \sin \phi \quad 0] \cdot [\cos \theta \cos \phi \quad \sin \theta \cos \phi \quad -\sin \phi] = 0 \\ &= [-\sin \theta \sin \phi \quad \cos \theta \sin \phi \quad 0] [\cos \theta \cos \phi \quad \sin \theta \cos \phi \quad -\sin \phi]^T = 0 \\ &= [-\cos \theta \sin \theta \cos \phi \sin \phi \quad \cos \theta \sin \theta \cos \phi \sin \phi \quad 0] = 0. \end{aligned} \quad (6-40)$$

6-6 ОТОБРАЖЕНИЕ ПАРАМЕТРИЧЕСКИХ ПОВЕРХНОСТЕЙ

Методы параметрического описания поверхностей наиболее удобно представляются в терминах отображения двупараметрической плоской поверхности из параметрического пространства uv в трехмерное объектное пространство xyz . Ограничимся здесь обсуждением отображения прямоугольной плоской поверхности

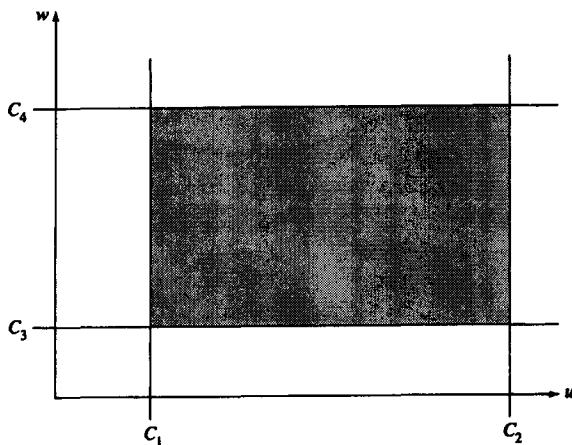


Рис. 6-22 Прямоугольная параметрическая плоская поверхность.

в параметрическом пространстве, показанной на рис. 6-22 и заданной уравнениями

$$\begin{aligned} u &= C_1 \quad C_3 \leq w \leq C_4, \\ u &= C_2 \quad C_3 \leq w \leq C_4, \\ w &= C_3 \quad C_1 \leq u \leq C_2, \\ w &= C_4 \quad C_1 \leq u \leq C_2. \end{aligned}$$

Поверхность в объектном пространстве представляется функциями, отображающими эту параметрическую поверхность в объектное пространство xyz , т. е.

$$\begin{aligned} x &= x(u, w), \\ y &= y(u, w), \\ z &= z(u, w). \end{aligned}$$

Простой двумерный пример послужит в качестве иллюстрации этого метода.

Пример 6-7 Двумерное отображение поверхности

Отобразить поверхность, описываемую в параметрическом пространстве уравнениями

$$\begin{aligned} x &= 3u + w \quad 0 \leq u \leq 1, \\ y &= 2u + 3w + uw \quad 0 \leq w \leq 1, \\ z &= 0 \end{aligned}$$

в объектное пространство. Для начала заметим, что так как $z = \text{constant} = 0$, то поверхность в объектном пространстве также является двумерной и лежит на плоскости $z = 0$.

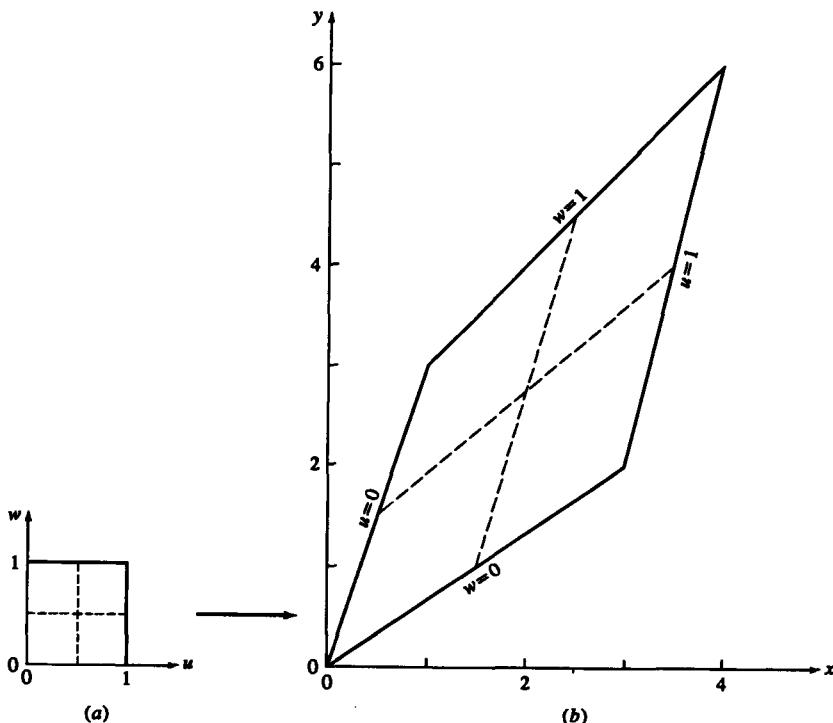


Рис. 6-23 Двумерное отображение поверхности. (а) Параметрическое пространство; (б) объектное пространство.

Границы поверхности в объектном пространстве определяются с помощью отображения в объектное пространство границ прямоугольника в параметрическом пространстве. Таким образом, для

$$\begin{aligned} u = 0; \quad & x = w, \quad y = 3w \quad \text{и} \quad y = 3x, \\ u = 1; \quad & x = w + 3, \quad y = 2(2w + 1) \quad \text{и} \quad y = 2(2x - 5), \\ w = 0; \quad & x = 3u, \quad y = 2u \quad \text{и} \quad y = (2/3)x, \\ w = 1; \quad & x = 3u + 1, \quad y = 3u + 3 \quad \text{и} \quad y = x + 2. \end{aligned}$$

Во всех случаях для получения уравнения вида $y = y(x)$ из соответствующего выражения исключался параметр (u или w). Результаты представлены на рис. 6-23.

Как показано в примере, задание постоянного значения одному из параметров порождает кривую на поверхности в объектном пространстве. Такая кривая называется изопараметрической или параметрической линией. Если задать один из параметров как функцию другого в параметрическом пространстве, т. е. $u = u(w)$, то в результате также получится кривая на поверхности в объектном простран-

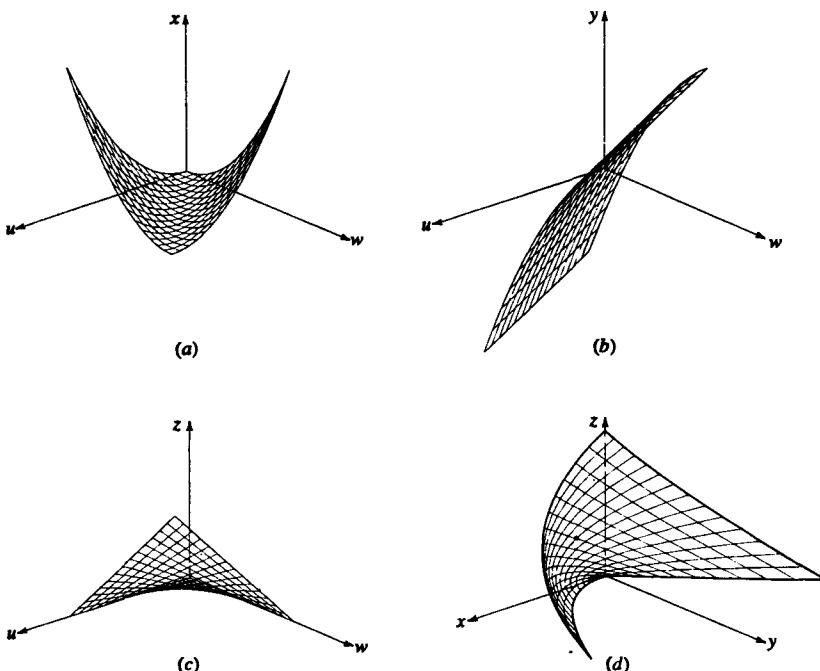


Рис. 6-24 Трехмерное отображение поверхности. (a) x компонента; (b) y компонента; (c) z компонента; (d) результат.

стве. Например, функции

$$\begin{aligned} u &= w, \quad 0 \leq w \leq 1, \\ u &= 1 - w, \end{aligned}$$

представляют диагонали единичного квадрата в параметрическом пространстве.

Спецификация значений обоих параметров задает точку на поверхности в объектном пространстве. Другим способом задания точки (или точек) может служить пересечение двух кривых в параметрическом пространстве, например $f(u, w) = 0$ и $g(u, w) = 0$. Пересечение в параметрическом пространстве отображается или преобразуется в пересечение в объектном пространстве.

В более сложном трехмерном примере дополнительно иллюстрируется описываемая идея отображения.

Пример 6-8 Трехмерное отображение поверхности

Отобразить описанную в параметрическом пространстве поверхность

$$x(u, w) = (u - w)^2, \quad 0 \leq u \leq 1,$$

$$y(u, w) = u - w^2, \quad 0 \leq w \leq 1,$$

$$z(u, w) = uw$$

в объектное пространство. Вычислить координаты в объектном пространстве точки на поверхности с параметрами $u = w = 0.5$.

Сначала найдем граничные кривые

$$\begin{aligned} u = 0; \quad &x = w^2, y = -w^2, z = 0 \quad \text{и} \quad x = -y, z = 0, \\ u = 1; \quad &x = (1-w)^2, y = 1-w^2, z = w \quad \text{и} \quad x = (1-z)^2, y = 1-z^2, \\ w = 0; \quad &x = u^2, y = u, z = 0 \quad \text{и} \quad x = y^2, z = 0, \\ w = 1; \quad &x = (u-1)^2, y = u-1, z = u \quad \text{и} \quad x = y^2, z = 1+y. \end{aligned}$$

Границные кривые изображены на рис. 6-24d более толстыми линиями. Записав параметрическую поверхность в виде векторной функции

$$Q(u, w) = [x(u, w) \quad y(u, w) \quad z(u, w)] = [(u-w)^2 \quad u-w^2 \quad uw]$$

получим

$$Q(0.5, 0.5) = [0 \quad 0.25 \quad 0.25]$$

в качестве координат точки $u = w = 0.5$, отмеченной на рис. 6-24d жирной точкой. Заметим, что каждая из компонент поверхности в объектном пространстве является также функцией параметров u, w . Каждая из этих отдельных компонент показана на рис. 6-24a, b, c. Поверхность, изложенная на рис. 6-24d, является композицией всех преобразованных компонент.

И наконец, представляют интерес отображения вырожденных кусков, соответствующих точке и прямой. Для точки отображение записывается

$$x = \text{constant}, \quad y = \text{constant}, \quad z = \text{constant}.$$

Для прямой отображение записывается в виде $x = u, y = \text{constant}, z = \text{constant}$.

6-7 БИЛИНЕЙНАЯ ПОВЕРХНОСТЬ

Одной из самых простых является билинейная поверхность. Билинейная поверхность конструируется из четырех угловых точек единичного квадрата в параметрическом пространстве, т. е. из точек $P(0, 0)$, $P(0, 1)$, $P(1, 1)$ и $P(1, 0)$. Любая точка на поверхности определяется линейной интерполяцией между противоположными границами единичного квадрата, как это показано на рис. 6-25. Любая точка внутри параметрического квадрата задается уравнением

$$Q(u, w) = P(0, 0)(1-u)(1-w) + P(0, 1)(1-u)w + P(1, 0)u(1-w) + P(1, 1)uw. \quad (6-41)$$

В матричном виде

$$Q(u, w) = [1-u \quad u] \begin{bmatrix} P(0, 0) & P(0, 1) \\ P(1, 0) & P(1, 1) \end{bmatrix} \begin{bmatrix} 1-w \\ w \end{bmatrix}. \quad (6-42)$$

Необходимо, чтобы интерполируемая поверхность удовлетворяла исходным данным. В этом случае легко проверить, что угловые точки принадлежат этой поверхности, т. е. $Q(0, 0) = P(0, 0)$ и т. д.

Уравнение (6-42) задано в обобщенном матричном представлении интерполированной поверхности, а именно — матрица функций смешения по одной из бипа-

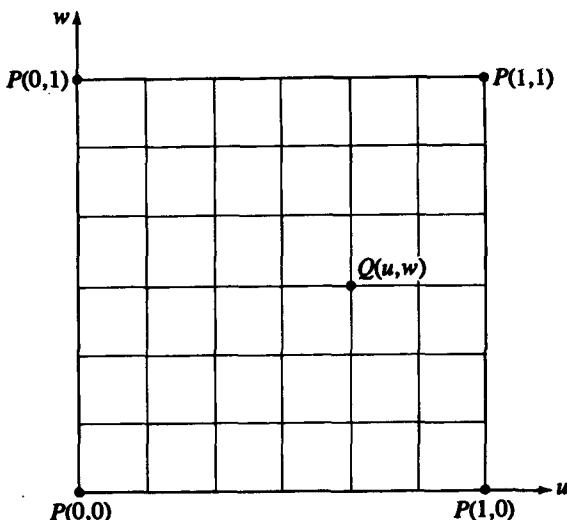


Рис. 6-25 Билинейная интерполяция в параметрическом пространстве.

раметрических переменных, геометрическая матрица, представляющая исходные данные, и матрица функций смещения по другой параметрической переменной. При изучении параметрических интерполированных поверхностей мы будем постоянно пользоваться этим представлением.

Если координатные векторы четырех точек, определяющих билинейную поверхность, заданы в трехмерном объектном пространстве, то будет трехмерна и билинейная поверхность, получаемая в результате отображения параметрического пространства в объектное. Если четыре определяющие точки не лежат в одной плоскости, то и билинейная поверхность также не лежит ни в какой плоскости. Действительно, в общем случае она сильно изогнута, пример этого показан на рис. 6-26. Определяющие точки являются концами противоположных диагоналей на противоположных гранях единичного куба. В результате получаем гиперболический параболоид. В качестве иллюстрации рассмотрим следующий пример.

Пример 6-9 Билинейная поверхность

Найти точку на билинейной поверхности, заданной точками $P(0,0) = [0 \ 0 \ 1]$, $P(0,1) = [1 \ 1 \ 1]$, $P(1,0) = [1 \ 0 \ 0]$, $P(1,1) = [0 \ 1 \ 0]$, т. е. концами противоположных диагоналей, лежащих на противоположных гранях единичного куба в объектном пространстве. Искомая точка имеет координаты $u = w = 0.5$ в параметрическом пространстве.

Напомним, что поверхность в объектном пространстве является векторной функцией:

$$Q(u, w) = [x(u, w) \ y(u, w) \ z(u, w)]$$

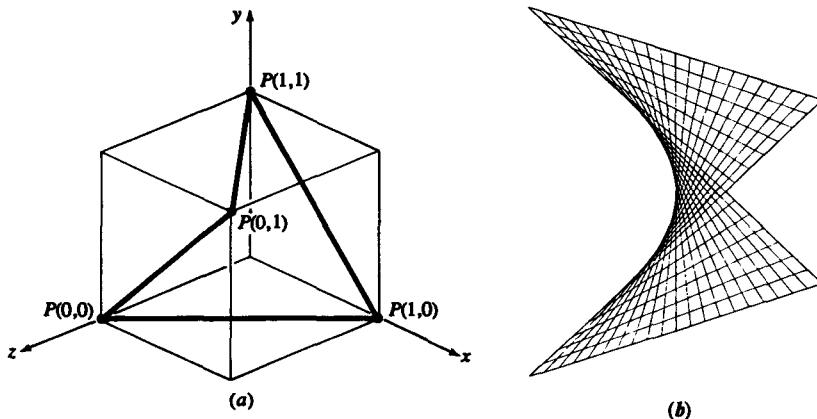


Рис. 6-26 Билинейная поверхность. (а) Определяющие угловые точки; (б) поверхность.

тогда из уравнения (6-41) имеем

$$\begin{aligned}
 Q(0.5, 0.5) &= [0 \quad 0 \quad 1](1 - 0.5)(1 - 0.5) + [1 \quad 1 \quad 1](1 - 0.5)(0.5) \\
 &\quad + [1 \quad 0 \quad 0](0.5)(1 - 0.5) + [0 \quad 1 \quad 0](0.5)(0.5) \\
 &= 0.25[0 \quad 0 \quad 1] + 0.25[1 \quad 1 \quad 1] \\
 &\quad + 0.25[1 \quad 0 \quad 0] + 0.25[0 \quad 1 \quad 0] \\
 &= [0.5 \quad 0.5 \quad 0.5]
 \end{aligned}$$

Вся поверхность изображена на рис. 6-26б.

Заметим, что каждая изопараметрическая линия на билинейной поверхности является прямой линией. В самом деле, эта поверхность является двулинейчатой (см. разд. 6-8).

6-8 ЛИНЕЙЧАТЫЕ И РАЗВЕРТЫВАЮЩИЕСЯ ПОВЕРХНОСТИ

Линейчатые поверхности часто используются в авиа- и кораблестроительной промышленности. Например, большинство авиационных крыльев являются цилиндрическими линейчатыми поверхностями. Линейчатая поверхность образуется при движении прямой линии вдоль направляющей с одной степенью свободы. Другой метод определения линейчатой поверхности состоит в следующем. Выберем произвольную точку на поверхности и будем вращать вокруг нормали плоскость, проходящую через нормаль к поверхности в этой точке (см. рис. 6-27). Если существует такая ориентация плоскости, при которой каждая точка на ребре плоскости контактирует с поверхностью, то поверхность линейчата в этом направлении. Если ребро вращающейся плоскости полностью соприкасается с поверхностью при более чем одной ориентации, то поверхность в этой точке многолинейчата.

Самой простой линейчатой поверхностью является плоскость. Для квадратичных поверхностей однолинейчаты конусы и цилиндры; однополостный гиперболоид и гиперболический параболоид двулинейчаты.

В терминах отображения параметрического пространства u, w в объективное пространство линейчатая поверхность получается с помощью линейного интерполяирования между двумя известными граничными кривыми, ассоциированными с противоположными сторонами единичного квадрата в параметрическом пространстве, скажем, между кривыми $P(u, 0)$ и $P(u, 1)$. Поверхность задается уравнениями:

$$Q(u, w) = P(u, 0)(1 - w) + P(u, 1)w \quad (6-43)$$

или

$$[Q] = [x(x, w) \quad y(u, w) \quad z(u, w)] = [1 - w \quad w] \begin{bmatrix} P(u, 0) \\ P(u, 1) \end{bmatrix}$$

Заметим, что снова $Q(0, 0) = P(0, 0)$ и т. д., то есть концы специфицированных кривых и углы поверхности совпадают. Кроме того, два края интерполированной поверхности совпадают с заданными кривыми, т. е. $Q(u, 0) = P(u, 0)$ и $Q(u, 1) = P(u, 1)$.

Теперь предположим, что известны кривые, соответствующие $Q(0, w)$ и $P(1, w)$. Тогда линейчатая поверхность задается следующим образом:

$$Q(u, w) = P(0, w)(1 - u) + P(1, w)u \quad (6-44)$$

или

$$[Q] = [x(x, w) \quad y(u, w) \quad z(u, w)] = [1 - u \quad u] \begin{bmatrix} P(0, w) \\ P(1, w) \end{bmatrix}$$

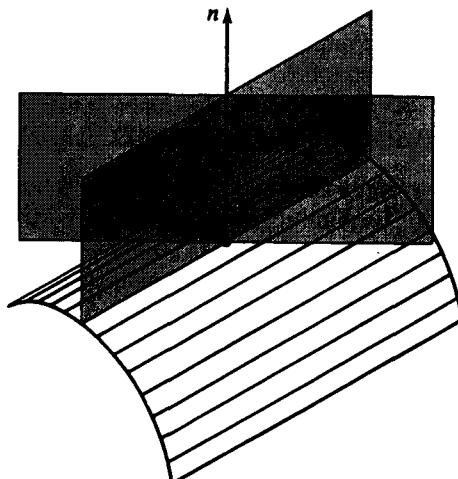


Рис. 6-27 Свойства линейчатой поверхности.

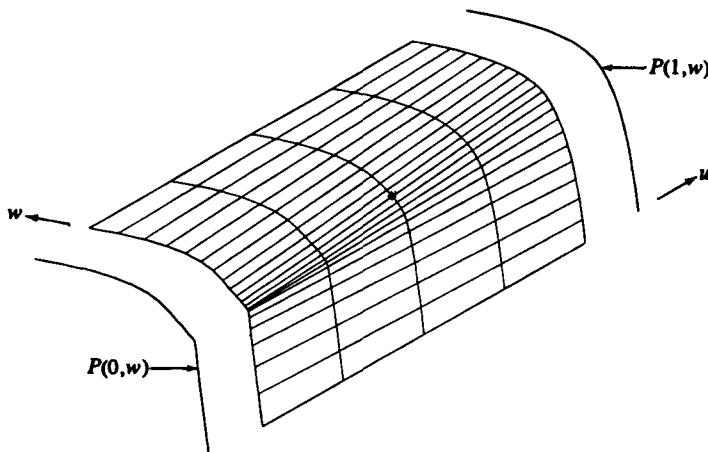


Рис. 6-28 Пример линейчатой поверхности.

В этом случае опять углы поверхности совпадают с концами заданных кривых, а соответствующие края интерполированной поверхности совпадают с заданными граничными кривыми. На рис. 6-28 приводится пример линейчатой поверхности. Изображенные на этом рисунке отстоящими на некотором расстоянии от поверхности граничные кривые являются В-сплайнами третьего порядка (см. разд. 5-9). Описанная методика иллюстрируется на примере.

Пример 6-10 Линейчатая поверхность

Рассмотрим линейчатую поверхность, сформированную линейным интерполярованием кривых $P(0, w)$ и $P(1, w)$. Найдем координаты точки $Q(u, w)$ на поверхности с параметрами $u = w = 0.5$.

$P(0, w)$ является незамкнутым В-сплайном третьего порядка ($k = 3$) с определяющими вершинами ломаной, заданными точками $B_1[0 \ 0 \ 0]$, $B_2[1 \ 1 \ 0]$, $B_3[1 \ 1 \ 0]$, $B_4[2 \ 1 \ 0]$ и $B_5[3 \ 0 \ 0]$ (см. разд. 5-9). Отметим, что вершины $B_2 = B_3$ совпадают, в результате чего на кривой получается излом. $P(1, w)$ также является незамкнутым В-сплайном третьей степени. Его вершины определяющей ломаной равны $\bar{B}_1[0 \ 0 \ 6]$, $\bar{B}_2[1 \ 1 \ 6]$, $\bar{B}_3[2 \ 1 \ 6]$, $\bar{B}_4[3 \ 0 \ 6]$.

Вспоминая обсуждение В-сплайнов, получим незамкнутые однородные узловые векторы для $P(0, w)$ и $P(1, w)$, соответственно,

$$\begin{aligned}[X] &= [0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 3 \ 3], \\ [Y] &= [0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2].\end{aligned}$$

Заметим, что ненормализованные диапазоны изменения параметров для этих двух кривых различны, $0 \leq t \leq 3$ для $P(0, w)$ и $0 \leq s \leq 2$ для $P(1, w)$. «Нормализованное» значение параметра линейчатой поверхности Q в точке $w = 0.5$ соответствует $t = 1.5$ для $P(0, w)$ и $s = 1.0$ для $P(1, w)$.

Используя уравнения (5-83) и (5-84), получим

$$\begin{aligned} P(0, w) &= P(t) \\ &= B_1 N_{1,3}(t) + B_2 N_{2,3}(t) + B_3 N_{3,3}(t) + B_4 N_{4,3}(t) + B_5 N_{5,3}(t). \end{aligned}$$

В точке $w = 0.5$ или $t = 1.5$

$$\begin{aligned} P(0, 0.5) &= P(1.5) = (0)B_1 + 0.125B_2 + 0.75B_3 + 0.125B_4 + (0)B_5 \\ &= 0.125[1 \ 1 \ 0] + 0.75[1 \ 1 \ 0] + 0.125[2 \ 1 \ 0] \\ &= [1.125 \ 1 \ 0]. \end{aligned}$$

Аналогичным образом

$$\begin{aligned} P(1, w) &= P(s) \\ &= \bar{B}_1 \bar{N}_{1,3}(s) + \bar{B}_2 \bar{N}_{2,3}(s) + \bar{B}_3 \bar{N}_{3,3}(s) + \bar{B}_4 \bar{N}_{4,3}(s). \end{aligned}$$

В точке $w = 0.5$ или $s = 1.0$

$$\begin{aligned} P(1, 0.5) &= P(1.0) = (0)\bar{B}_1 + 0.5\bar{B}_2 + 0.5\bar{B}_3 + (0)\bar{B}_4 \\ &= 0.5[1 \ 1 \ 6] + 0.5[2 \ 1 \ 6] \\ &= [1.5 \ 1 \ 6]. \end{aligned}$$

Использование уравнения (6-44) для получения точки на линейчатой поверхности даст

$$Q(u, w) = P(0, w)(1 - u) + P(1, w)u$$

и

$$\begin{aligned} Q(0.5, 0.5) &= P(0, 0.5)(1 - 0.5) + P(1, 0.5)(0.5) \\ &= 0.5[1.125 \ 1 \ 0] + 0.5[1.5 \ 1 \ 6] \\ &= [1.3125 \ 1 \ 3]. \end{aligned}$$

Результаты показаны на рис. 6-28. Жирной точкой отмечена точка поверхности, соответствующая $Q(0.5, 0.5)$. Обратите внимание на то, что кривая $P(0, w)$, содержащая излом, плавно переходит в гладкую кривую $P(1, w)$.

Особый практический интерес представляет вопрос, является ли линейчатая поверхность развертывающейся? Не все линейчатые поверхности развертывающиеся, однако все развертывающиеся поверхности являются линейчатыми.

Если поверхность развертывающаяся, то с помощью последовательности небольших поворотов вокруг образующей линии она может быть без растяжений и разрывов развернута или раскрыта в плоскость. Развертывающиеся поверхности особенно важны для листопрокатной промышленности и в меньшей степени для текстильной промышленности.

Ясно, что среди линейчатых квадратичных поверхностей развертывающимися являются конусы и цилиндры. Однако после небольшого размышления становится ясно, что ни однополостный гиперболоид (см. рис. 6-18d), ни гиперболический параболоид (см. рис. 6-26) не являются развертывающимися поверхностями, хотя они линейчатые.

Чтобы определить, будет ли развертывающейся поверхность или ее часть, необходимо рассмотреть кривизну параметрической поверхности. В произвольной

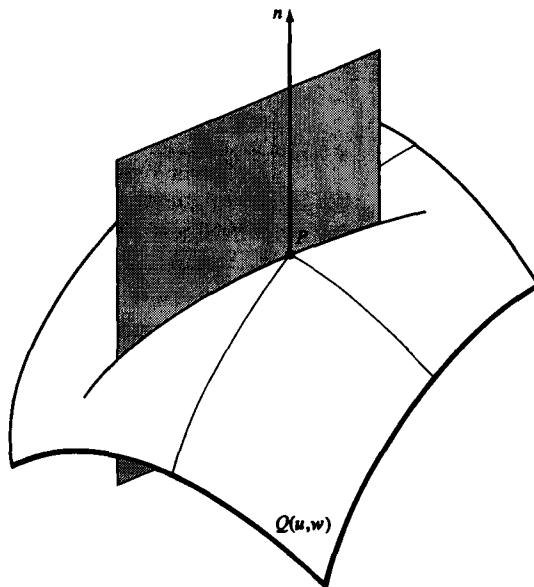


Рис. 6-29 Кривизна бипараметрической поверхности.

точке P на поверхности кривая пересечения поверхности и плоскости, содержащей нормаль к поверхности в точке P , имеет кривизну κ (см. рис. 6-29). При вращении этой плоскости вокруг нормали кривизна меняется. Великий швейцарский математик Эйлер показал, что существуют только два направления, для которых кривизна принимает минимальное и максимальное значения. Кривизны в этих направлениях называются главными кривизнами, κ_{\min} и κ_{\max} . Кроме того, направления главных кривизн ортогональны. Два сочетания главных кривизн представляют особый интерес — средняя и гауссова кривизны:

$$H = \frac{\kappa_{\min} + \kappa_{\max}}{2}, \quad (6-45)$$

$$K = \kappa_{\min} \kappa_{\max}. \quad (6-46)$$

Для развертывающейся поверхности гауссова кривизна K в любой точке равна нулю, т. е. $K = 0$. Дил [6-18] показал, что для бипараметрических поверхностей средняя и гауссова кривизны задаются выражением¹

$$H = \frac{A|Q_w|^2 - 2BQ_u \cdot Q_w + C|Q_u|^2}{2|Q_u \times Q_w|^3}, \quad (6-47)$$

$$K = \frac{AC - B^2}{|Q_u \times Q_w|^4}, \quad (6-48)$$

где

$$(ABC) = [Q_u \times Q_w] \cdot [Q_{uu} \quad Q_{uw} \quad Q_{ww}].$$

¹ Здесь подстрочные индексы используются для обозначения частных производных, т. е. $Q_u = \partial Q / \partial u$ и т. д.

Таблица 6-2 Типы поверхностей

$\kappa_{\min}, \kappa_{\max}$	K	Форма
Однаковые знаки	< 0	Эллиптическая (выпуклость или впадина)
Противоположные знаки	> 0	Гиперболическая (седловая точка)
Одна или обе кривизны равны нулю	0	Цилиндрическая/коническая (гребень, впадина, плоскость)

Как показано в табл. 6-2, знак гауссовой кривизны характеризует локальную форму поверхности: эллиптическую, гиперболическую, цилиндрическую или коническую. Так как гауссова кривизна развертывающейся поверхности должна быть нулевой, то поверхность должна быть скомпонована из цилиндрических, конических или плоских кусков. Приведенный ниже пример поможет проиллюстрировать эти рассуждения

Пример 6-11 Разворачивающаяся поверхность

Показать, что эллиптический конус является разворачивающейся поверхностью.

Перепишем уравнение (6-32) для параметрического эллиптического конуса в терминах u и w :

$$Q(u, w) = [au \cos w \quad bu \sin w \quad cu].$$

Частные производные будут

$$\begin{aligned} Q_u &= [a \cos w \quad b \sin w \quad c], \\ Q_w &= [-au \sin w \quad bu \cos w \quad 0], \\ Q_{uw} &= [-a \sin w \quad b \cos w \quad 0], \\ Q_{uu} &= [0 \quad 0 \quad 0], \\ Q_{ww} &= [-au \cos w \quad -bu \sin w \quad 0], \\ Q_u \times Q_w &= [-bcu \cos w \quad -acu \sin w \quad abu], \\ |Q_u \times Q_w|^2 &= (abu)^2 \left\{ \left(\frac{c}{a} \cos w \right)^2 + \left(\frac{c}{b} \sin w \right)^2 + 1 \right\} \neq 0 \quad u > 0 \end{aligned}$$

и

$$\begin{aligned} A &= [-bcu \cos w \quad -acu \sin w \quad abu] \cdot [0 \quad 0 \quad 0] = 0, \\ B &= [-bcu \cos w \quad -acu \sin w \quad abu] \cdot [-a \sin w \quad b \cos w \quad 0] \\ &\quad = abc u \sin w \cos w - abc u \sin w \cos w = 0, \\ C &= [-bcu \cos w \quad -acu \sin w \quad abu] \cdot [-au \cos w \quad -bu \sin w \quad 0] \\ &\quad = abc u^2 \cos^2 w + abc u^2 \sin^2 w = abc u^2. \end{aligned}$$

Таким образом, используя уравнение (6-48), получим, что в любой точке поверхности

$$K = \frac{AC - B^2}{|Q_u \times Q_w|^4} = \frac{(0)(abcu^2) - (0)}{|Q_u \times Q_w|^4} = 0$$

и, следовательно, поверхность является развертывающейся. Заметим, что хотя $|Q_u \times Q_w|^2 = 0$ для $u = 0$, из правила Лопитала получаем, что $K = 0/0 = 0$ в $u = 0$.

6-9 ЛИНЕЙНАЯ ПОВЕРХНОСТЬ КУНСА

Если известны четыре граничные кривые $P(u, 0)$, $P(u, 1)$, $P(0, w)$ и $P(1, w)$ и для внутренней части куска поверхности используется билинейная смешивающая функция, то в результате получаем линейную поверхность Кунса. На первый взгляд можно предположить, что желаемый результат дает простая сумма отдельных линейчатых поверхностей (уравнения (6-43) и (6-44)) в двух направлениях u , w :

$$Q(u, w) = P(u, 0)(1 - w) + P(u, 1)w + P(0, w)(1 - u) + P(1, w)u.$$

Однако, проверив этот результат в угловых точках куска поверхности, например

$$Q(0, 0) = P(0, 0) + P(0, 0) = 2P(0, 0)$$

и на границах, например

$$Q(0, w) = P(0, 0)(1 - w) + P(0, 1)w + P(0, w)$$

получим, что ни одно из этих значений не соответствует исходным данным. Это происходит из-за того, что угловые точки учитываются дважды, так как $P(0, 0)$ содержится в обеих граничных кривых $P(u, 0)$ и $P(0, w)$.

Правильный результат можно получить с помощью вычитания дополнительных членов, возникающих из-за удвоения угловых точек:

$$\begin{aligned} Q(u, w) = & P(u, 0)(1 - w) + P(u, 1)w + P(0, w)(1 - u) + P(1, w)u \\ & - P(0, 0)(1 - u)(1 - w) - P(0, 1)(1 - u)w \\ & - P(1, 0)u(1 - w) - P(1, 1)uw. \end{aligned} \quad (6-49)$$

Теперь в угловых точках

$$Q(0, 0) = P(0, 0) \quad \text{и т. д.}$$

и вдоль границ

$$Q(0, w) = P(0, w)$$

$$Q(u, 1) = P(u, 1)$$

и т. д.

В матричной форме уравнение (6-49) имеет вид:

$$\begin{aligned} Q(u, w) = & [1 - u \quad w] \begin{bmatrix} P(0, w) \\ P(1, w) \end{bmatrix} + [P(u, 0) \quad P(u, 1)] \begin{bmatrix} 1 - w \\ w \end{bmatrix} \\ & - [1 - u \quad w] \begin{bmatrix} P(0, 0) & P(0, 1) \\ P(1, 0) & P(1, 1) \end{bmatrix} \begin{bmatrix} 1 - w \\ w \end{bmatrix} \end{aligned}$$

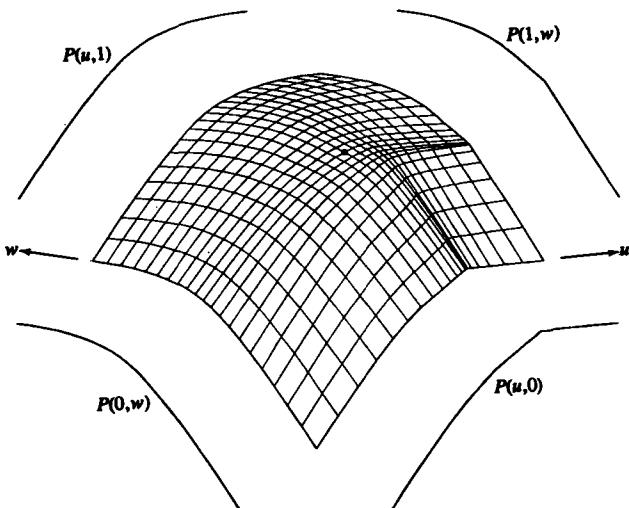


Рис. 6-30 Линейная поверхность Кунса.

или более компактно

$$Q(u, w) = [1 - u \quad u \quad 1] \begin{bmatrix} -P(0, 0) & -P(0, 1) & P(0, w) \\ -P(1, 0) & -P(1, 1) & P(1, w) \\ P(u, 0) & P(u, 1) & 0 \end{bmatrix} \begin{bmatrix} 1 - w \\ w \\ 1 \end{bmatrix}. \quad (6-50)$$

Функции $(1 - u)$, u , $(1 - w)$ и w называются функциями смешения потому, что они смешивают граничные кривые для получения внутренней формы поверхности. Линейная поверхность показана на рис. 6-30. Линейная поверхность Кунса является самой простой из поверхностей Кунса. В разд. 6-10 обсуждается более общая поверхность Кунса. Следующий пример иллюстрирует использованный выше метод.

Пример 6-12 Линейная поверхность Кунса

Найти точку с координатами $u = w = 0.5$, расположенную на линейной поверхности Кунса, если четыре граничные кривые $P(u, 0)$, $P(u, 1)$, $P(0, w)$, $P(1, w)$ задаются незамкнутыми B -сплайнами третьего порядка ($k = 3$)

$$P(u, 0) : B_1[0 \quad 0 \quad 0], B_2[1 \quad 1 \quad 0], B_3[2 \quad 1 \quad 0], B_4[3 \quad 0 \quad 0],$$

$$B_5[3 \quad 0 \quad 3]$$

$$P(u, 1) : \bar{B}_1[0 \quad 0 \quad 3], \bar{B}_2[1 \quad 1 \quad 3], \bar{B}_3[2 \quad 1 \quad 3], \bar{B}_4[2 \quad 1 \quad 3]$$

$$P(0, w) : \bar{C}_1[0 \quad 0 \quad 3], \bar{C}_2[0 \quad 1 \quad 2], \bar{C}_3[0 \quad 1 \quad 1], \bar{C}_4[0 \quad 0 \quad 0]$$

$$P(1, w) : C_1[3 \quad 0 \quad 3], C_2[3 \quad 1 \quad 2], C_3[3 \quad 1 \quad 2], C_4[3 \quad 1 \quad 1],$$

$$C_5[3 \quad 0 \quad 0]$$

Вспоминая предыдущее обсуждение B -сплайнов (см. разд. 5-9), получим узловой вектор для $P(u, 0)$ и $P(1, w)$

$$[X] = [0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 3 \quad 3]$$

Таким образом, ненормализованный диапазон параметра есть $0 \leq t \leq 3$. Для $(u, 1)$ и $P(0, w)$ узловой вектор с ненормализованным диапазоном параметра $0 \leq s \leq 2$ имеет вид

$$[Y] = [0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2].$$

Соответствующими значениями для параметров в нормализованном диапазоне $0 \leq u, w \leq 1$ являются $t = 1, 5$ и $s = 1.0$.

Воспользовавшись уравнениями (5-83) и (5-84), получим

$$P(u, 0) = P(t) = B_1 N_{1,3}(t) + B_2 N_{2,3}(t) + B_3 N_{3,3}(t) + B_4 N_{4,3}(t) + B_5 N_{5,3}(t)$$

$$\begin{aligned} P(0.5, 0) = P(1.5) &= (0)B_1 + 0.125B_2 + 0.75B_3 + 0.125B_4 + (0)B_5 \\ &= 0.125[1 \ 1 \ 3] + 0.75[2 \ 1 \ 3] + 0.125[2 \ 1 \ 3] \\ &= [1.875 \ 1 \ 3] \end{aligned}$$

$$P(u, 1) = P(s) = \bar{B}_1 \bar{N}_{1,3}(s) + \bar{B}_2 \bar{N}_{2,3}(s) + \bar{B}_3 \bar{N}_{3,3}(s) + \bar{B}_4 \bar{N}_{4,3}(s)$$

$$\begin{aligned} P(0.5, 1) = P(1.0) &= (0)\bar{B}_1 + 0.5\bar{B}_2 + 0.5\bar{B}_3 + (0)\bar{B}_4 \\ &= 0.5[1 \ 1 \ 0] + 0.5[2 \ 1 \ 0] \\ &= [1.5 \ 1 \ 0] \end{aligned}$$

$$P(0, w) = P(s) = \bar{C}_1 \bar{N}_{1,3}(s) + \bar{C}_2 \bar{N}_{2,3}(s) + \bar{C}_3 \bar{N}_{3,3}(s) + \bar{C}_4 \bar{N}_{4,3}(s)$$

$$\begin{aligned} P(0, 0.5) = P(1.0) &= (0)\bar{C}_1 + 0.5\bar{C}_2 + 0.5\bar{C}_3 + (0)\bar{C}_4 \\ &= 0.5[0 \ 1 \ 2] + 0.5[0 \ 1 \ 1] \\ &= [0 \ 1 \ 1.5] \end{aligned}$$

$$P(1, w) = P(t) = C_1 N_{1,3}(t) + C_2 N_{2,3}(t) + C_3 N_{3,3}(t) + C_4 N_{4,3}(t) + C_5 N_{5,3}(t)$$

$$\begin{aligned} P(1, 0.5) = P(1.5) &= (0)C_1 + 0.125C_2 + 0.75C_3 + 0.125C_4 + (0)C_5 \\ &= 0.125[3 \ 1 \ 2] + 0.75[3 \ 1 \ 2] + 0.125[3 \ 1 \ 1] \\ &= [3 \ 1 \ 1.875] \end{aligned}$$

Теперь, используя уравнение (6-50), имеем

$$Q(u, w) = [1 - u \ u \ 1] \begin{bmatrix} -P(0, 0) & -P(0, 1) & P(0, w) \\ -P(1, 0) & -P(1, 1) & P(1, w) \\ P(u, 0) & P(u, 1) & 0 \end{bmatrix} \begin{bmatrix} 1 - w \\ w \\ 1 \end{bmatrix},$$

$$\begin{aligned} Q(0.5, 0.5) &= [0.5 \ 0.5 \ 1] \times \\ &\quad \times \begin{bmatrix} -[0 \ 0 \ 3] & -[0 \ 0 \ 0] & [0 \ 1 \ 1.5] \\ -[3 \ 0 \ 3] & -[3 \ 0 \ 0] & [3 \ 1 \ 1.875] \\ [1.875 \ 1 \ 3] & [1.5 \ 1 \ 0] & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 1 \end{bmatrix} = \\ &= [1.6375 \ 2.25 \ 1.6375]. \end{aligned}$$

Полученные результаты изображены на рис. 6-30. Отметим наличие плоского участка на поверхности, гауссова кривизна в этой области равна нулю. Следовательно, эта часть поверхности развертывающаяся. В остальной области гауссова кривизна положительна и поверхность не является развертывающейся.

6-10 БИКУБИЧЕСКАЯ ПОВЕРХНОСТЬ КУНСА

Хотя аналитические поверхности, например квадратичные, линейчатые и простые куски линейно интерполированных поверхностей, обсуждавшиеся в предыдущих разделах, и важны для конструирования и производства, но для многих приложений они не обладают достаточной гибкостью. Описание поверхностей, обсуждаемое в этом и дальнейших разделах главы, предоставляют необходимую гибкость с помощью использования полиномов более высоких степеней как для граничных кривых куска поверхности, так и для внутренних смешивающих функций. Поверхности, сгенерированные с помощью объединения таких кусков, называются скульптурными поверхностями. Для описания скульптурных поверхностей фундаментальное значение имеет кусок бикубической поверхности Кунса (см. [6-19]).

Для всех четырех граничных кривых куска бикубической поверхности Кунса используются нормализованные кубические сплайны (см. разд. 5-4). Для задания внутренней части куска используются кубические смешивающие функции. Таким образом, каждую граничную кривую можно представить в общем виде

$$P(t) = B_1 + B_2 t + B_3 t^2 + B_4 t^3, \quad 0 \leq t \leq 1. \quad (5-2)$$

Для одного сегмента нормализованного кубического сплайна с известными касательными и координатными векторами на концах, каждая из четырех граничных кривых, $P(u, 0)$, $P(u, 1)$, $P(0, w)$ и $P(1, w)$ задается следующим образом (см.

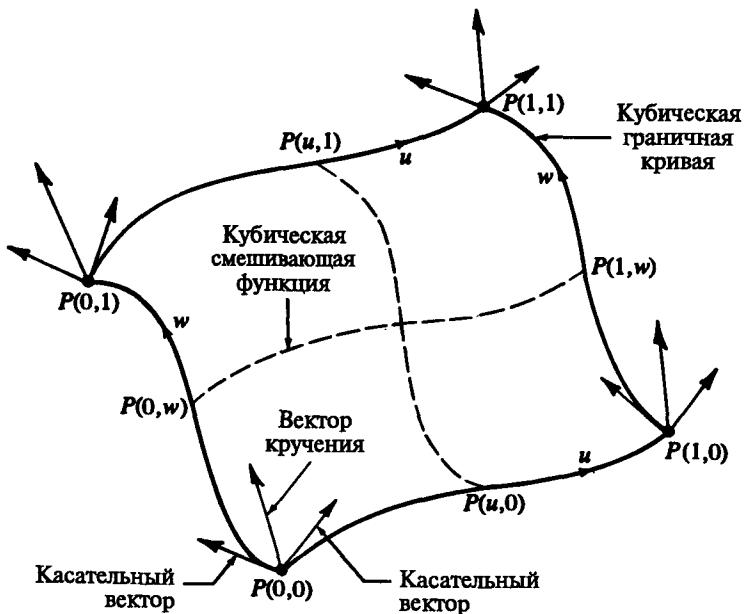


Рис. 6-31 Геометрия для куска бикубической поверхности Кунса.

уравнения (5-24), (5-26), (5-27):

$$\begin{aligned} P(t) &= [T][N][G] = \\ &= [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P'_1 \\ P'_2 \end{bmatrix} \quad 0 \leq t \leq 1, \end{aligned} \quad (5-27)$$

где t становится соответственно u или w , а P_1, P_2, P'_1, P'_2 — координатные и касательные векторы на концах соответствующей граничной кривой (см. рис. 6-31). Кубическая смешивающая функция, использованная для обоих параметрических направлений, идентична функции, использовавшейся для интерполяции внутренней части нормализованного кубического сплайна, т. е.

$$\begin{aligned} [F] &= [F_1(t) \quad F_2(t) \quad F_3(t) \quad F_4(t)] = [T][N] \\ &= [t^3 \quad t^2 \quad t \quad 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (5-26)$$

Расписывая смешивающие функции, имеем

$$F_1(t) = 2t^3 - 3t^2 + 1, \quad (5-25a)$$

$$F_2(t) = -2t^3 + 3t^2, \quad (5-25b)$$

$$F_3(t) = t^3 - 2t^2 + t, \quad (5-25c)$$

$$F_4(t) = t^3 - t^2, \quad (5-25d)$$

где t есть либо u либо w , соответственно. Эти функции изображены на рис. 5-8. Тогда определение куска бикубической поверхности Кунса записывается в виде

$$\begin{aligned} Q(u, w) &= [F_1(u) \quad F_2(u) \quad F_3(u) \quad F_4(u)] \times \\ &\quad \begin{bmatrix} P(0, 0) & P(0, 1) & P_w(0, 0) & P_w(0, 1) \\ P(1, 0) & P(1, 1) & P_w(1, 0) & P_w(1, 1) \\ P_u(0, 0) & P_u(0, 1) & P_{uw}(0, 0) & P_{uw}(0, 1) \\ P_u(1, 0) & P_u(1, 1) & P_{uw}(1, 0) & P_{uw}(1, 1) \end{bmatrix} \begin{bmatrix} F_1(w) \\ F_2(w) \\ F_3(w) \\ F_4(w) \end{bmatrix} \end{aligned} \quad (6-51)$$

для $0 \leq u \leq 1$ и $0 \leq w \leq 1$. С помощью уравнения (5-26) это можно записать более компактно:

$$Q(u, w) = [U][N][P][N]^T[W], \quad (6-52)$$

где $[U] = [u^3 \quad u^2 \quad u \quad 1]$ и $[W]^T = [w^3 \quad w^2 \quad w \quad 1]$.

Матрица $[P]$ в уравнении (6-51) содержит всю геометрическую информацию, необходимую для генерации одного бикубического куска поверхности Кунса. Подматрицы размера 2×2 в каждом углу $[P]$ содержат

$$[P] = \begin{bmatrix} \text{угловые коор-} & : & w\text{-каса-} \\ \text{динатные} & : & \text{тельные} \\ \text{векторы} & : & \text{векторы} \\ \dots & \dots & \dots \\ i\text{-каса-} & : & \text{векторы} \\ \text{тельные} & : & \text{кручения} \\ \text{векторы} & : & \end{bmatrix}.$$

Таким образом, кусок бикубической поверхности определяется четырьмя координатными векторами в угловых точках, восемью касательными векторами (по два в каждом углу), четырьмя векторами кручения в углах и четырьмя кубическими смещающими функциями F_1, F_2, F_3 и F_4 .

Пример куска бикубической поверхности представлен на рис. 6-32. Заметим, что каждая из изображенных на рис. 6-32 изопараметрических линий является нормализованным кубическим сплайном. Несколько кусков могут быть соединены вдоль C^1 -гладких границ для образования более сложных поверхностей, являющихся C^1 -гладкими везде. Детально условия объединения кусков бикубических поверхностей обсуждаются в [6-19].

Параметрические производные в любой точке на бикубической поверхности получаются с помощью формального дифференцирования уравнения (6-52):

$$Q_u(u, w) = [U][N][P][N]^T[W], \quad (6-53)$$

$$Q_w(u, w) = [U][N][P][N]^T[W'], \quad (6-54)$$

$$Q_{uw}(u, w) = [U'][N][P][N]^T[W'], \quad (6-55)$$

$$Q_{uu}(u, w) = [U''][N][P][N]^T[W], \quad (6-56)$$

$$Q_{ww}(u, w) = [U][N][P][N]^T[W''], \quad (6-57)$$

где штрихи обозначают дифференцирование по соответствующей переменной и

$$[U'] = [3u^2 \quad 2u \quad 1 \quad 0],$$

$$[W']^T = [3w^2 \quad 2w \quad 1 \quad 0],$$

$$[U''] = [6u \quad 2 \quad 0 \quad 0],$$

$$[W'']^T = [6w \quad 2 \quad 0 \quad 0].$$

Нормаль к поверхности, используемая для определения невидимых поверхностей, вычисления освещенности (см. [1-11]) и создания управляющих программ для станков, задается выражением

$$\mathbf{n} = Q_u \times Q_w.$$

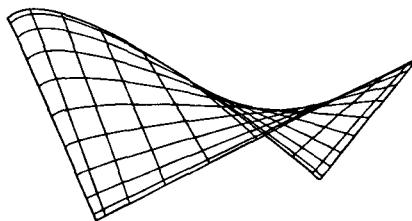


Рис. 6-32 Кусок бикубической поверхности Кунса.

Подробный пример иллюстрирует метод вычислений для бикубических кусков.

Пример 6-13 Кусок бикубической поверхности Кунса

Определить точку на куске бикубической поверхности Кунса, соответствующую параметрам $u = w = 0.5$. Вектор-точки для четырех углов равны $P(0,0) = [-100 \ 0 \ 100]$, $P(1,0) = [-100 \ -100 \ -100]$, $P(1,0) = [100 \ -100 \ -100]$, $P(1,1) = [100 \ 0 \ -100]$. Касательные векторы равны $P_u(0,0) = [100 \ 100 \ 0]$, $P_u(0,1) = [1 \ 1 \ 0]$, $P_u(1,0) = [1 \ -1 \ 0]$, $P_u(1,1) = [1 \ -1 \ 0]$, $P_w(0,0) = [0 \ 10 \ -10]$, $P_w(0,1) = [0 \ -1 \ -1]$, $P_w(1,0) = [0 \ 1 \ -1]$, $P_w(1,1) = [0 \ -1 \ -1]$. Векторы кручения равны $P_{uw}(0,0) = [0 \ 0 \ 0]$, $P_{uw}(0,1) = [0.1 \ 0.1 \ 0.1]$, $P_{uw}(1,0) = [0.1 \ -0.1 \ -0.1]$, $P_{uw}(1,1) = [0 \ 0 \ 0]$. Определить параметрические производные Q_u , Q_w и вектор нормали для $u = w = 0.5$.

Используя матричную запись из уравнения (6-51), получим

$$Q(u, w) = [U][N] \begin{bmatrix} [-100 \ 0 \ 100] & [-100 \ -100 \ -100] \\ [100 \ -100 \ 100] & [100 \ 0 \ -100] \\ [100 \ 100 \ 0] & [1 \ 1 \ 0] \\ [1 \ -1 \ 0] & [1 \ -1 \ 0] \\ [0 \ 10 \ -10] & [0 \ -1 \ -1] \\ [0 \ 1 \ -1] & [0 \ -1 \ -1] \\ [0 \ 0 \ 0] & [0.1 \ 0.1 \ 0.1] \\ [0.1 \ -0.1 \ -0.1] & [0 \ 0 \ 0] \end{bmatrix} [N]^T [W]^T$$

Для $u = 0.5$ смещающие функции по u имеют вид:

$$\begin{aligned} [U][N] &= [u^3 \ u^2 \ u \ 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\ &= [0.125 \ 0.25 \ 0.5 \ 1] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\ &= [0.5 \ 0.5 \ 0.125 \ -0.125] \end{aligned}$$

Для $w = 0.5$ смешивающие функции по w :

$$\begin{aligned}[N]^T[W] &= \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} w^3 \\ w^2 \\ w \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.125 \\ 0.25 \\ 0.5 \\ 1 \end{bmatrix} \\ &= [0.5 \quad 0.5 \quad 0.125 \quad -0.125]^T.\end{aligned}$$

Таким образом, x компонента $Q(u, w)$ при $u = w = 0.5$ равна

$$\begin{aligned}Q_x(0.5, 0.5) &= [0.5 \quad 0.5 \quad 0.125 \quad -0.125] \times \\&\quad \times \begin{bmatrix} -100 & -100 & 0 & 0 \\ 100 & 100 & 0 & 0 \\ 100 & 1 & 0 & 0.1 \\ 1 & 1 & 0.1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.125 \\ -0.125 \end{bmatrix} = \\&= [12.375 \quad 0 \quad -0.0125 \quad 0.0125] \begin{bmatrix} 0.5 \\ 0.5 \\ 0.125 \\ -0.125 \end{bmatrix} = \\&= 6.18\end{aligned}$$

Аналогично y и z компоненты таковы:

$$\begin{aligned}Q_y(0.5, 0.5) &= [0.5 \quad 0.5 \quad 0.125 \quad -0.125] \times \\&\quad \times \begin{bmatrix} 0 & -100 & 10 & -1 \\ -100 & 0 & 1 & -1 \\ 100 & 1 & 0 & 0.1 \\ -1 & -1 & -0.1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.125 \\ -0.125 \end{bmatrix} = \\&= -42.75 \\Q_z(0.5, 0.5) &= [0.5 \quad 0.5 \quad 0.125 \quad -0.125] \times \\&\quad \times \begin{bmatrix} 100 & -100 & -10 & -1 \\ 100 & -100 & -1 & -1 \\ 0 & 0 & 0 & 0.1 \\ 0 & 0 & -0.1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.125 \\ -0.125 \end{bmatrix} = \\&= -0.56\end{aligned}$$

Следовательно, $Q(0.5, 0.5) = [6.18 \quad -42.75 \quad -0.56]$. Вся поверхность изображена на рис. 6-32.

Обратимся теперь к производным, вспомним уравнения (6-53)–(6-55) и заметим, что

$$\begin{aligned}[U'][N] &= [3u^2 \quad 2u \quad 1 \quad 0][N] = \\&= [0.75 \quad 1 \quad 1 \quad 0] \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} = \\&= [-1.5 \quad 1.5 \quad -0.25 \quad -0.25]\end{aligned}$$

$$\begin{aligned}[N]^T [W'] &= [N]^T [3w^2 \quad 2w \quad 1 \quad 0]^T = \\ &= \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.75 \\ 1 \\ 1 \\ 0 \end{bmatrix} = \\ &= [-1.5 \quad 1.5 \quad -0.25 \quad -0.25]^T\end{aligned}$$

параметрические производные в $u = w = 0.5$ равны

$$\begin{aligned}Q_{uw}(0.5, 0.5) &= [-1.5 \quad 1.5 \quad -0.25 \quad -0.25] \times \\ &\times \begin{bmatrix} -100 & -100 & 0 & 0 \\ 100 & 100 & 0 & 0 \\ 100 & 1 & 0 & 0.1 \\ 1 & 1 & 0.1 & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.125 \\ -0.125 \end{bmatrix} = \\ &= [274.75 \quad 299.5 \quad -0.025 \quad -0.025] \begin{bmatrix} 0.5 \\ 0.5 \\ 0.125 \\ -0.125 \end{bmatrix} = \\ &= 287.13\end{aligned}$$

Q_{uy} и Q_{uz} находятся аналогичным образом. В результате получим

$$Q_u(0.5, 0.5) = [287.13 \quad -14.06 \quad 1.69].$$

Теперь

$$\begin{aligned}Q_{ww}(0.5, 0.5) &= [0.5 \quad 0.5 \quad 0.125 \quad -0.125] \times \\ &\times \begin{bmatrix} -100 & -100 & 0 & 0 \\ 100 & 100 & 0 & 0 \\ 100 & 1 & 0 & 0.1 \\ 1 & 1 & 0.1 & 0 \end{bmatrix} \begin{bmatrix} -1.5 \\ 1.5 \\ -0.25 \\ -0.25 \end{bmatrix} = \\ &= [12.38 \quad 0 \quad -0.0125 \quad 0.0125] \begin{bmatrix} -1.5 \\ 1.5 \\ -0.25 \\ -0.25 \end{bmatrix} = \\ &= -18.56\end{aligned}$$

Q_{wy} и Q_{wz} находятся аналогичным образом, откуда

$$Q_w(0.5, 0.5) = [-18.56 \quad -19.69 \quad -298.38]$$

И наконец, вектор нормали имеет вид:

$$\begin{aligned}\mathbf{n} &= Q_u \times Q_w = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ Q_{uw} & Q_{uy} & Q_{uz} \\ Q_{ww} & Q_{wy} & Q_{wz} \end{vmatrix} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 287.13 & -14.06 & 1.69 \\ -18.56 & -19.69 & -298.38 \end{vmatrix} \\ &= 4227.5\mathbf{i} + 85641.2\mathbf{j} - 5915.5\mathbf{k}.\end{aligned}$$

Единичная нормаль равна

$$\hat{\mathbf{n}} = [0.049 \quad 0.996 \quad -0.069]$$

Заметим, что векторы кручения в угловых точках слегка сместили нормаль от направления y .

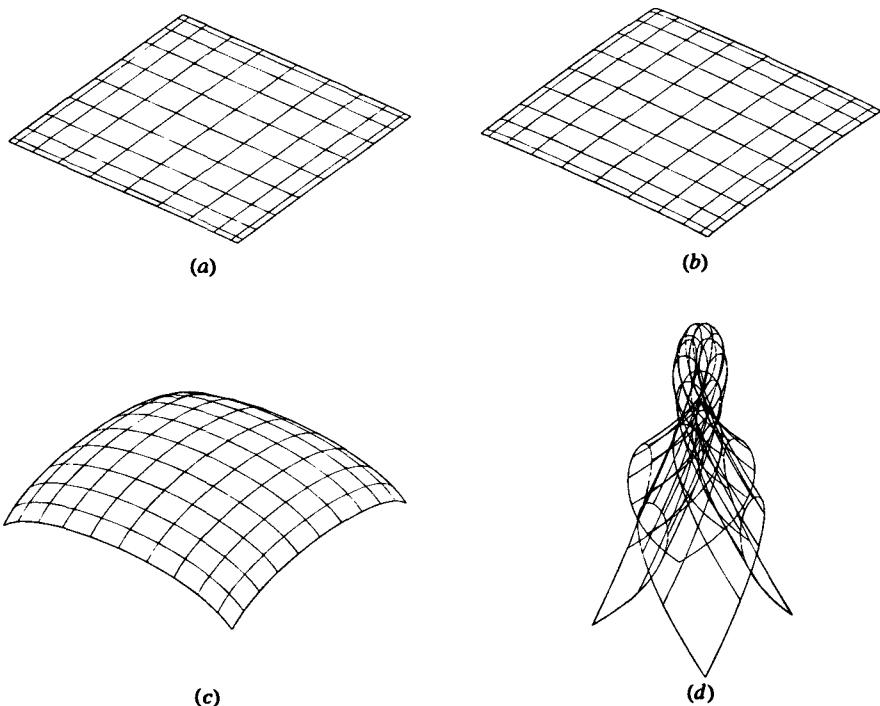


Рис. 6-33 Действие на кусок бикубической поверхности варьирования величины касательного вектора. Значения компонент касательного вектора равны (a) 1; (b) 10; (c) 100; (d) 1000.

Одна из проблем, возникающих при использовании бикубических кусков поверхности, заключается в несоответствии значений векторов положения, кручения и касательных векторов. Все они в общем случае имеют различные порядки величин. Например, для куска поверхности со стороной 10 единиц, касательные векторы могут иметь значения порядка 10 единиц векторы кручения — 100 единиц и векторы положения — 1000 единиц. Кроме того, не всегда очевиден результат изменения величины и направления касательных векторов и/или векторов кручения.

На рис. 6-33 показан результат варьирования величин касательных векторов в угловых точках куска бикубической поверхности. Векторы положения углов куска лежат в плоскости $y = 0$, отцентрированные относительно начала координат в $\pm 100x, z$. Касательные векторы лежат в плоскостях, расположенных перпендикулярно плоскости угловых точек и содержащих смежные углы куска поверхности. Направление касательных векторов выбрано таким образом, чтобы сделать кубическую граничную кривую выпуклой вверх («горбом вверх»), например $P_u(0, 0) = [1 \ 1 \ 0]$, $P_u(1, 0) = [1 \ -1 \ 0]$, и т. д. Величина векторов кручения равна нулю. Если значение компоненты u каждого касательного вектора равно нулю, то кусок поверхности плоский; если не равно нулю, то кусок изогнут.

На рис. 6-33 a изображен кусок бикубической поверхности с единичными значениями компонент касательного вектора. Хотя поверхность выглядит плоской, на самом деле она слегка округлена. Компонента y в центре поверхности, т. е. в $x = z = 0$ равна 0.5. На рисунках 6-33 $b-d$ показаны бикубические поверхности, для которых значения компонент касательного вектора увеличены соответственно на 1, 2 или 3 порядка величины, т. е. равны 10, 100 и 1000. Значения компоненты y в центральной точке теперь равны 5, 50 и 500. Заметим, что они также увеличиваются на порядок величины всякий раз, когда увеличивается значение касательного вектора.

Особый интерес представляет рис. 3-33 d . Изображенная на нем поверхность является самопересекающейся. Чтобы лучше понять этот результат, можно вспомнить обсуждение кубических сплайнов кривых в разд. 5-3 и в особенности рис. 5-10. Кроме того, заметим, что каждая изопараметрическая линия на поверхности является кубической сплайновой кривой. Каждая из изопараметрических линий образует петлю, показывая, что значение касательного вектора в угловой точке превысило критическое значение.

Результат изменения направления касательного вектора в угловой точке показан на рис. 6-34. Здесь изменены знаки компонент y касательных векторов $P(0, 1)$; их значения стали равны 100. Заметим, что на рис. 6-34 правый передний угол поверхности теперь вогнутый, а не выпуклый, как на рис. 6-33 c .

Кручение было ранее упомянуто в разд. 6-5 при обсуждении куска сферической поверхности. Там оно определялось как смешанная производная поверхности в угловых точках. В действительности вектор кручения в угловой точке позволяет модифицировать внутреннюю форму поверхности без модификации касательных векторов в этой точке. На рис. 6-35 показан результат модификации вектора кручения в одной угловой точке $P(0, 0)$ на поверхности. Для каждой из поверхностей на рис. 6-35 $P(0, 0)$ является нижней угловой точкой. Поверхность аналогична представленной на рис. 6-33 b , за исключением ненулевого вектора кручения в $P(0, 0)$. На рисунках 6-35 a и b со значениями компонент вектора кручения 10 и 100, соответственно, показан небольшой видимый эффект ненулевого вектора кручения. На рис. 6-35 c поверхность имеет выраженную выпуклую форму благодаря ненулевому вектору кручения. Рис. 6-35 d иллюстрирует действие больших значений этого вектора. Заметим, однако, что в отличие от рис. 6-33 d поверх-

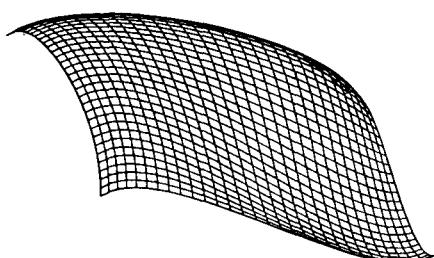


Рис. 6-34 Действие на кусок бикубической поверхности изменения направления касательного вектора.

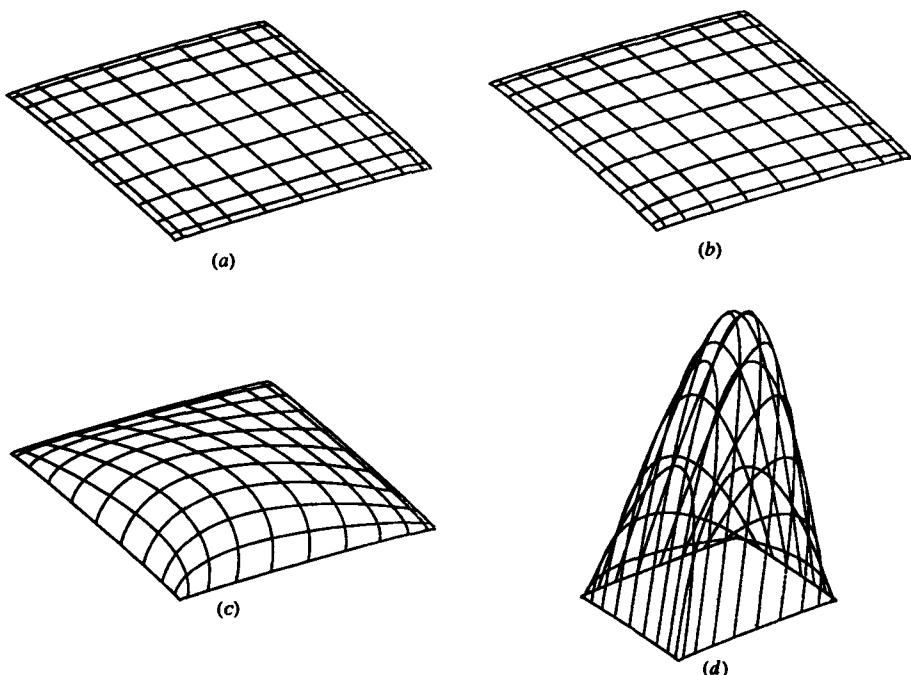


Рис. 6-35 Влияние вектора кручения на кусок бикубической поверхности. Значение компоненты вектора кручения в точке $P(0,0)$ равно (a) 10; (b) 100; (c) 1000; (d) 10000.

ность не самопересекается. Дополнительное тщательное исследование рис. 6-35 d показывает, что поверхность слегка вогнута в трех угловых точках с нулевым кручением.

Поверхность с ненулевым кручением в угловой точке является в ней локально «плоской». Поверхность с нулевым кручением во всех четырех углах, как это показано на рис. 6-33, называется куском поверхности Фергюсона или F куском [6-20]. При совместном объединении для формирования более сложных поверхностей местоположение угловых точек кусков вполне очевидно. По этой причине такие куски поверхностей редко используются на практике.

Хотя приведенное здесь обсуждение ограничивается четырехсторонними кусками поверхностей, и другие формы кусков, как это показано на рис. 6-36, тоже имеют значение. Треугольные куски поверхностей рассмотрены в [6-21] и [6-22]. Пятиугольные куски поверхностей обсуждаются в [6-23].

6-11 ПОВЕРХНОСТИ БЕЗЬЕ

Бикубические поверхности Кунса предоставляют гибкое и мощное средство разработки поверхностей. Однако их практическое использование, как и для кубических сплайновых кривых, затрудняется необходимостью задания точной, интуитивно неочевидной математической информации, например координат точек,

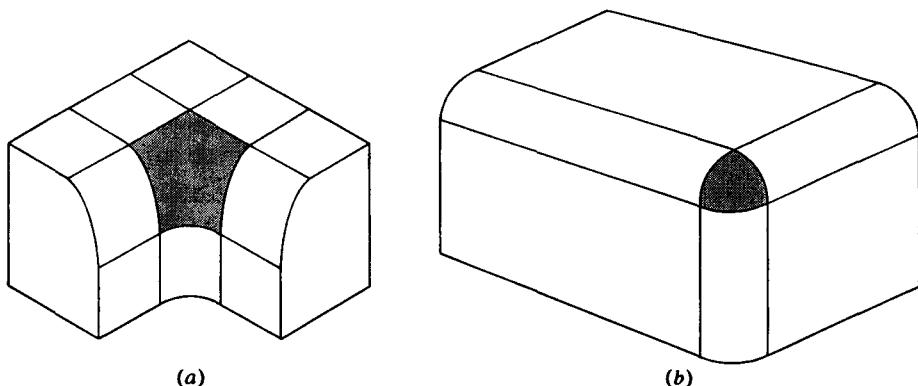


Рис. 6-36 Нечетырехугольные куски. (a) Пятиугольный; (b) треугольный.

касательных векторов и векторов кручения. Возникающие проблемы иллюстрируются на рисунках 6-33 — 6-35. Большинство из этих проблем можно преодолеть, распространив понятия кривых Безье на поверхности.

Декартово или тензорное произведение поверхности Безье задается в виде

$$Q(u, w) = \sum_{i=0}^n \sum_{j=0}^m B_{i,j} J_{n,i}(u) K_{m,j}(w), \quad (6-58)$$

где $J_{n,i}(u)$ и $K_{m,j}(w)$ есть базисные функции Бернштейна в параметрических направлениях u и w (см. уравнения 5-63 и 5-64). Для удобства повторим здесь определение, приведенное ранее в разд. 5-8

$$J_{n,i}(u) = \binom{n}{i} u^i (1-u)^{n-i}, \quad (6-63)$$

$$K_{m,j}(w) = \binom{m}{j} w^j (1-w)^{m-j}$$

с

$$\binom{n}{j} = \frac{n!}{i!(n-i)!}, \quad (6-64)$$

$$\binom{m}{j} = \frac{m!}{j!(m-j)!}$$

Здесь элементы $B_{i,j}$ являются вершинами задающей полигональной сетки, как это показано на рис. 6-37. Индексы n и m на единицу меньше числа вершин многогранника в направлениях u и w , соответственно. Для четырехсторонних кусков поверхностей задающая полигональная сетка должна быть топологически прямоугольной, т. е. должна иметь одинаковое количество вершин в каждом «ряду».

Снова, как и для кривых Безье, из-за того что для смещающих функций используется базис Бернштейна, многие свойства поверхности известны. Например:

Степень поверхности в каждом параметрическом направлении на единицу меньше числа вершин задающего многогранника в этом направлении.

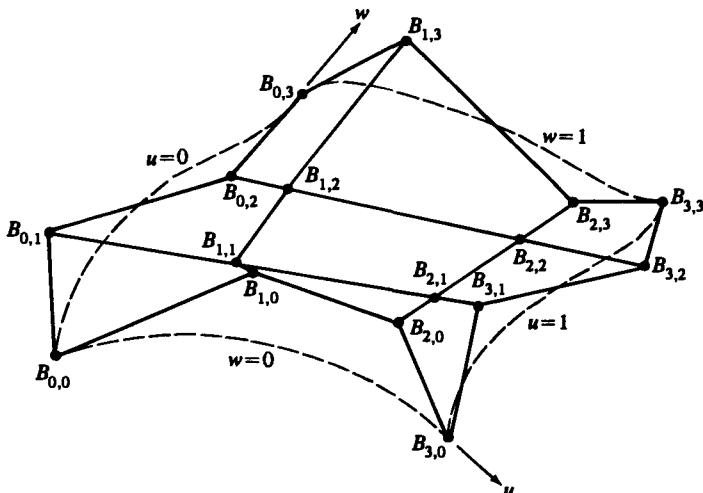


Рис. 6-37 Поверхность Безье и вершины характеристического многогранника.

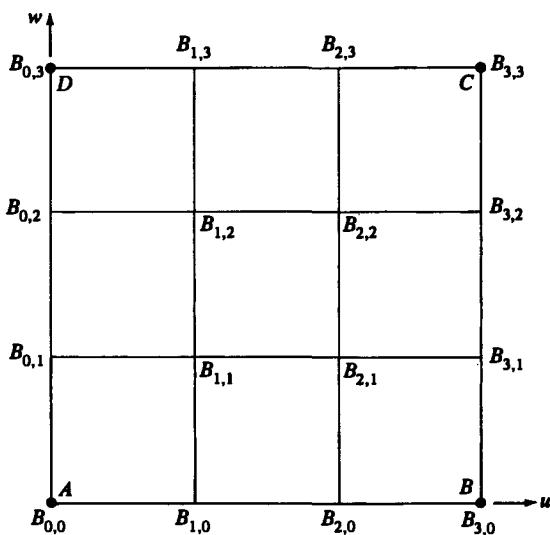


Рис. 6-38 Схема задающей полигональной сетки 4×4 для поверхности Безье.

Гладкость поверхности в каждом параметрическом направлении на две единицы меньше числа вершин задающего многогранника в этом направлении. Поверхность отображает в общем виде форму задающей полигональной сетки. Совпадают только угловые точки задающей полигональной сетки и поверхности.

Поверхность содержится внутри выпуклой оболочки задающей полигональной сетки.

Поверхность не проявляет свойства затухания изменений. Это свойство не определено и неизвестно для поверхностей двух переменных.

Поверхность инвариантна относительно аффинного преобразования.

Каждая из граничных кривых поверхности Безье является кривой Безье. Запомним этот факт и рассмотрим задающую полигональную сетку для бикубической поверхности Безье размера 4×4 , изображенную схематично на рис. 6-38. Легко видеть, что направление и величина касательных векторов в угловых точках куска управляются положением соседних точек вдоль сторон сетки. А именно касательные векторы в направлениях u, w в точке A управляются вершинами полигональной сетки $B_{0,1}$ и $B_{1,0}$, соответственно. Аналогичным образом, вершины полигональной сетки $B_{2,0}, B_{3,1}, B_{3,2}, B_{2,3}$ и $B_{1,3}, B_{0,2}$ управляют касательными векторами в угловых точках B, C, D , соответственно. Четыре внутренние вершины полигональной сетки, $B_{1,1}, B_{2,1}, B_{2,2}$ и $B_{1,2}$ влияют на направление и величину векторов кручения в угловых точках A, B, C, D куска поверхности. Следовательно, пользователь может управлять формой куска поверхности, не зная конкретных значений касательных векторов и векторов кручения.

На рис. 6-39 показано несколько бикубических поверхностей Безье и их задающих полигональных сеток. Базовая полигональная сетка имеет размер 4×4 и отцентрирована относительно начала координат с угловыми точками, находящимися в ± 15 по x, z . Компонента y угловых вершин равна нулю. У всех других вершин эта компонента равна пяти. Базовая полигональная сетка и соответствующая ей поверхность Безье изображены на рис. 6-39a. На рис. 6-39 точка $B_{0,0}$ является левой угловой вершиной, а $B_{3,3}$ — правой угловой вершиной. Заметим, что центральные вершины базовой полигональной сетки образуют плоский крест (показанный затененным). Следовательно, центр получившейся поверхности minimally изогнут, хотя и не плоский.

На рис. 6-39b проиллюстрирован эффект увеличения в 2 раза величины касательного вектора в точке $B_{0,0}$ в обоих параметрических направлениях u и w с помощью перемещения точек $B_{1,0}$ и $B_{0,1}$. Вектор кручения не меняется. Отметим увеличение кривизны граничных кривых, отвечающих значениям параметров $u = 0$ и $w = 0$, и соответствующее изменение внутренности поверхности.

На рис. 6-39c показано действие изменения направления касательных векторов в точке $B_{0,3}$ в обоих параметрических направлениях u и w с помощью перемещения точек $B_{0,2}$ и $B_{1,3}$. Отметим изменение знака кривизны граничной кривой около точки $B_{0,3}$ и формы внутренней части поверхности по сравнению с базовой поверхностью.

На рис. 6-39d проиллюстрирован результат удвоения величины вектора кручения в точке $B_{0,0}$ без изменения его направления. В этом случае перемещается только точка $B_{1,1}$. Эффект этого изменения неуловим, но тем не менее важен для конструирования. Внимательное сравнение с базовой поверхностью на рис. 6-39a показывает, что параметрические линии вблизи точки $B_{0,0}$ имеют большую кривизну. Этот эффект распространяется приблизительно до центра поверхности.

В матричном виде декартово произведение поверхности Безье задается выражением

$$Q(u, w) = [U][N][B][M]^T[W], \quad (6-59)$$

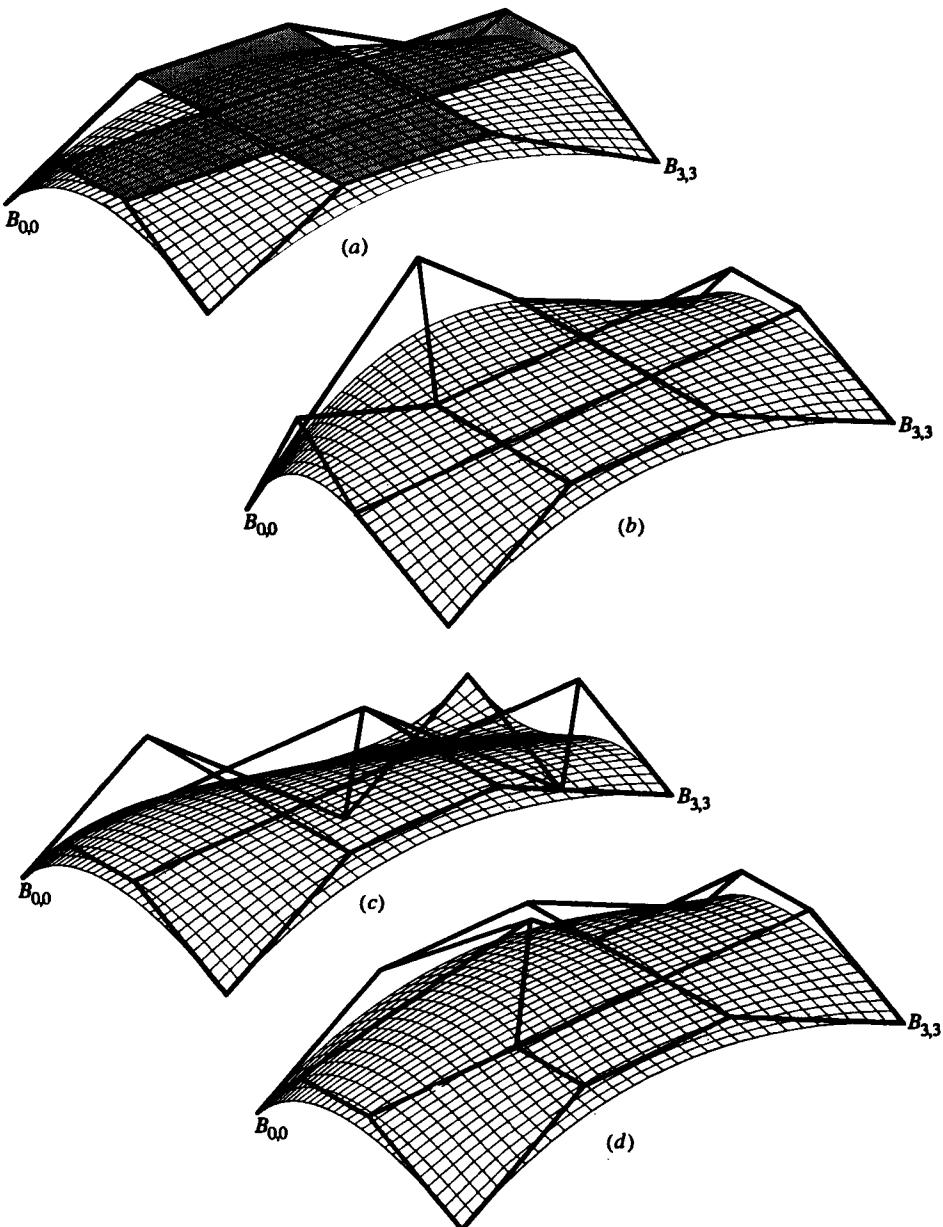


Рис. 6-39 Бикубические поверхности Безье. (a) Основная поверхность; (b) эффект изменения величины обоих касательных векторов в $B_{0,0}$; (c) эффект изменения направления касательного вектора в $B_{0,3}$; (d) эффект изменения величины вектора кручения в $B_{0,0}$.

где

$$\begin{aligned}[U] &= [u^n \ u^{n-1} \ \dots \ 1] \\ [W] &= [w^m \ w^{m-1} \ \dots \ 1]^T \\ [B] &= \begin{bmatrix} B_{0,0} & \dots & B_{0,m} \\ \vdots & \ddots & \vdots \\ B_{n,0} & \dots & B_{n,m} \end{bmatrix},\end{aligned}$$

а матрицы $[N]$ и $[M]$ задаются уравнением (5-70) или (5-71).

Для специального случая бикубической поверхности Безье размера 4×4 уравнение (6-59) сокращается до

$$Q(u, w) = [u^3 \ u^2 \ u \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w^3 \\ w^2 \\ w \\ 1 \end{bmatrix}. \quad (6-60)$$

Поверхность Безье не обязательно должна быть квадратной. Для сетки размера 5×3 уравнение (6-59) превращается в

$$Q(u, w) = [u^4 \ u^3 \ u^2 \ u \ 1] \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & -12 & -12 & 4 & 0 \\ 6 & -12 & 6 & 0 & 0 \\ -4 & 4 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} \\ B_{1,0} & B_{1,1} & B_{1,2} \\ B_{2,0} & B_{2,1} & B_{2,2} \\ B_{3,0} & B_{3,1} & B_{3,2} \\ B_{4,0} & B_{4,1} & B_{4,2} \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} w^2 \\ w \\ 1 \end{bmatrix}. \quad (6-61)$$

Поверхность Безье размера 5×3 состоит из полиномиальных кривых четвертой степени в параметрическом направлении u и из квадратичных полиномиальных кривых в направлении w . Пример такой поверхности Безье показан на рис. 6-40. В данном случае, как это показано на рис. 6-40b, изменение центральной вершины стороны задающей сетки с пятью вершинами не влияет на касательные векторы в угловых точках.

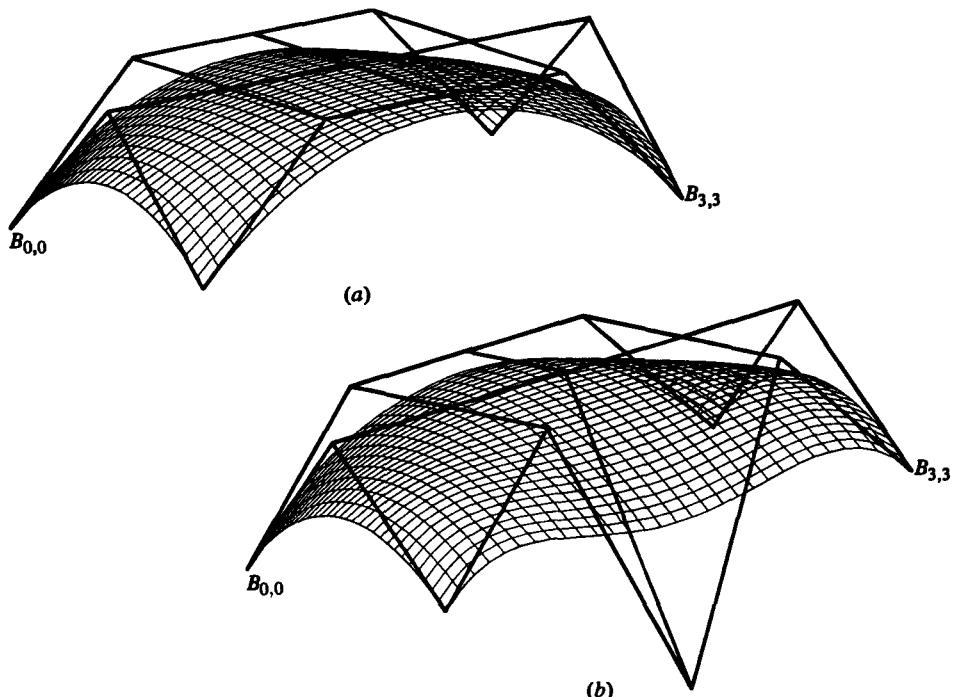


Рис. 6-40 Поверхность Безье размера 5×3 . (a) Основная поверхность; (b) эффект изменения центральной вершины граничной ломаной с пятью вершинами.

Производные поверхности Безье получаются с помощью формального дифференцирования уравнения (6-58) или (6-59). Если воспользоваться уравнением (6-58), то первые и вторые параметрические производные будут

$$Q_u(u, w) = \sum_{i=0}^n \sum_{j=0}^m B_{i,j} J'_{n,i}(u) K_{m,j}(w) \quad (6-62)$$

$$Q_w(u, w) = \sum_{i=0}^n \sum_{j=0}^m B_{i,j} J_{n,i}(u) K'_m(w) \quad (6-63)$$

$$Q_{u,w}(u, w) = \sum_{i=0}^n \sum_{j=0}^m B_{i,j} J'_{n,i}(u) K_{m,j}(w) \quad (6-64)$$

$$Q_{u,u}(u, w) = \sum_{i=0}^n \sum_{j=0}^m B_{i,j} J''_{n,i}(u) K_{m,j}(w) \quad (6-65)$$

$$Q_{w,w}(u, w) = \sum_{i=0}^n \sum_{j=0}^m B_{i,j} J_{n,i}(u) K''_m(w) \quad (6-66)$$

где штрих обозначает дифференцирование относительно параметрической переменной. Производные функций базиса Бернштейна $J'_{n,i}$, $J''_{n,i}$, $K'_{m,j}$ и $K''_{m,j}$ приведены в уравнениях (5-74) и (5-75).

Легко найти соотношение между бикубическими поверхностями Безье и Кунса. Приравнивая уравнения (6-52) и (6-59), получим

$$Q_{\text{Кунс}}(u, w) = Q_{\text{Безье}}(u, w),$$

$$[U][N_C][P][N_C]^T[W] = [U][N_B][B][N_B]^T[W]$$

где $[N_C]$ задан уравнением (5-76) и $[N_B]$ — уравнением (5-70). Следовательно геометрическая матрица $[P]$ бикубической поверхности Кунса задается в терминах полигональной сетки поверхности Безье следующим образом

$$[P] = [N_C]^{-1}[N_B][B][N_B]^T[[N_C]^T]^{-1}$$

или

$$\begin{bmatrix} P(0,0) & P(0,1) & P_w(0,0) & P_w(0,0) \\ P(1,0) & P(1,1) & P_w(1,0) & P_w(1,1) \\ P_u(0,0) & P_u(0,1) & P_{uw}(0,0) & P_{uw}(0,1) \\ P_u(1,0) & P_u(1,1) & P_{uw}(1,0) & P_{uw}(1,1) \end{bmatrix} = \begin{bmatrix} B_0 & B_{0,3} \\ B_{3,0} & B_{3,3} \\ 3(B_{1,0} - B_{0,0}) & 3(B_{1,3} - B_{0,3}) \\ 3(B_{3,0} - B_{2,0}) & 3(B_{3,3} - B_{2,3}) \end{bmatrix}$$

$$\begin{bmatrix} 3(B_{0,1} - B_{0,0}) & 3(B_{0,3} - B_{0,2}) \\ 3(B_{3,1} - B_{3,0}) & 3(B_{3,3} - B_{3,2}) \\ 9(B_{0,0} - B_{1,0} - B_{0,1} + B_{1,1}) & 9(B_{0,2} - B_{1,2} - B_{0,3} + B_{1,3}) \\ 9(B_{2,0} - B_{3,0} - B_{2,1} + B_{3,1}) & 9(B_{2,2} - B_{3,2} - B_{2,3} + B_{3,3}) \end{bmatrix} \quad (6-67)$$

Исследование правой нижней подматрицы размера 2×2 в уравнении (6-67) подтверждает, что четыре центральные вершины задающей полигональной сетки влияют на кручение в угловых точках куска бикубической поверхности Безье. Тем не менее кручение в угловых точках управляет не только центральными вершинами, но также и соседними касательными векторами. В самом деле, кручение в угловой точке управляется формой неплоского четырехугольника, сформированного угловой точкой, двумя соседними граничными точками и соседней центральной точкой.

Из уравнений (6-62)–(6-64) следует, что

$$\begin{bmatrix} P(0,0) & P(0,1) & P_w(0,0) & P_w(0,0) \\ P(1,0) & P(1,1) & P_w(1,0) & P_w(1,1) \\ P_u(0,0) & P_u(0,1) & P_{uw}(0,0) & P_{uw}(0,1) \\ P_u(1,0) & P_u(1,1) & P_{uw}(1,0) & P_{uw}(1,1) \end{bmatrix} =$$

$$\begin{bmatrix} Q(0,0) & Q(0,1) & Q_w(0,0) & Q_w(0,0) \\ Q(1,0) & Q(1,1) & Q_w(1,0) & Q_w(1,1) \\ Q_u(0,0) & Q_u(0,1) & Q_{uw}(0,0) & Q_{uw}(0,1) \\ Q_u(1,0) & Q_u(1,1) & Q_{uw}(1,0) & Q_{uw}(1,1) \end{bmatrix}. \quad (6-68)$$

Аналогичным образом обратное соотношение между матрицами $[P]$ и $[B]$, выражающее вершины полигональной сетки Безье в терминах параметров бикубической поверхности Кунса, равно

$$\begin{aligned} & \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{bmatrix} = \\ & \left(\frac{1}{3} \right) \begin{bmatrix} 3P(0,0) & 3P(0,0) + P_w(0,0) \\ 3P(0,0) + P_u(0,0) & \frac{1}{3} \{ P_{uw}(0,0) + 9P(0,0) - 3(P_w(0,0) + P_u(0,0)) \} \\ 3P(1,0) - P_u(1,0) & \frac{1}{3} \{ P_{uw}(1,0) + 9P(1,0) + 3(P_w(1,0) - P_u(1,0)) \} \\ 3P(1,0) & 3P(1,0) + P_w(1,0) \end{bmatrix} \\ & \begin{bmatrix} 3P(0,1) - P_w(0,1) & 3P(0,1) \\ \frac{1}{3} \{ P_{uw}(0,1) + 9P(0,1) + 3(P_u(0,1) - P_w(0,1)) \} & 3P(0,1) + P_u(0,1) \\ \frac{1}{3} \{ P_{uw}(1,1) + 9P(1,1) - 3(P_u(1,1) + P_w(1,1)) \} & 3P(1,1) - P_u(1,1) \\ 3P(1,1) - P_w(1,1) & 3P(1,1) \end{bmatrix}. \end{aligned} \quad (6-69)$$

Более полно концепция поверхности Безье иллюстрируется на следующем примере.

Пример 6-14 Поверхность Безье

Для изображенной на рис. 6-39а поверхности Безье для значений параметров $u = w = 0.5$ определить координаты точки на поверхности и первые производные в u и w направлениях. Найти также координаты точки и производные для модифицированной поверхности, показанной на рис. 6-39d. Сравнить полученные результаты. Вершины многогранника 4×4 поверхности Безье таковы:

$$[B] = \begin{bmatrix} [-15 \ 0 \ 15] & [-15 \ 5 \ 5] & [-15 \ 5 \ -5] & [-15 \ 0 \ -15] \\ [-5 \ 5 \ 15] & [-5 \ 5 \ 5] & [-5 \ 5 \ -5] & [-5 \ 5 \ -15] \\ [5 \ 5 \ 15] & [5 \ 5 \ 5] & [5 \ 5 \ -5] & [5 \ 5 \ -15] \\ [15 \ 0 \ 15] & [15 \ 5 \ 5] & [15 \ 5 \ -5] & [15 \ 0 \ -15] \end{bmatrix}$$

Для модифицированной поверхности на рис. 6-39d изменяется только вершина $B_{1,1}[0 \ 10 \ 0]$, т. е. изменение сказывается лишь на кручении в угловой точке $B_{0,0}$.

Напомним матричное представление поверхности, заданное в уравнениях (6-59) и (6-60), т. е.

$$Q(u, w) = [U][N][B][N]^T [W].$$

Здесь

$$\begin{aligned} [N][B][N]^T &= \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} [B] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} [0 \ 0 \ 0] & [0 \ 0 \ 0] & [0 \ 0 \ 0] & [0 \ 0 \ 0] \\ [0 \ 0 \ 0] & [0 \ -45 \ 0] & [0 \ 45 \ 0] & [0 \ -15 \ 0] \\ [0 \ 0 \ 0] & [0 \ 45 \ 0] & [0 \ -45 \ 0] & [30 \ 15 \ 0] \\ [0 \ 0 \ 0] & [0 \ -15 \ 0] & [0 \ 15 \ -30] & [-15 \ 0 \ 15] \end{bmatrix} \end{aligned}$$

Точка на поверхности, таким образом, будет

$$Q(0.5, 0.5) = [0.125 \quad 0.25 \quad 0.5 \quad 1][N][B][N]^T \begin{bmatrix} 0.125 \\ 0.25 \\ 0.5 \\ 1 \end{bmatrix}$$

$$= [0 \quad 4.6875 \quad 0].$$

Производные по направлениям:

$$Q_u(0.5, 0.5) = [3u^2 \quad 2u \quad 1 \quad 0][N][B][N]^T \begin{bmatrix} w^3 \\ w^2 \\ w \\ 1 \end{bmatrix}$$

$$= [0.75 \quad 1 \quad 1 \quad 0][N][B][N]^T \begin{bmatrix} 0.125 \\ 0.25 \\ 0.5 \\ 1 \end{bmatrix}$$

$$= [30 \quad 0 \quad 0]$$

$$Q_w(0.5, 0.5) = [u^3 \quad u^2 \quad u \quad 1][N][B][N]^T \begin{bmatrix} 3w^2 \\ 2w \\ 1 \\ 0 \end{bmatrix}$$

$$= [0.125 \quad 0.25 \quad 0.5 \quad 1][N][B][N]^T \begin{bmatrix} 0.75 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$$= [0 \quad 0 \quad -30]$$

Заметим, что векторы Q_u и Q_w ортогональны.

В модифицированной поверхности на рис. 6-39d изменено значение только точки $B_{1,1}$ стандартной поверхности. Новое значение координат равно $B_{1,1}[0 \quad 10 \quad 0]$. Новое значение произведения:

$$[N][B][N]^T =$$

$$\begin{bmatrix} [45 \quad 45 \quad -45] & [-90 \quad -90 \quad 90] & [45 \quad 45 \quad -45] & [0 \quad 0 \quad 0] \\ [-90 \quad -90 \quad 90] & [180 \quad 135 \quad -180] & [-90 \quad -45 \quad 90] & [0 \quad -15 \quad 0] \\ [45 \quad 45 \quad -45] & [-90 \quad -45 \quad 90] & [45 \quad 0 \quad -45] & [30 \quad 15 \quad 0] \\ [0 \quad 0 \quad 0] & [-0 \quad -15 \quad 0] & [0 \quad -15 \quad -30] & [-15 \quad 0 \quad 15] \end{bmatrix}$$

Новые координаты точки поверхности для $u = w = 0.5$:

$$Q(0.5, 0.5) = [0.703 \quad 5.391 \quad -0.703].$$

Новые параметрические производные для $u = w = 0.5$:

$$Q_u(0.5, 0.5) = [28.594 \quad -1.406 \quad 1.406]$$

$$Q_w(0.5, 0.5) = [-1.406 \quad -1.406 \quad -28.594]$$

Заметим в этом случае, что поскольку задающая полигональная сетка более не симметрична относительно оси y , то поверхность теперь не симметрична относительно этой оси. Далее, заметим, что хотя Q_u и Q_w все еще ортогональны, но как их величины, так и направления отличаются. Данные результаты показывают, что вектор кручения в одной угловой точке оказывает трудно уловимое, но существенное влияние на форму всей поверхности.

Приведенное выше обсуждение поверхностей Безье касалось определения и характеристик одного куска поверхности. Для того чтобы получить более сложные поверхности, надо объединить несколько кусков поверхности Безье. Подробное обсуждение этого вопроса лежит вне сферы данной книги. Интересующегося читателя мы отсылаем к [6-24] и [6-25]. Проблемы, возникающие при объединении кусков поверхности Безье с обеспечением гладкости вдоль соприкасающихся сторон, иллюстрируются рис. 6-41 на примере объединения двух кусков бикубической поверхности Безье вдоль одной стороны.

Для обеспечения непрерывности или C^0 гладкости вдоль границы необходимо, чтобы совпадали¹ две граничные кривые, а следовательно, и две граничные ломаные вдоль края поверхности. Для обеспечения непрерывности векторов наклона или касательных векторов или C^1 гладкости вдоль границы куска направление нормали к поверхности вдоль граничной кривой должно быть одинаковым для обоих кусков. Для этого можно использовать два условия. Первое требует, чтобы четыре отрезка полигональной сетки, встречающиеся у границы и пересекающие ее, были коллинеарными, как это показано выделенными линиями на рис. 6-41a. Второе, менее жесткое условие требует, чтобы только три ребра полигональной сетки, встречающиеся в концевых точках граничной кривой, были компланарными, как это показано выделенными линиями на рис. 6-41b.

6-12 В-СПЛАЙН ПОВЕРХНОСТИ

Естественным расширением понятия поверхности Безье является декартово произведение В-сплайна поверхности, определяемой выражением

$$Q(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M_{j,l}(w), \quad (6-70)$$

где $N_{i,k}(u)$ и $M_{j,l}(w)$ — базисные функции В-сплайна в бипараметрических направлениях u и w соответственно (см. уравнение 5-84). Для удобства повторим здесь определение базисных функций, данное ранее в разд. 5-9,

$$N_{i,1}(u) = \begin{cases} 1, & \text{если } x_i \leq u < x_{i+1}, \\ 0, & \text{в противном случае,} \end{cases} \quad (5-84a)$$

$$N_{i,k}(u) = \frac{(u - x_i) N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u) N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}} \quad (5-84b)$$

¹ Вспомним обсуждение в разд. 5-8 гладкости сегментов кривой Безье.

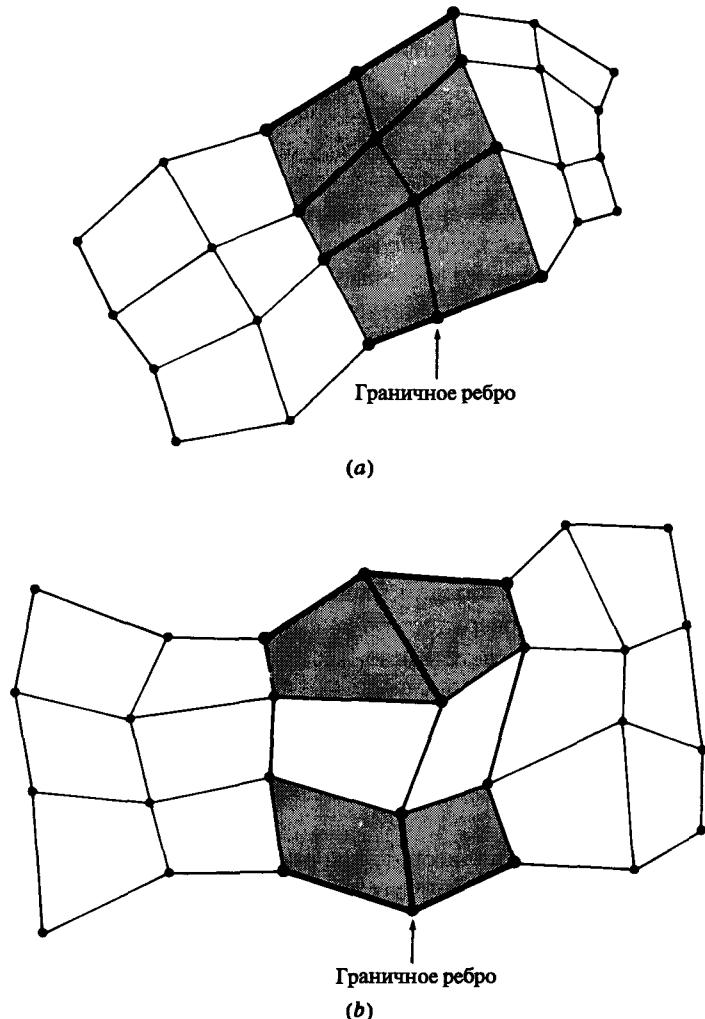


Рис. 6-41 Гладкость стыковки кусков поверхности Безье. (a) Линии сетки коллинеарны; (b) ребра ломаных компланарны.

и

$$M_{j,l}(w) = \begin{cases} 1, & \text{если } y_j \leq w < y_{j+1}, \\ 0, & \text{в противном случае,} \end{cases} \quad (5-84a)$$

$$M_{j,l}(w) = \frac{(w - y_i)M_{j,l-1}(w)}{y_{j+l-1} - y_j} + \frac{(y_{j+l} - w)M_{j+1,l-1}(w)}{y_{j+l} - y_{j+1}}, \quad (5-84b)$$

где x_i и y_j являются элементами узловых векторов, как это обсуждалось в разд. 5-9. Снова $B_{i,j}$ являются вершинами задающей полигональной сетки. Для четырехугольных «кусков» поверхности задающая полигональная сетка должна

быть топологически прямоугольной. Индексы n и m на единицу меньше числа вершин задающего многогранника соответственно в u и w параметрических направлениях.

Как и для B-сплайн кривых на форму и свойства B-сплайн поверхности существенно влияют узловые векторы $[X]$ и $[Y]$, причем используются незамкнутые, периодические и неоднородные узловые векторы. Хотя обычно для обоих параметрических направлений применяют узловые векторы одного и того же типа, но это не обязательно. Например, можно использовать незамкнутый узловой вектор и его соответствующие B-сплайн базисные функции для одного параметрического направления и периодический узловой вектор и его соответствующие B-сплайн базисные функции для другого. Практическим примером служит цилиндрическая поверхность с переменным сечением.

Так как для описания граничных кривых и для интерполяирования внутренней части поверхности используется B-сплайн базис, то сразу же можно перечислить некоторые свойства B-сплайн поверхности:

Максимальный порядок поверхности в каждом параметрическом направлении равен числу вершин задающего многогранника в этом направлении.

Гладкость поверхности в каждом параметрическом направлении на две единицы меньше порядка поверхности в каждом направлении; т. е. C^{k-2} и C^{l-2} в u и w направлениях, соответственно.

Поверхность инвариантна относительно аффинного преобразования; т. е. поверхность преобразуется с помощью преобразования задающей полигональной сетки.

Свойство затухания изменений для B-сплайн поверхности в настоящее время неизвестно.

Влияние одной вершины полигональной сетки ограничивается $\pm k/2$, $\pm l/2$ интервалами в каждом параметрическом направлении.

Если число вершин задающей полигональной сетки равно порядку в каждом параметрическом направлении и внутренних узловых величин нет, то B-сплайн поверхность превращается в поверхность Безье (см. рис. 6-39).

При триангуляции задающая полигональная сетка образует плоскую аппроксимацию поверхности.

Поверхность лежит внутри выпуклой оболочки задающей полигональной сетки, образуемой объединением всех выпуклых оболочек k , l соседних вершин полигональной сетки.

Из предыдущего обсуждения свойств выпуклой оболочки B-сплайн кривых (см. разд. 5-9) сразу же следует, что B-сплайн поверхность может содержать плоские области и линии резкого нарушения гладкости. Это свойство очень полезно во многих ситуациях, возникающих при конструировании. На рис. 6-42 $a - d$ изображена серия незамкнутых B-сплайн поверхностей и их характеристических многогранников третьего порядка в каждом характеристическом направлении. Отметим, что каждая из линий задающей полигональной сетки в направлении w является прямой линией с четырьмя вершинами. Получающаяся поверхность

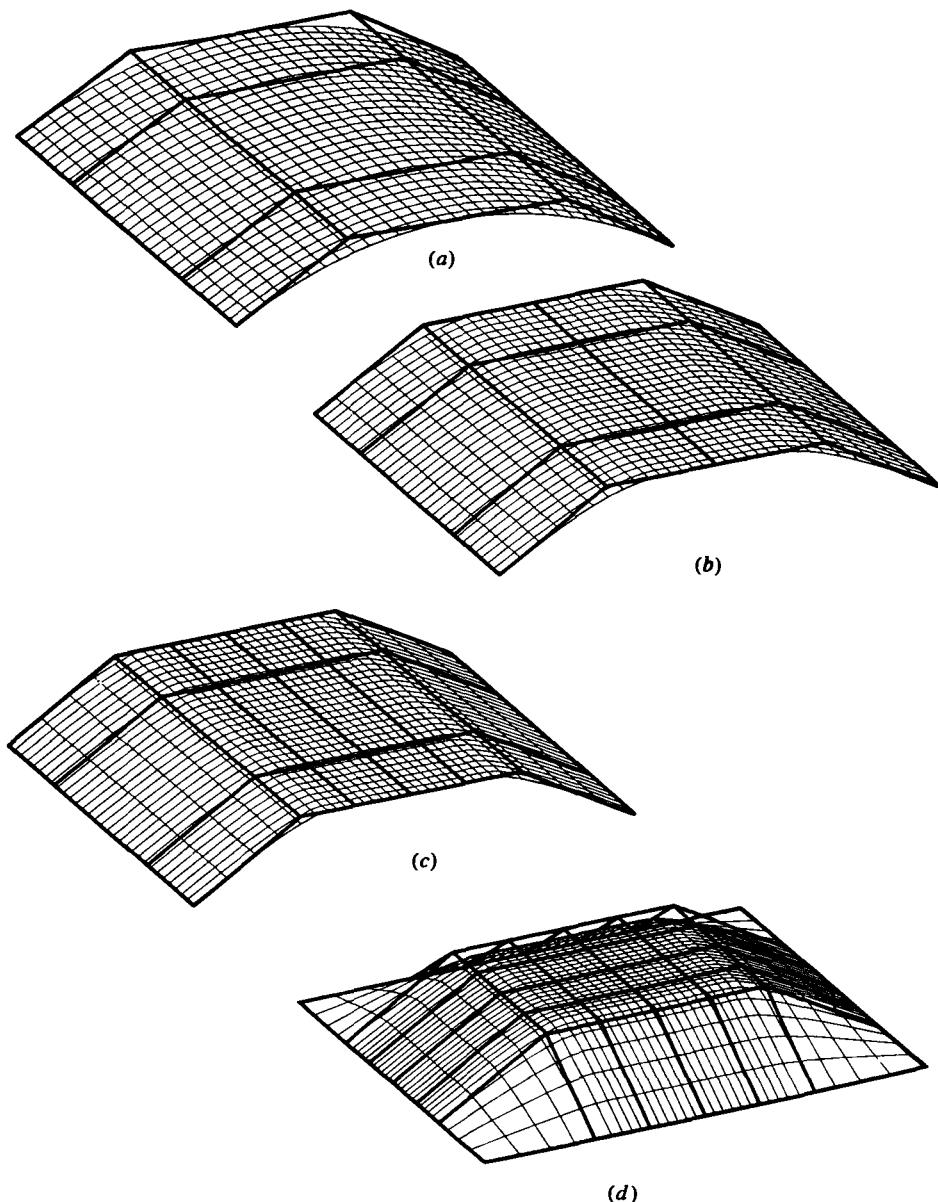


Рис. 6-42 В-сплайн поверхности третьего порядка. (а) Гладкая линейчатая поверхность; (б) небольшая внутренняя плоская область, вызванная коллинеарностью трех вершин сетки в u направлении; (с) обширная внутренняя плоская область, вызванная коллинеарность пяти вершин сетки в u направлении; (д) плоская область внутри скульптурной поверхности.

линейчата в направлении w . Изображенная на рис. 6-42a В-сплайн поверхность, заданная четырьмя вершинами полигональной сетки в направлении u , плавно изогнута в этом направлении.

Изображенная на рис. 6-42b В-сплайн поверхность задана пятью вершинами полигональной сетки в направлении u . Три центральные вершины коллинеарны. Заметим, что центр получившейся поверхности имеет плоскую форму. Аналогичным образом коллинеарны пять из семи вершин задающей полигональной сетки в направлении u для поверхности, изображенной на рис. 6-42c. И снова в центральной области, которая имеет большие размеры, чем на рис. 6-42b, поверхность плоская.

На рис. 6-42d показано, что эти очень сильные свойства выпуклой оболочки распространяются на оба параметрических направления. Таким образом, плоская область может быть встроена во внутреннюю часть скульптурной поверхности. При увеличении порядка поверхности плоская область становится меньше.

На рис. 6-43 иллюстрируется эффект, возникающий при совпадении линий сетки. На рис. 6-43a три совпадающие линии сетки используются для образования линии складки в центре В-сплайн поверхности четвертого порядка. На рис. 6-43b показан результат совмещения трех линий сетки в обоих параметрических направлениях. В этом случае В-сплайн поверхность четвертого порядка содержит два гребня, поднимающихся к точке в центре поверхности. Так же как и для В-сплайн кривых, линия складки возникает в том случае, когда совпадают $k - 1$ или $l - 1$ линий сетки. Кроме того, так как В-сплайн поверхность везде C^{k-2}/C^{l-2} гладка, то она гладка и на этой линии. Вдобавок данное свойство также гарантирует C^{k-2}/C^{l-2} гладкость перехода изогнутой поверхности в плоскую.

Превосходные свойства локального изменения В-сплайн кривых (см. разд. 5-9) переносятся на В-сплайн поверхности. Пример этого приведен на рис. 6-44, где незамкнутая бикубическая ($k = l = 4$) В-сплайн поверхность определена полигональной сеткой 9×9 ($m = n = 8$). Эта сетка, показанная как верхняя поверхность на рис. 6-44, плоская везде, за исключением центральной точки. Незамкнутый узловой вектор в обоих параметрических направлениях равен $[0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 6 \ 6 \ 6]$. Таким образом, у нас есть шесть промежутков параметров в каждом направлении, т. е. $0 - 1, 1 - 2, \dots, 5 - 6$. Каждый параметрический четырехугольник, например $0 \leq u \leq 1, 0 \leq w \leq 1$, образует подкусок¹ В-сплайн поверхности. Средняя поверхность, изображенная на рис. break 6-44, составлена из параметрических линий на концах каждого параметрического интервала, т. е. в $u = w = 0, 1, 2, 3, 4, 5, 6$. Каждый четырехугольник представляет подкусок поверхности. Отметим, что влияние смещенной точки ограничивается $\pm k/2, l/2$ интервалами или подкусками.

¹ Некоторые авторы обозначают каждый подкусок как В-сплайн поверхность. В данном случае поверхность рассматривается как объект, заданный всей полигональной сеткой.

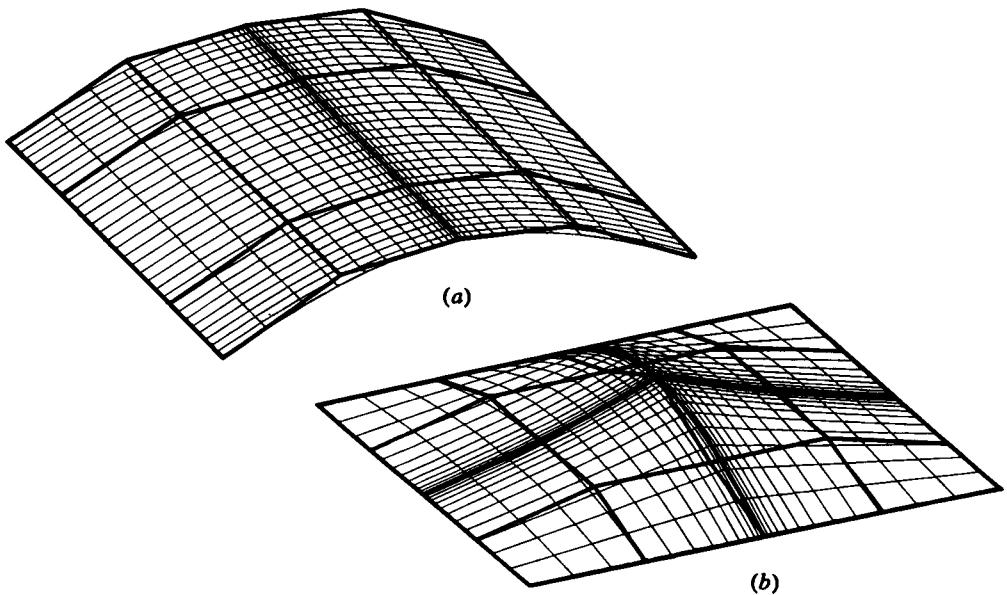


Рис. 6-43 В-сплайн поверхности четвертого порядка с несколькими совпадающими линиями сетки.

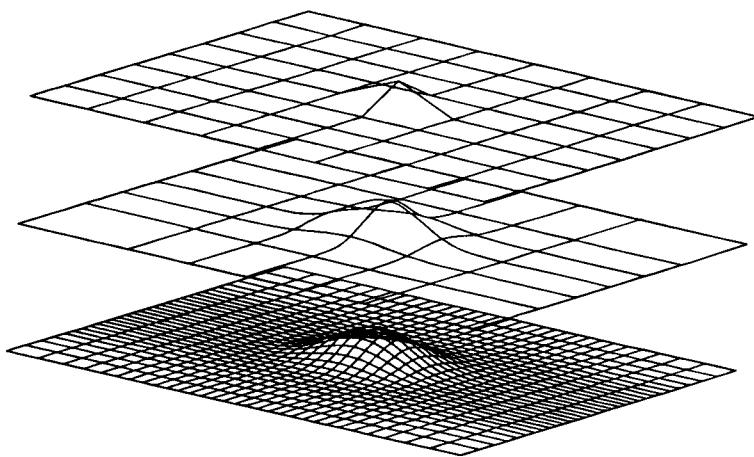


Рис. 6-44 Локальное изменение В-сплайн поверхностей.

Параметрические производные В-сплайн поверхности получаются с помощью формального дифференцирования уравнения (6-70):

$$Q_u(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N'_{i,k}(u) M_{j,l}(w), \quad (6-71)$$

$$Q_w(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M'_j(w), \quad (6-72)$$

$$Q_{uw}(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N'_{i,k}(u) M'_{j,l}(w), \quad (6-73)$$

$$Q_{uu}(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N''_{i,k}(u) M_{j,l}(w), \quad (6-74)$$

$$Q_{ww}(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} N_{i,k}(u) M''_{j,l}(w), \quad (6-75)$$

где штрих обозначает дифференцирование относительно соответствующего параметра. Производные В-сплайн базисных функций задаются уравнениями (5-97)–(5-100).

Приведенный пример иллюстрирует метод вычисления В-сплайн поверхности.

Пример 6-15 Вычисление незамкнутой В-сплайн поверхности

Рассмотрим В-сплайн поверхность, заданную полигональной сеткой размера 4×4 :

$$\begin{array}{llll} B_{1,1}[-15 & 0 & 15] & B_{2,1}[-5 & 5 & -15] & B_{3,1}[5 & 5 & 15] & B_{4,1}[15 & 0 & 15] \\ B_{1,2}[-15 & 5 & 5] & B_{2,2}[-5 & 10 & 5] & B_{3,2}[5 & 10 & 5] & B_{4,2}[15 & 5 & 5] \\ B_{1,3}[-15 & 5 & -5] & B_{2,3}[-5 & 10 & -5] & B_{3,3}[5 & 10 & -5] & B_{4,3}[15 & 5 & -5] \\ B_{1,4}[-15 & 0 & -15] & B_{2,4}[-5 & 5 & -15] & B_{3,4}[5 & 5 & -15] & B_{4,4}[15 & 0 & -15] \end{array}$$

Это поверхность четвертого порядка в направлении u ($k = 4$) и третьего порядка в направлении w ($l = 3$). Таким образом, поверхность составлена из двух подкусков: один для $0 \leq u \leq 1$, $0 \leq w \leq 1$ и другой для $0 \leq u \leq 1$, $1 \leq w \leq 2$. Надо найти точку в центре поверхности, т. е. при $u = 1/2$, $w = 1$.

Расписывая уравнение (6-70), получим

$$\begin{aligned} Q(u, w) &= \sum_{i=1}^4 \sum_{j=1}^4 B_{i,j} N_{i,4}(u) M_{j,3}(w) \\ &= N_{1,4}(B_{1,1}M_{1,3} + B_{1,2}M_{2,3} + B_{1,3}M_{3,3} + B_{1,4}M_{4,3}) \\ &\quad + N_{2,4}(B_{2,1}M_{1,3} + B_{2,2}M_{2,3} + B_{2,3}M_{3,3} + B_{2,4}M_{4,3}) \\ &\quad + N_{3,4}(B_{3,1}M_{1,3} + B_{3,2}M_{2,3} + B_{3,3}M_{3,3} + B_{3,4}M_{4,3}) \\ &\quad + N_{4,4}(B_{4,1}M_{1,3} + B_{4,2}M_{2,3} + B_{4,3}M_{3,3} + B_{4,4}M_{4,3}). \end{aligned}$$

Здесь узловой вектор в u направлении $[X] = [0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1]$. Вспомнив пример (5-12), получим выражения для базисных функций, т. е.

$$\begin{aligned} N_{1,4}\left(\frac{1}{2}\right) &= (1-u)^3 = \left(\frac{1}{2}\right)^3 = \frac{1}{8}, \\ N_{2,4}\left(\frac{1}{2}\right) &= 3u(1-u)^2 = (3)\left(\frac{1}{2}\right)\left(\frac{1}{2}\right)^2 = \frac{3}{8}, \\ N_{3,4}\left(\frac{1}{2}\right) &= 3u^2(1-u) = (3)\left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right) = \frac{3}{8}, \\ N_{4,4}\left(\frac{1}{2}\right) &= u^3 = \left(\frac{1}{2}\right)^3 = \frac{1}{8}. \end{aligned}$$

Аналогичным образом узловой вектор в направлении w имеет вид $[Y] = [0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2]$. Вспомнив пример (5-10), получим выражения для базисных функций, т. е.

$$\begin{aligned} M_{1,3}(1) &= 0, \\ M_{2,3}(1) &= \frac{(2-w)^2}{2} = \frac{(2-1)^2}{2} = \frac{1}{2}, \\ M_{3,3}(1) &= \frac{(2-w)(3w-2)}{2} = \frac{(2-1)(3-2)}{2} = \frac{1}{2}, \\ M_{4,3}(1) &= (w-1)^2 = (1-1) = 0. \end{aligned}$$

Таким образом,

$$\begin{aligned} Q(1/2, 1) &= \frac{1}{8}\{(0)B_{1,1} + \frac{1}{2}B_{1,2} + \frac{1}{2}B_{1,3} + (0)B_{1,4}\} \\ &\quad + \frac{3}{8}\{(0)B_{2,1} + \frac{1}{2}B_{2,2} + \frac{1}{2}B_{2,3} + (0)B_{2,4}\} \\ &\quad + \frac{3}{8}\{(0)B_{3,1} + \frac{1}{2}B_{3,2} + \frac{1}{2}B_{3,3} + (0)B_{3,4}\} \\ &\quad + \frac{1}{8}\{(0)B_{4,1} + \frac{1}{2}B_{4,2} + \frac{1}{2}B_{4,3} + (0)B_{4,4}\} \\ Q(1/2, 1) &= \frac{1}{16}(B_{1,2} + B_{1,3}) + \frac{3}{16}(B_{2,2} + B_{2,3}) \\ &\quad + \frac{3}{16}(B_{3,2} + B_{3,3}) + \frac{1}{16}(B_{4,2} + B_{4,3}) \\ &= \frac{1}{16}\{[-15 \ 5 \ 5] + [-15 \ 5 \ -5]\} \\ &\quad + \frac{3}{16}\{[-5 \ 10 \ 5] + [-5 \ 10 \ -5]\} \\ &\quad + \frac{3}{16}\{[5 \ 10 \ 5] + [5 \ 10 \ -5]\} \\ &\quad + \frac{1}{16}[15 \ 5 \ 5] + [15 \ 5 \ -5]\} \\ &= [0 \ 35/4 \ 0] \end{aligned}$$

Периодические B-сплайн поверхности легко генерируются с помощью периодических базисных функций в уравнении (6-70), для получения которых используются периодические узловые векторы. На рис. 6-45 показано несколько приме-

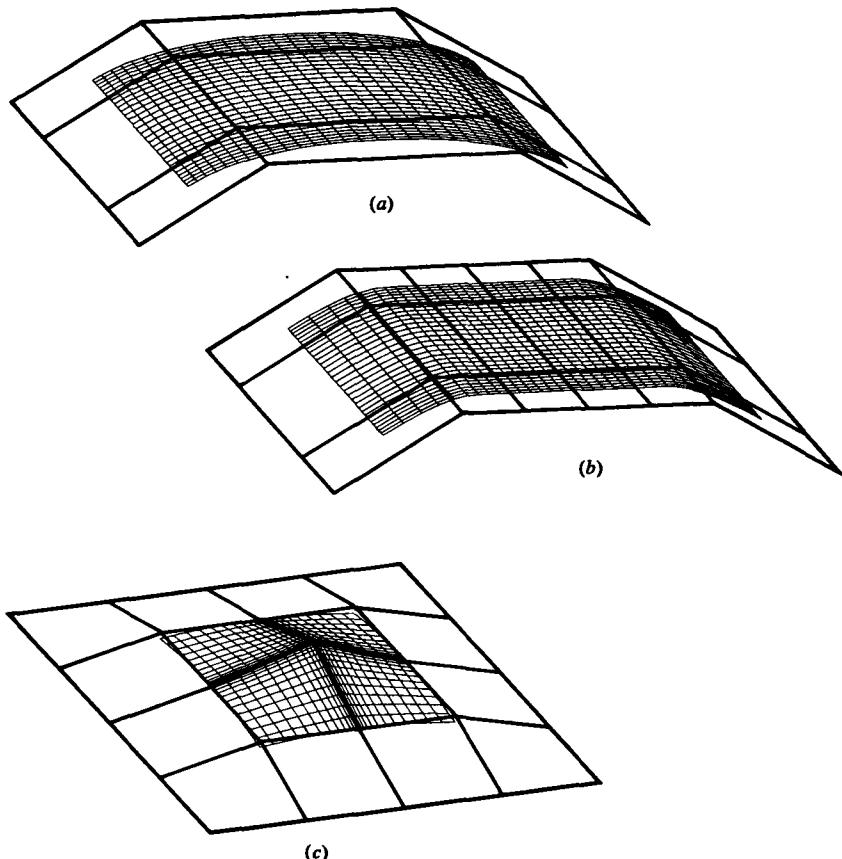


Рис. 6-45 Периодические В-сплайн поверхности для незамкнутых характеристических многогранников. (а) Гладкая линейчатая поверхность третьего порядка; (б) большая внутренняя плоская область третьего порядка, вызванная коллинеарностью пяти вершин сетки в *u* и направлении; (с) острый выступ на поверхности четвертого порядка, вызванный пересечением нескольких совпадающих линий сетки.

ров периодических В-сплайн поверхностей, формируемых незамкнутыми задающими полигональными сетками. На рисунках 6-45*a* и *b* задающие полигональные сетки соответствуют сеткам рисунков 6-42*a* и *c*, а сетка рис. 6-45*c* соответствует сетке рис. 6-43*b*. Заметим, что во всех случаях, так же как и для периодических В-сплайн кривых, границы поверхности и многогранника не совпадают из-за уменьшения диапазона параметра, используемого для периодических В-сплайн базисных функций.

Замкнутые периодические В-сплайн поверхности демонстрируют свойства, аналогичные свойствам замкнутых периодических В-сплайн кривых. На рис. 6-46 показаны примеры трех поверхностей третьего порядка. Задающая полигональная сетка на рис. 6-46*a* образуется с помощью повторения через одинаковые промежутки вдоль оси *z* от точки *z* = -8 до *z* = 8 задающего многоугольника для

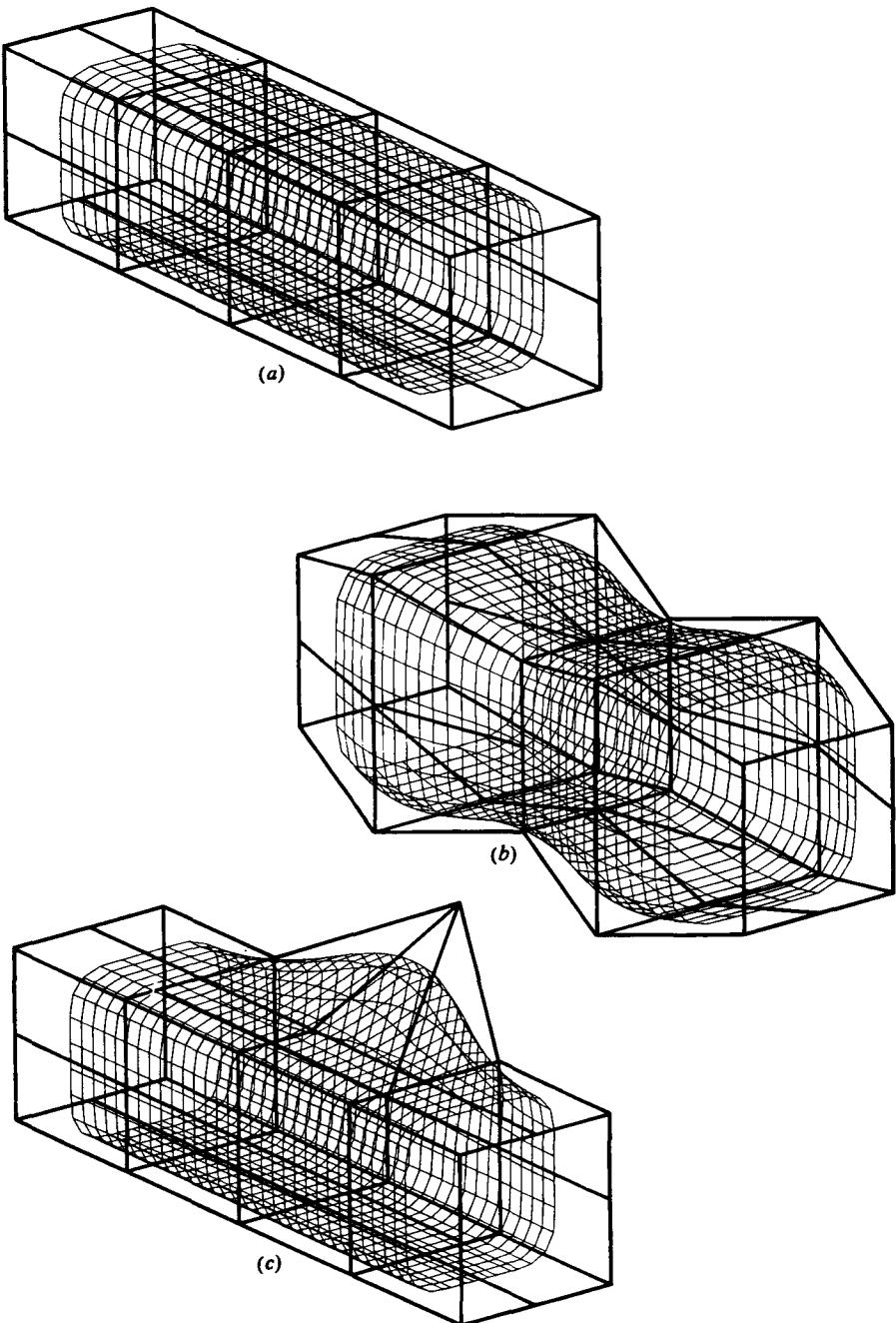


Рис. 6-46 Замкнутые периодические В-сплайн поверхности. (а) Прямая цилиндрическая поверхность; (б) возмущенная волнистая цилиндрическая поверхность; (с) эффект возмущения одной вершины сетки.

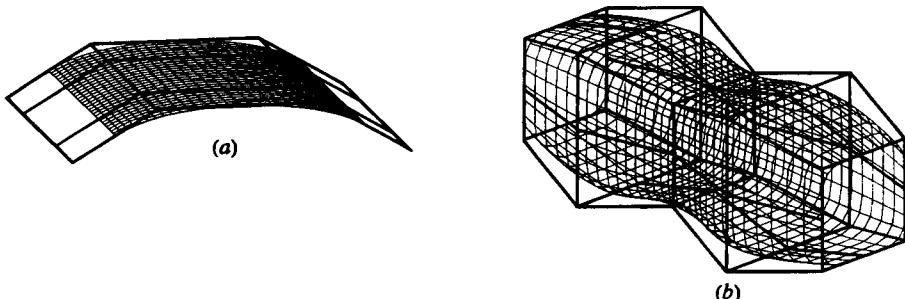


Рис. 6-47 Объединенные В-сплайн поверхности третьего порядка. (a) Незамкнутый характеристический многогранник; (b) замкнутый характеристический многогранник.

замкнутой В-сплайн кривой на рис. 6-46b. В результате получается цилиндрическая поверхность. Отметим, что поверхность не касается плоскостей первого и последнего задающих многоугольников. Характеристический многогранник для рис. 6-46b получен с помощью увеличения на единицу x - и y -размеров второго и четвертого задающих многоугольников на рис. 6-46a. В результате получается волнистый цилиндр. На рис. 6-46c показан локальный эффект возмущения одной вершины задающей сетки.

В уравнении (6-70) можно комбинировать незамкнутые и периодические В-сплайн базисные функции. Два примера этого представлены на рис. 6-47. Здесь в одном параметрическом направлении используются незамкнутый узловый вектор и базисная функция, а в другом направлении используются периодический узловый вектор и базисная функция. На рис. 6-47a показана комбинированная В-сплайн поверхность, заданная незамкнутой полигональной сеткой с рис. 6-42a. На рис. 6-47b показана комбинированная В-сплайн поверхность, заданная замкнутой полигональной сеткой с рис. 6-46b. Отметим, что поверхность совпадает с крайними линиями полигональной сетки в направлении u . Это свойство бывает полезным в некоторых случаях.

Матричное выражение для периодических В-сплайн поверхностей имеет вид

$$Q(s, t) = [U^*][N^*][B_{s,t}^*][M^*]^T[W^*]^T \quad (6-76)$$

где $[U^*]$ и $[W^*]$ являются репараметризованными параметрическими переменными в интервалах $0 \leq u^* \leq 1$ и $0 \leq w^* \leq 1$, заданными в уравнении (5-90). $[N^*]$ и $[M^*]$ задаются уравнением (5-91). Матрица $[B_{s,t}^*]$ представляет скользящую сетку $k \times l$ вершин характеристического многогранника, задающего подкусок на поверхности. Для периодических В-сплайн поверхностей, заданных незамкнутыми полигональными сетками

$$[B_{s,t}^*] = [B_{i,j}], \quad (6-77)$$

где

$$\begin{aligned} 1 \leq s &\leq n - k + 2, & s \leq i &\leq s + k - 1, \\ 1 \leq t &\leq m - l + 2, & t \leq j &\leq t + l - 1 \end{aligned} \quad (6-78)$$

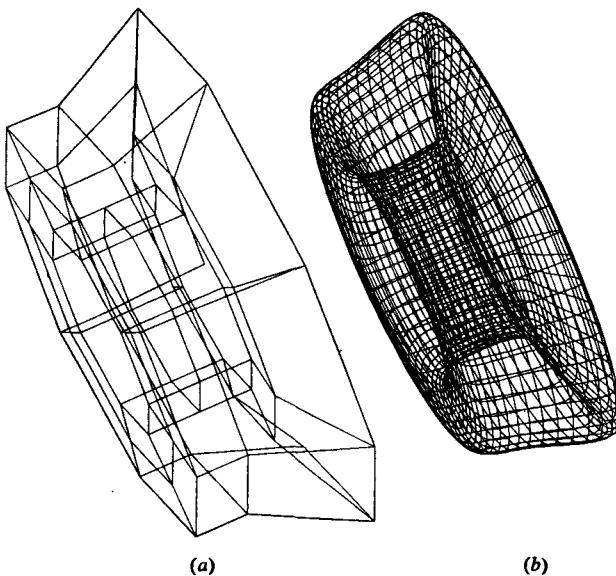


Рис. 6-48 Замкнутая тороидальная бикубическая ($k = l = 4$) В-сплайн поверхность.
(a) Характеристический многогранник; (b) поверхность.

и $B_{i,j}$ представляет индивидуальные элементы задающей полигональной сетки.

Для полигональных сеток, замкнутых вдоль $u = 0$, т. е. с совпадающими первой и последней сеточными линиями в направлении u , скользящая сетка задается так¹

$$\begin{aligned} 1 &\leq s \leq n - k + 2, \\ 1 &\leq t \leq m + l, \\ s &\leq i \leq s + k - 1, \\ j &\in [\{(t-1) \bmod (m+1)\} + 1 : (t+l-2) \bmod (m+1) + 1]. \end{aligned} \quad (6-79)$$

Аналогичным образом, для полигональных сеток, замкнутых вдоль $w = 0$, скользящая сетка задается в виде

$$\begin{aligned} 1 &\leq s \leq n + 1, \\ 1 &\leq t \leq m - l + 2, \\ i &\in [\{(s-1) \bmod (n+1)\} + 1 : (s+k-2) \bmod (n+1) + 1], \\ t &\leq j \leq t + l - 1. \end{aligned} \quad (6-80)$$

¹ $j \in [a : b]$ означает принадлежность множеству $a \dots b$. В данном случае списки вершин многогранника должны рассматриваться как заключенные. Например, если $m+1=4$, то $j \in [3 : 2]$ означает принадлежность множеству 3, 4, 1, 2 в указанном порядке.

И наконец, для полигональных сеток, замкнутых вдоль как $u = 0$, так и $w = 0$, скользящая сетка задается так:

$$\begin{aligned} 1 &\leq s \leq n - k + 2, \\ 1 &\leq t \leq m - l + 2, \\ i &\in [\{(s-1) \bmod (n+1)\} + 1 : (s+k-2) \bmod (n+1) + 1], \\ j &\in [\{(t-1) \bmod (m+1)\} + 1 : (t+l-2) \bmod (m+1) + 1]. \end{aligned} \quad (6-81)$$

В этом случае образуется полностью замкнутая поверхность. Пример изображен на рис. 6-48. Задающая полигональная сетка, изображенная на рис. 6-48a, формируется с помощью переноса вершин задающего многоугольника для периодической В-сплайн кривой на рис. 6-47 на -2 единицы по x и $+4$ единицы по y и затем вращения на 360° вокруг оси x с шагом 45° . Замкнутая периодическая бикубическая ($k = l = 4$) В-сплайн поверхность, изображенная на рис. 6-48b, имеет форму тора.

Матричное выражение для незамкнутых В-сплайн поверхностей имеет ту же форму, что и уравнение (6-76). Однако, так же как и для матричного выражения незамкнутых В-сплайн кривых, существование нескольких узловых значений на концах узлового вектора делает этот результат менее компактным и менее полезным, чем для периодических В-сплайн поверхностей. По этим причинам данный вопрос не рассматривается здесь более подробно.

6-13 В-СПЛАЙН ИНТЕРПОЛЯЦИЯ

В предыдущих разделах обсуждались свойства и создание В-сплайн поверхностей по известной задающей полигональной сетке. Также представляет интерес и обратная задача: т. е. задано известное множество данных на поверхности, надо найти задающую полигональную сетку для В-сплайн поверхности, наилучшим образом интерполирующей эти данные. Так как границы поверхности, представляемой этими данными, обычно известны, то здесь будут рассматриваться только незамкнутые В-сплайн поверхности. Разработка аналогичного метода для замкнутых поверхностей с помощью периодических В-сплайн поверхностей не представляет труда. Обсуждение ограничивается топологически прямоугольными сетками, т. е. данные умозрительно могут быть представлены как находящиеся в точках пересечения прямоугольной решетки. Эта задача схематично изображена на рис. 6-49, где показана задающая полигональная сетка размера 4×4 для матрицы данных 8×8 . Эта задача рассматривалась в [6-5] и в [6-26]. Для того чтобы повысить эффективность вычислений, в работе [6-26] были использованы известные свойства бикубических В-сплайн поверхностей. Применяемый нами метод более прямолинеен, но имеет меньшую вычислительную эффективность (см. разд. 5-11).

Вспомним уравнение (6-70) и заметим, что $Q(u, w)$ в этом уравнении являются известными точками на поверхности. Базисные функции $N_{i,k}(u)$ и $M_{j,l}(w)$ могут быть определены для известного порядка и известного числа вершин задающей полигональной сетки в каждом параметрическом направлении при условии, что

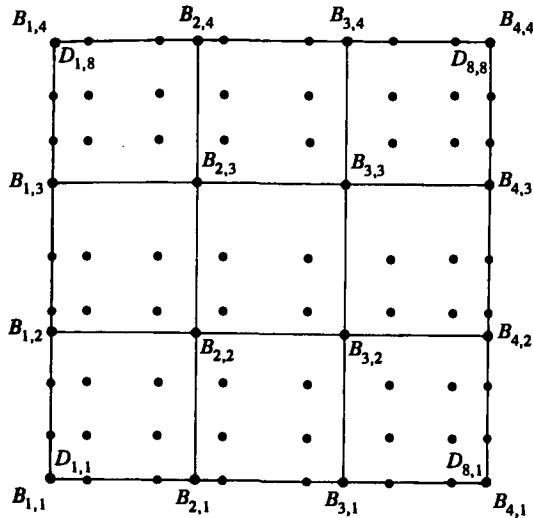


Рис. 6-49 Определение В-сплайнов поверхности по известному набору данных.

известны значения параметров u, w в исходных точках. Следовательно, для каждой известной точки поверхности уравнение (6-70) дает линейное уравнение от неизвестных вершин B_{ij} задающей полигональной сетки. В результате расписывания уравнения (6-70) для исходной точки поверхности получим

$$\begin{aligned} D_{1,1}(u_1, w_1) = & \\ N_{1,k}(u_1)[M_{1,l}(w_1)B_{1,1} + M_{2,l}(w_1)B_{1,2} + \cdots + M_{m+1,l}(w_1)B_{1,m+1}] + & \\ \vdots & \\ N_{n+1,k}(u_1)[M_{1,l}(w_1)B_{n+1,1} + M_{2,l}(w_1)B_{n+1,2} + \cdots + M_{m+1,l}(w_1)B_{n+1,m+1}] & \end{aligned}$$

где для топологически прямоугольного множества $r \times s$ данных выполняются неравенства $2 \leq k \leq n+1 \leq r$ и $2 \leq l \leq m+1 \leq s$. Записав уравнения такого вида для всех исходных точек, мы получим систему уравнений. В матричном виде данный результат записывается так:

$$[D] = [C][B], \quad (6-82)$$

где $C_{i,j} = N_{i,k} M_{j,l}$. Для топологически прямоугольных исходных данных размера $r \times s$ матрица $[D]$ является матрицей $r \times s \times 3$, содержащей трехмерные координаты исходных точек, $[C]$ является матрицей $r \times s \times n \times m$ трехмерных координат искомых точек полигональной сетки.

Если матрица $[C]$ квадратная, то задающая полигональная сетка находится с помощью обращения матрицы, т. е.

$$[B] = [C]^{-1}[D]. \quad (6-83)$$

В этом случае получившаяся поверхность проходит через все исходные точки. Хотя эта поверхность будет везде C^{k-2}, C^{l-2} гладкая, но она может быть непри-

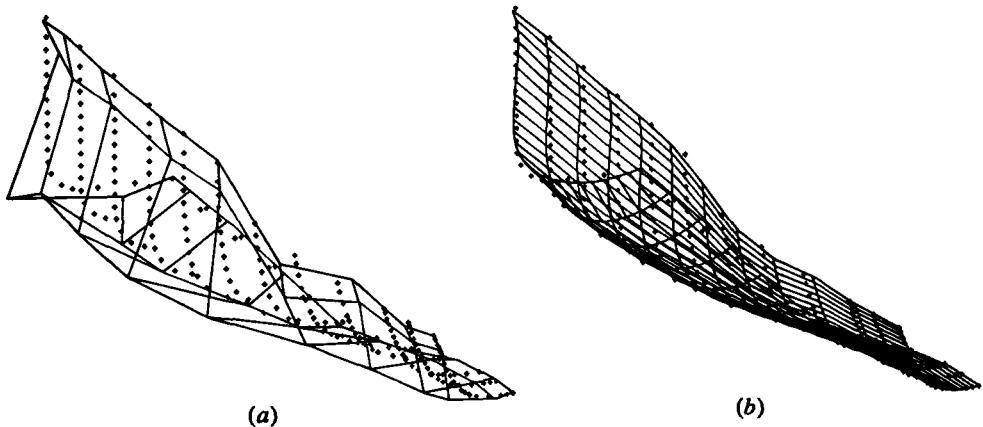


Рис. 6-50 Интерполяция В-сплайн поверхностью. (a) Исходные точки и интерполирующий характеристический многогранник; (b) исходные точки и сгенерированная поверхность.

емлема с точки зрения пользователя. Опыт показывает, что в общем случае чем меньше точек задающей полигональной сетки, тем лучше выглядит поверхность.

Если $[C]$ не квадратная, то мы имеем избыточное количество исходных данных и решение может быть получено в некотором усредненном смысле. В частности, решение может быть получено в виде

$$[B] = [[C]^T [C]]^{-1} [C]^T [D]. \quad (6-84)$$

Параметрические значения u и w для каждой исходной точки поверхности получаются с помощью аппроксимации длины хорды (см. разд. 5-11). Конкретнее, для r исходных точек значение параметра в l -й точке в параметрическом направлении u равно

$$u_1 = 0 \quad \frac{u_l}{u_{\max}} = \frac{\sum_{g=2}^l |D_{g,s} - D_{g-1,s}|}{\sum_{g=2}^r |D_{g,s} - D_{g-1,s}|}.$$

Аналогично для s исходных точек в параметрическом направлении w

$$w_1 = 0 \quad \frac{w_l}{w_{\max}} = \frac{\sum_{g=2}^l |D_{r,g} - D_{r,g-1}|}{\sum_{g=2}^s |D_{r,g} - D_{r,g-1}|},$$

где u_{\max} и w_{\max} являются максимальными значениями соответствующих узловых векторов. На рис. 6-50 a показаны исходные данные и задающая полигональная

сетка, сгенерированная с помощью данного метода. На рис. 6-50 b изображена В-сплайн поверхность, созданная по этой сетке.

Как было ранее отмечено для интерполяции с помощью В-сплайн кривой (см. разд. 5-11), ни данный, ни указанный в [6-5] методы не могут привести к появлению точек или линий складок (разрывов первой или второй производной) в получаемой поверхности.

Отметим из рис. 6-50, что вершины задающей полигональной сетки, полученные из уравнения (6-84), расположены произвольным образом в пространстве. Это неудобно, если в дальнейшем поверхность предполагается модифицировать. В работе [6-27] разработан итерационный метод, основанный на параметрических значениях u , w , в результате которого вершины сетки располагаются на плоскостях или вдоль кривых в трехмерном пространстве.

6-14 РАЗБИЕНИЕ В-СПЛАЙН ПОВЕРХНОСТЕЙ

В-сплайн поверхность разбивается с помощью отдельного разбиения каждой линии задающей полигональной сетки в одном или обоих параметрических направлениях. Может использоваться любой из методов разбиения В-сплайн кривой (см. разд. 5-12). Лучше всего продемонстрировать это на примере.

Пример 6-16 Разбиение незамкнутой В-сплайн поверхности

Рассмотрим незамкнутую В-сплайн поверхность, заданную полигональной сеткой размера 4×4 :

$$\begin{array}{llll} B_{1,1}[-15 & 0 & 15] & B_{2,1}[-5 & 5 & 15] & B_{3,1}[5 & 5 & 15] & B_{4,1}[15 & 0 & 15], \\ B_{1,2}[-15 & 5 & 5] & B_{2,2}[-5 & 10 & 5] & B_{3,2}[5 & 10 & 5] & B_{4,2}[15 & 5 & 5], \\ B_{1,3}[-15 & 5 & -5] & B_{2,3}[-5 & 10 & -5] & B_{3,3}[5 & 10 & -5] & B_{4,3}[15 & 5 & -5], \\ B_{1,4}[-15 & 0 & -15] & B_{2,4}[-5 & 5 & -15] & B_{3,4}[5 & 5 & -15] & B_{4,4}[15 & 0 & -15]. \end{array}$$

Это поверхность четвертого порядка в обоих параметрических направлениях ($k = l = 4$), состоящая из одного куска с диапазонами параметров $0 \leq u \leq 1$, $0 \leq w \leq 1$. Требуется разбить поверхность на 4 подкуска. Следует сохранять однородный незамкнутый узловый вектор.

Вспомним узловой вектор $[0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1]$; для обоих параметрических направлений, найденных в разд. 5-12 и примере 5-18, репараметризуем этот вектор в $[0 \ 0 \ 0 \ 0 \ 2 \ 2 \ 2 \ 2]$, тогда поверхность разбивается с помощью вставки узлового значения 1 в интервале $0 \rightarrow 2$. Таким образом, новый узловой вектор задается координатами $[0 \ 0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2 \ 2]$. Применив уравнения (5-119) и (5-120) к каждой линии сетки в обоих направлениях, получим сетку для разбитой на 4 куска поверхности. Например, рассмотрим разбиение линии сетки в направлении w , заданной $B_{1,j}$, $1 \leq j \leq 4$. Здесь только

$$\alpha'_{4,1} = \alpha'_{4,2} = \alpha'_{4,3} = \alpha'_{4,4} = \alpha'_{4,5} = 1$$

не равны нулю.

Воспользовавшись уравнениями (5-120), получим

$$\alpha_{3,1}^2 = \alpha_{3,2}^2 = \alpha_{3,3}^2 = \alpha_{3,5}^2 = 1; \quad \alpha_{3,4}^2 = \alpha_{4,4}^2 = 1/2,$$

$$\alpha_{2,1}^3 = \alpha_{2,2}^3 = \alpha_{3,5}^3 = 1; \quad \alpha_{2,3}^3 = \alpha_{3,3}^3 = \alpha_{3,4}^3 = \alpha_{4,4}^3 = 1/2,$$

$$\alpha_{1,1}^4 = \alpha_{3,5}^4 = 1; \quad \alpha_{1,2}^4 = \alpha_{2,2}^4 = \alpha_{2,3}^4 = \alpha_{3,3}^4 = \alpha_{3,4}^4 = \alpha_{4,4}^4 = 1/2.$$

Тогда из уравнения (5-119) получим вершины нового характеристического многогранника. В частности,

$$\begin{aligned}
 C_{1,1} &= \alpha_{1,1}^4 B_{1,1} + \alpha_{2,1}^4 B_{1,2} + \alpha_{3,1}^4 B_{1,3} + \alpha_{4,1}^4 B_{1,4} \\
 &= B_{1,1} = [-15 \quad 0 \quad 15] \\
 C_{1,2} &= \alpha_{1,2}^4 B_{1,1} + \alpha_{2,2}^4 B_{1,2} + \alpha_{3,2}^4 B_{1,3} + \alpha_{4,2}^4 B_{1,4} \\
 &= \frac{1}{2}(B_{1,1} + B_{1,2}) = [-15 \quad 2.5 \quad 10] \\
 C_{1,3} &= \alpha_{1,3}^4 B_{1,1} + \alpha_{2,3}^4 B_{1,2} + \alpha_{3,3}^4 B_{1,3} + \alpha_{4,3}^4 B_{1,4} \\
 &= \frac{1}{2}(B_{1,2} + B_{1,3}) = [-15 \quad 5 \quad 0] \\
 C_{1,4} &= \alpha_{1,4}^4 B_{1,1} + \alpha_{2,4}^4 B_{1,2} + \alpha_{3,4}^4 B_{1,3} + \alpha_{4,4}^4 B_{1,4} \\
 &= \frac{1}{2}(B_{1,3} + B_{1,4}) = [-15 \quad 2.5 \quad -10] \\
 C_{1,5} &= \alpha_{1,5}^4 B_{1,1} + \alpha_{2,5}^4 B_{1,2} + \alpha_{3,5}^4 B_{1,3} + \alpha_{4,5}^4 B_{1,4} \\
 &= B_{1,4} = [-15 \quad 0 \quad -15]
 \end{aligned}$$

Выполнив ту же операцию для каждой линии сетки в направлении w , мы получим задающую полигональную сетку размера 4×5 для поверхности, состоящей из двух подкусков в направлении w и одного в направлении u

$$\begin{array}{llll}
 C_{1,1}[-15 \quad 0 \quad 15] & C_{2,1}[-5 \quad 5 \quad 15] & C_{3,1}[5 \quad 5 \quad 15] & C_{4,1}[15 \quad 0 \quad 15] \\
 C_{1,2}[-15 \quad 2.5 \quad 10] & C_{2,2}[-5 \quad 7.5 \quad 10] & C_{3,2}[5 \quad 7.5 \quad 10] & C_{4,2}[15 \quad 2.5 \quad 10] \\
 C_{1,3}[-15 \quad 5 \quad 0] & C_{2,3}[-5 \quad 10 \quad 0] & C_{3,3}[5 \quad 10 \quad 0] & C_{4,3}[15 \quad 5 \quad 0] \\
 C_{1,4}[-15 \quad 2.5 \quad -10] & C_{2,4}[-5 \quad 7.5 \quad -10] & C_{3,4}[5 \quad 7.5 \quad -10] & C_{4,4}[15 \quad 2.5 \quad -10] \\
 C_{1,5}[-15 \quad 0 \quad -15] & C_{2,5}[-5 \quad 5 \quad -15] & C_{3,5}[5 \quad 5 \quad -15] & C_{4,5}[15 \quad 0 \quad -15]
 \end{array}$$

Элементы $\alpha_{i,j}^k$, приведенные выше, применяются также при разбиении поверхности в направлении u . В этом случае задающая полигональная сетка размера 5×4 такова:

$$\begin{array}{llll}
 C_{1,1}[-15 \quad 0 \quad 15] & C_{2,1}[-10 \quad 2.5 \quad 15] & C_{3,1}[5 \quad 5 \quad 15] & C_{4,1}[15 \quad 0 \quad 15] \\
 C_{1,2}[-15 \quad 5 \quad 5] & C_{2,2}[-10 \quad 7.5 \quad 5] & C_{3,2}[5 \quad 7.5 \quad 5] & C_{4,2}[15 \quad 5 \quad 5] \\
 C_{1,3}[-15 \quad 5 \quad -5] & C_{2,3}[-10 \quad 7.5 \quad -5] & C_{3,3}[5 \quad 7.5 \quad -5] & C_{4,3}[15 \quad 5 \quad -5] \\
 C_{1,4}[-15 \quad 0 \quad -15] & C_{2,4}[-10 \quad 2.5 \quad -15] & C_{3,4}[5 \quad 5 \quad -15] & C_{4,4}[15 \quad 0 \quad -15] \\
 \\
 C_{3,1}[0 \quad 5 \quad 15] & C_{4,1}[10 \quad 2.5 \quad 15] & C_{5,1}[15 \quad 0 \quad 15] & \\
 C_{3,2}[0 \quad 10 \quad 5] & C_{4,2}[10 \quad 7.5 \quad 5] & C_{5,2}[15 \quad 5 \quad 5] & \\
 C_{3,3}[0 \quad 10 \quad -5] & C_{4,3}[10 \quad 7.5 \quad -5] & C_{5,3}[15 \quad 5 \quad -5] & \\
 C_{3,4}[0 \quad 5 \quad -15] & C_{4,4}[10 \quad 2.5 \quad -15] & C_{5,4}[15 \quad 0 \quad -15] &
 \end{array}$$

Здесь поверхность состоит из двух подкусков в направлении u и одного в направлении w .

Разбиение поверхности как в u , так и в w направлениях приведет в результате к задающей полигональной сетке размера 5×5 , заданной в виде

$$\begin{array}{lll}
 C_{1,1}[-15 & 0 & 15] & C_{2,1}[-10 & 2.5 & 15] \\
 C_{1,2}[-15 & 2.5 & 10] & C_{2,2}[-10 & 5 & 10] \\
 C_{1,3}[-15 & 5 & 0] & C_{2,3}[-10 & 7.5 & 0] \\
 C_{1,4}[-15 & 2.5 & -10] & C_{2,4}[-10 & 5 & -10] \\
 C_{1,5}[-15 & 0 & -15] & C_{2,5}[-10 & 2.5 & -15] \\
 \\
 C_{3,1}[0 & 5 & 15] & C_{4,1}[10 & 2.5 & 15] & C_{5,1}[15 & 0 & 15] \\
 C_{3,2}[0 & 7.5 & 10] & C_{4,2}[10 & 5 & 10] & C_{5,2}[15 & 2.5 & 10] \\
 C_{3,3}[0 & 10 & 0] & C_{4,3}[10 & 7.5 & 0] & C_{5,3}[15 & 5 & 0] \\
 C_{3,4}[0 & 7.5 & -10] & C_{4,4}[10 & 5 & -10] & C_{5,4}[15 & 2.5 & -10] \\
 C_{3,5}[0 & 5 & -15] & C_{4,5}[10 & 2.5 & -15] & C_{5,5}[15 & 0 & -15]
 \end{array}$$

Заметим, что последняя сетка получается из первой или второй с помощью уравнений (5-119) и (5-120).

Исходная поверхность и все три полученные при ее разбиении сетки изображены на рис. 6-51. Каждая из поверхностей идентична исходной.

Ясно, что при дальнейшем разбиении поверхности задающая полигональная сетка приближается к поверхности.

6-15 ГАУССОВА КРИВИЗНА И КАЧЕСТВО ПОВЕРХНОСТИ

В автоматизированном проектировании большой интерес представляет разработка соответствующих методов для определения и/или визуализации качества или гладкости поверхностей. Хорошо известно, что используемые обычно бикубические поверхности (Кунса, Безье или В-сплайн), хотя и являются во всех точках C^2 гладкими, в некоторых местах могут быть плоскими или выпуклыми либо волнистыми. В настоящее время самые лучшие математические методы определения качества поверхности используют Эйлеровы (ортогональные) сетки минимальной и максимальной кривизны (см. [6-28] и [6-29]) и гауссовой кривизны (см. [6-28] – [6-32], и разд. 6-8).

Вспомним (разд. 6-8), что две комбинации главных кривизн, называемые средней и гауссовой (общей) кривизнами, характеризуют локальную форму поверхности. Средняя кривизна определяется как

$$\kappa_a = \frac{\kappa_{\min} + \kappa_{\max}}{2}. \quad (6-45)$$

Гауссова кривизна определяется как

$$\kappa_g = \kappa_{\min} \cdot \kappa_{\max}, \quad (6-46)$$

где κ_{\min} и κ_{\max} являются главными кривизнами. Гауссова кривизна в точке на поверхности показывает, является ли поверхность локально эллиптической, гиперболической или параболической (гауссова кривизна положительна, отрицательна или равна нулю).

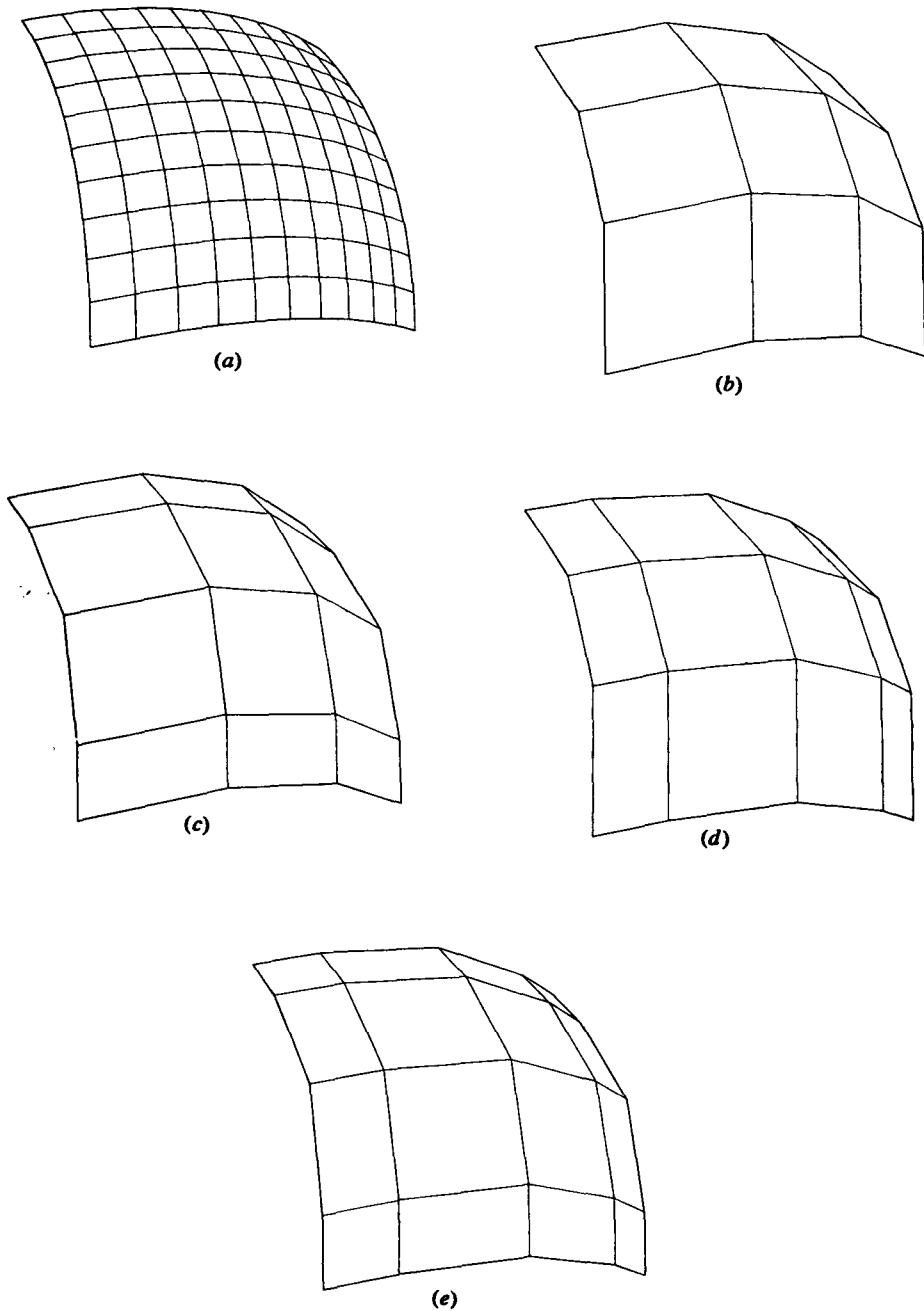


Рис. 6-51 Разбиение В-сплайн поверхности. (a) Поверхность; (b) исходная задающая полигональная сетка; (c) сетка, разбитая в w направлении; (d) сетка, разбитая в u направлении; (e) сетка, разбитая в обоих направлениях.

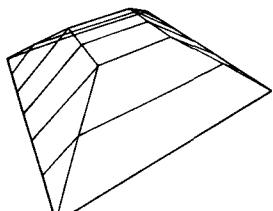
Интересно отметить здесь, что если гауссова кривизна равна нулю, то поверхность является развертывающейся, т. е. она может быть развернута в плоскость. Такая поверхность изогнута в одном направлении, например, конус или цилиндр. Это подразумевает, что одна из главных кривизн, κ_{\min} и κ_{\max} равна нулю. Следовательно, равна нулю и гауссова кривизна.

Существуют несколько методов визуализации средней и гауссовой кривизн поверхности. Если изобразительные возможности ограничены рисованием отрезков, то наиболее полезны контурные чертежи (см. [6-28] и [6-29]). В работах [6-30] и [6-32] показано, что эффективным методом этого является кодирование гауссовой кривизны на растровом изображении с помощью цветов или набора полутоонов серого цвета.

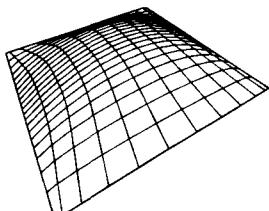
На рис. 6-52 показаны закодированные¹ с помощью полутоонов серого цвета изображения гауссовой кривизны для нескольких тестовых поверхностей вместе с соответствующим характеристическим многогранником (слева) и проволочным параметрическим представлением поверхностей (по середине). Все поверхности являются бикубическими ($k = l = 4$) В-сплайн поверхностями. На рис. 6-52 изображены три поверхности по мере увеличения степени нарушения их гладкости. На рис. 6-52a представлена совершенно гладкая поверхность, без изъянов. Два ярко выраженных гребня уменьшенной гладкости на рис. 6-52b вызваны совпадением на каждой стороне трех линий полигональной сетки. На рис. 6-52c удлиненная линия «складки» в середине поверхности получилась из-за совпадения трех линий полигональной сетки на участке, пересекающем несколько внутренних линий сетки, что показано на характеристическом многограннике.

Вообще, изображения закодированной гауссовой кривизны более наглядно показывают свойства поверхностей. Например, рисунки 6-52a и b демонстрируют большое отрицательное значение кривизны в угловых точках. Эта отрицательная кривизна является результатом ограничений на границы поверхности — они должны быть прямыми и плоскими, тогда как внутренняя область выпукла и положительно изогнута. Закодированное изображение гауссовой кривизны на рис. 6-52b подчеркивает уплощение области, расположенной между гребнями. Отметим, что так как гауссова кривизна равна нулю в этой области, то эта часть поверхности является развертывающейся. Отметим также, что задающая полигональная сетка в этой области является развертывающейся. И наконец, полоса поперек середины закодированного изображения на рис. 6-52c показывает, что поверхность в этой области представляет собой плоскость, согнутую посередине. Тот факт, что сгиб является прямой линией, объясняет исчезновение гауссовой кривизны вдоль этой линии.

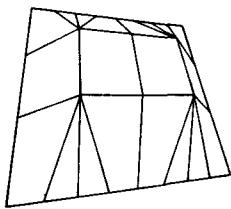
¹ Эти изображения являются монохромными копиями цветных изображений из [6-33]. При кодировании изображений гауссовой кривизны были усреднены значения кривизн четырех вершинах многоугольников, образуемых густой сеткой аппроксимирующих поверхность четырехугольников. Эта усредненная величина была присвоена каждому многоугольнику. Диапазон изменения кривизны был разделен на ряд одинаковых интервалов (за исключением концов), соответствующих доступному диапазону интенсивностей. Справа от изображения указан список полутоонов и соответствующих им значений кривизны. Ступенчатость (границы между различными интенсивностями) возникает из-за ограниченного числа доступных интенсивностей, а не из-за аппроксимации с помощью многоугольников.



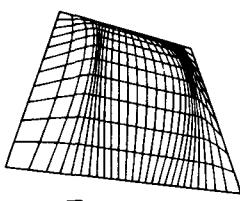
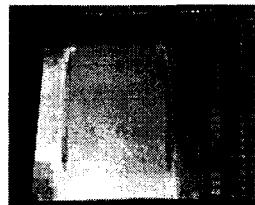
Полигональная сетка

Поверхность
(a)

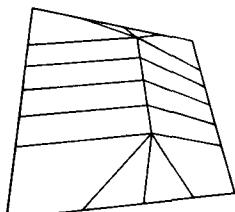
Гауссова кривизна



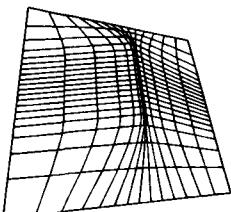
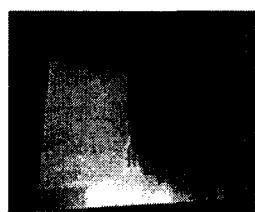
Полигональная сетка

Поверхность
(b)

Гауссова кривизна



Полигональная сетка

Поверхность
(c)

Гауссова кривизна

Рис. 6-52 Гауссова кривизна. (a) Гладкая поверхность; (b) короткая линия «складки»; (c) более длинная линия «складки». (С разрешения Дж. Дилла и Д. Роджерса.)

Метод вычисления гауссовой кривизны будет проиллюстрирован на примере.

Пример 6-17 Гауссова кривизна

Найти гауссову кривизну в точке с параметрами $u = 1/2$, $w = 1$ для незамкнутой В-сплайн поверхности, определенной ранее в примере 6-15.

Вспомним сначала базисные функции $N_{i,4}$ и $M_{j,l}$ из примера 6-15. По этим данным можно вычислить первую и вторую производные, необходимые для нахождения Q_u , Q_w , Q_{uw} , Q_{uu} и Q_{ww} , и последующего вычисления гауссовой кривизны. А именно

$$\begin{array}{lll} N_{1,4} = (1-u)^3 & N'_{1,4} = -3(1-u)^2 & N_{1,4} = 6(1-u) \\ N_{2,4} = 3u(1-u)^2 & N'_{2,4} = 3(1-u)(1-3u) & N''_{2,4} = 6(3u-2) \\ N_{3,4} = 3u^2(1-u) & N'_{3,4} = 3u(2-3u) & N'''_{3,4} = 6(1-3u) \\ N_{4,4} = u^3 & N'_{4,4} = 3u^2 & N''_{4,4} = 6u \end{array}$$

и

$$\begin{array}{lll} M_{1,3} = 0 & M'_{1,3} = 0 & M''_{1,3} = 0 \\ M_{2,3} = \frac{(2-w)^2}{2} & M'_{2,3} = w - 2 & M''_{2,3} = 1 \\ M_{3,3} = \frac{(2-w)(3w-2)}{2} & M'_{3,3} = 4 - 3w & M''_{3,3} = -3 \\ M_{4,3} = (w-1)^2 & M'_{4,3} = 2(w-1) & M''_{4,3} = 2 \end{array}$$

Вычисление производных в точке $u = 1/2$, $w = 1$ и подстановка в уравнения (6-71) – (6-75) приводят к следующим результатам:

$$\begin{aligned} Q(1/2, 1) &= [0 \quad 35/4 \quad 0], \\ Q_u(1/2, 1) &= [30 \quad 0 \quad 0], \\ Q_w(1/2, 1) &= [0 \quad 0 \quad 10], \\ Q_{uw}(1/2, 1) &= [0 \quad 0 \quad 0], \\ Q_{uu}(1/2, 1) &= [0 \quad -30 \quad 0], \\ Q_{ww}(1/2, 1) &= [0 \quad -10 \quad 10]. \end{aligned}$$

Компоненты уравнения (6-48) для гауссовой кривизны таковы:

$$\begin{aligned} Q_u \times Q_w &= [30 \quad 0 \quad 0] \times [0 \quad 0 \quad 10] = [0 \quad -300 \quad 0] \\ |Q_u \times Q_w|^4 &= (300)^4 \\ A &= [Q_u \times Q_w] \cdot Q_{uu} = [0 \quad -300 \quad 0] \cdot [0 \quad -30 \quad 0] = 9000 \\ B &= [Q_u \times Q_w] \cdot Q_{uw} = [0 \quad -300 \quad 0] \cdot [0 \quad 0 \quad 0] = 0 \\ C &= [Q_u \times Q_w] \cdot Q_{ww} = [0 \quad -300 \quad 0] \cdot [0 \quad -10 \quad 10] = 3000 \end{aligned}$$

Используя уравнение (6-48), получим гауссову кривизну

$$\kappa_g = \frac{AC - B}{|Q_u \times Q_w|^4} = \frac{(9000)(3000) - (0)}{(300)^4} = 3.33 \times 10^{-3}$$

Так как $\kappa_g > 0$, то поверхность является локально эллиптической.

6-16 РАЦИОНАЛЬНЫЕ В-СПЛАЙН ПОВЕРХНОСТИ

Так же как и для рациональных кривых, возможно существование рациональных форм квадратичных поверхностей, бикубических поверхностей Кунса и поверхностей Безье. Однако здесь будут рассмотрены только рациональные В-сплайн поверхности, как из-за нехватки места, так и в связи с тем, что такие поверхности являются обобщением всех этих форм.

Декартово произведение рациональной В-сплайн поверхности в четырехмерном пространстве однородных координат задается формулой

$$Q(u, w) = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j}^h N_{i,k}(u) M_{j,l}(w), \quad (6-85)$$

где $B_{i,j}^h$ являются 4D однородными вершинами характеристического многогранника, а $N_{i,k}(u)$ и $M_{j,l}(w)$ являются нерациональными В-сплайн базисными функциями, приведенными ранее в уравнении (5-84).

Проектирование обратно в трехмерное пространство с помощью деления на однородную координату дает рациональную В-сплайн поверхность

$$Q(u, w) = \frac{\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N_{i,k}(u) M_{j,l}(w)}{\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M_{j,l}(w)} = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} B_{i,j} S_{i,j}(u, w), \quad (6-86)$$

где $B_{i,j}^h$ являются 3D-точками задающей полигональной сетки и $S_{i,j}(u, w)$ — базисными функциями от двух переменных рациональной В-сплайн поверхности

$$S_{i,j}(u, w) = \frac{h_{i,j} N_{i,k}(u) M_{j,l}(w)}{\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M_{j,l}(w)}. \quad (6-87)$$

Удобно положить $h_{i,j} \geq 0$ для всех i, j .

Важно отметить здесь, что $S_{i,j}(u, w)$ не являются произведением $R_{i,k}(w)$ и $R_{j,l}(w)$ (см. уравнение 5-123). Тем не менее $S_{i,j}(u, w)$ имеют форму и аналитические свойства, похожие на функцию произведения $N_{i,k}(u) M_{j,l}(w)$. Следовательно, рациональные В-сплайн поверхности имеют аналитические и геометрические свойства, похожие на их нерациональные двойники. Конкретнее,

Сумма базисных функций рациональной поверхности для любых значений u, w равна

$$\sum_{i=1}^{n+1} \sum_{j=1}^{m+1} S_{i,j}(u, w) \equiv 1. \quad (6-88)$$

Каждая базисная функция рациональной поверхности положительна или равна нулю для всех значений параметров u, w , т. е. $S_{i,j} \geq 0$.

Кроме случая $k = 1$ или $l = 1$, каждая базисная функция рациональной поверхности имеет ровно один максимум.

Максимальный порядок рациональной В-сплайн поверхности в каждом параметрическом направлении равен числу вершин характеристического многоугольника в этом направлении.

Рациональная В-сплайн поверхность порядка k, l (степени $k - 1, l - 1$) гладкая во всех точках C^{k-2}, C^{l-2} .

Рациональная В-сплайн поверхность инвариантна относительно проективного преобразования; т. е. любое проективное преобразование может быть применено к поверхности путем его применения к задающей полигональной сетке. Отметим, что это условие является более строгим, чем для нерациональной В-сплайн поверхности.

Поверхность лежит внутри выпуклой оболочки задающей полигональной сетки, образуемой объединением всех выпуклых оболочек k, l соседних вершин полигональной сетки.

Свойство затухания изменений для рациональных В-сплайн поверхностей неизвестно.

Влияние одной вершины полигональной сетки ограничивается $\pm k/2, \pm 1/2$ интервалами в каждом параметрическом направлении.

При триангуляции задающая полигональная сетка образует плоскую аппроксимацию поверхности.

Если число вершин задающей полигональной сетки равно порядку в каждом параметрическом направлении и дублированных внутренних узловых величин нет, то рациональная B-сплайн поверхность является рациональной поверхностью Безье.

Из уравнений (6-86) и (6-87) ясно, что когда все $h_{i,j} = 1$, тогда $S_{i,j}(u,w) = N_{i,k}(u)M_{j,l}(w)$. Таким образом, базисные функции рациональной B-сплайн поверхности и сами поверхности превращаются в их нерациональные эквиваленты. Следовательно, рациональные B-сплайн поверхности представляют соответственно обобщение нерациональных B-сплайн поверхностей и рациональных и нерациональных поверхностей Безье.

Снова, как и в случае рациональных B-сплайн кривых, алгоритмы для увеличения степени, разбиения (см. разд. 6-14) и интерполяции (см. разд. 6-13) нерациональных B-сплайн поверхностей можно применять просто путем их использования для 4D-вершин задающей полигональной сетки.

Незамкнутый однородный, периодический однородный и неоднородный узловые векторы могут быть использованы для генерации рациональных B-сплайн базисных функций и рациональных B-сплайн поверхностей. Типы узловых векторов могут смешиваться. Например, в параметрическом u -направлении может использоваться незамкнутый однородный узловый вектор, а в w -направлении — неоднородный узловый вектор. Сначала мы остановимся на незамкнутых однородных узловых векторах.

На рис. 6-53 представлены бикубическая ($k = l = 4$) рациональная B-сплайн поверхность и ее задающая полигональная сетка для $h_{1,3} = h_{2,3} = 0, 1, 5$. Изображение на рис. 6-53c с $h_{1,3} = h_{2,3} = 1$ идентично нерациональной B-сплайн поверхности. Результаты варьирования значений однородной координаты можно заметить, сравнивая рис. 6-53c с рис. 6-53b и d. Эти результаты аналогичны полученным для рациональных B-сплайн кривых (см. разд. 5-13), но проявляются не в столь сильной степени. В данном случае уменьшение эффекта обусловлено тем, что $S_{i,j}(u,w)$ является функцией смешения от двух переменных.

На рис. 6-54a и b иллюстрируется результат, полученный при присвоении всем внутренним $h_{i,j}$ значений 0 и 500, соответственно; т. е. $h_{2,2} = h_{2,3} = h_{3,2} = h_{3,3} = h_{4,2} = h_{4,3} = 0$ или 500. Все остальные $h_{i,j} = 1$. Характеристический многогранник изображен на рис. 6-53a. Установка всех внутренних $h_{i,j} = 0$ приводит к игнорированию внутренних вершин характеристического многогранника. Интерполируются только вершины граничных кривых. Напротив, присвоение всем внутренним $h_{i,j} = 500$ сводит к минимуму влияние граничных вершин. Заметим, что изменение $h_{i,j}$ влияет на параметризацию поверхности. Это иллюстрируется скоплением параметрических линий около границ поверхности, когда внутренние $h_{i,j} = 0$ (см. рис. 6-54a), и во внутренней части поверхности, когда внутренние $h_{i,j} = 500$ (см. рис. 6-54b).

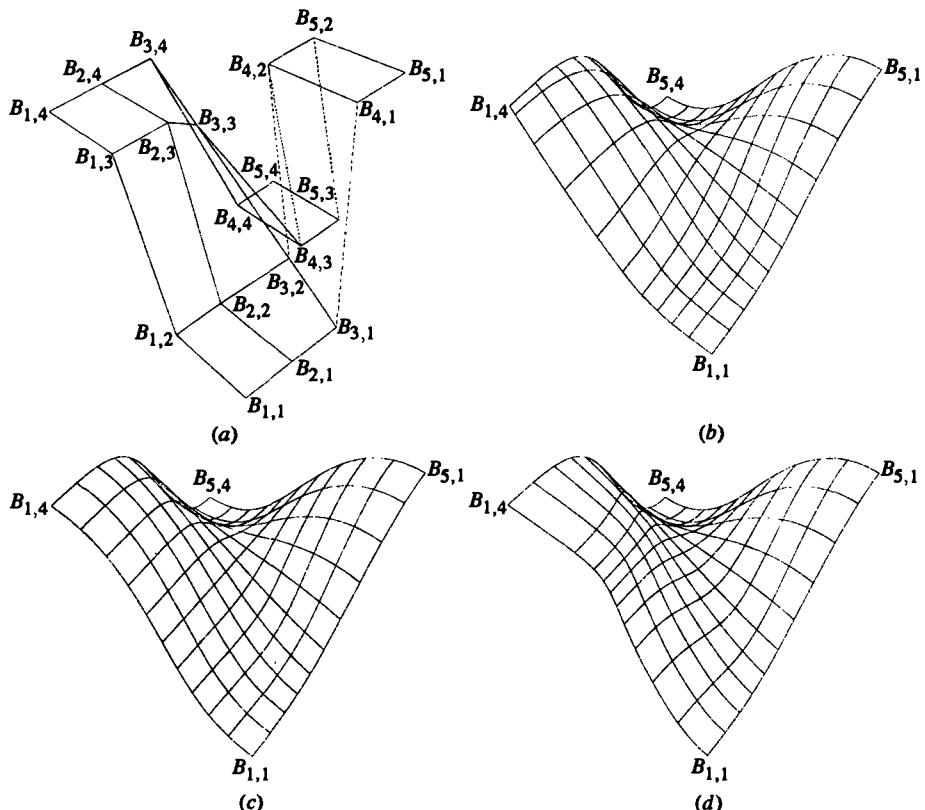


Рис. 6-53 Рациональные В-сплайн поверхности с $n+1 = 5$, $m+1 = 4$, $k = l = 4$.
 (a) Характеристический многогранник; (b) $h_{1,3} = h_{2,3} = 0$; (c) $h_{1,3} = h_{2,3} = 1$; (d) $h_{1,3} = h_{2,3} = 5$.

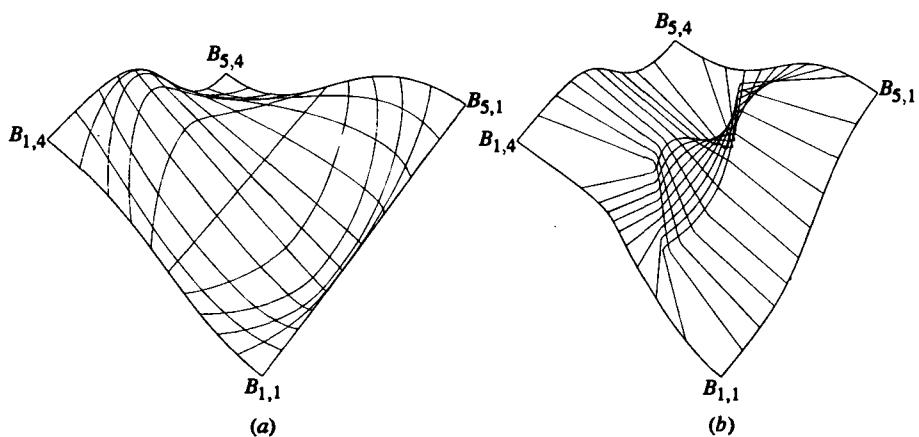


Рис. 6-54 Рациональные В-сплайн поверхности с $n+1 = 5$, $m+1 = 4$, $k+l = 4$. (a) Все внутренние $h_{i,j} - e = 0$; (b) все внутренние $h_{i,j} - e = 500$.

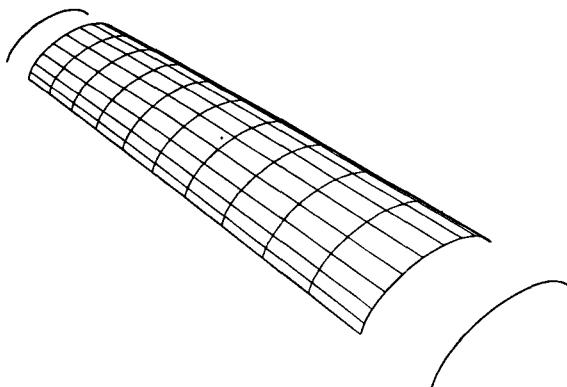


Рис. 6-55 Рациональный В-сплайн эллиптический цилиндр, сгенерированный заметанием рациональной эллиптической кривой на рис. 5-67б.

Результаты, возникающие при совпадении нескольких вершин или линий сетки аналогичны результатам, полученным для нерациональных В-сплайн поверхностей (см. разд. 6-12) и рациональных В-сплайн кривых (см. разд. 5-13). Также аналогичны результаты смещения одной вершины на поверхности.

Одним из самых привлекательных свойств рациональных В-сплайн поверхностей является их способность представлять квадратичные поверхности и плавно переходить в скульптурные поверхности высоких степеней. В качестве простого примера квадратичной поверхности рассмотрим обобщенный цилиндр, образуемый заметанием кривой. Ясно, что в направлении заметания такая поверхность должна быть второго порядка, т. е. прямой линией. Следовательно, для заметаемой в направлении u поверхности формула выглядит (см. [6-33]) следующим образом:

$$Q(u, w) = \sum_{i=1}^2 \sum_{j=1}^{m+1} B_{i,j} S_{i,j}(u, w), \quad (6-89)$$

где $S_{i,j}(u, w)$ имеет в параметрическом направлении w тот же порядок, что и кривая, и порядок 2 в направлении u . Кроме того, вершины характеристического многогранника в направлении u равны $B_{1,j} = B_j$ и $B_{2,j} = B_j + sD$, где D задает направление и расстояние заметания. Параметр s изменяется в диапазоне $0 \leq s \leq 1$. Элементы B_j являются вершинами задающего многоугольника для заметающей кривой. Однородные координаты остаются неизменными в направлении заметания; т. е. $h_{1,j} = h_{2,j} = h_j$, где h_j – это однородная координата для заметаемой кривой. На рис. 6-55 изображен эллиптический цилиндр, созданный с использованием эллиптической кривой, представленной на рис. 5-67б. С каждой стороны поверхности показана отодвинутая заметаемая кривая.

Рациональные В-сплайн поверхности используются также для создания линейчатых поверхностей. Эллиптический цилиндр, изображенный на рис. 6-55, является, конечно, линейчатой поверхностью. Для генерации с помощью рацио-

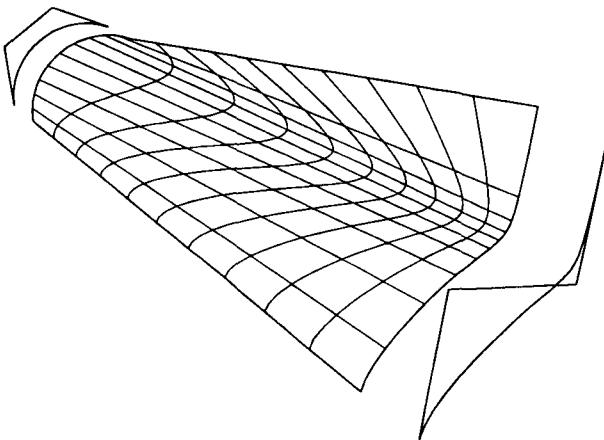


Рис. 6-56 Рациональная В-сплайн линейчатая поверхность.

нальных В-сплайнов линейчатой поверхности более общего вида требуется, чтобы обе кривые имели одинаковый порядок (степень), одинаковый узловый вектор и одинаковое число вершин задающего многоугольника. Если кривые имеют разный порядок (степень), то степень кривой меньшего порядка увеличивают (см. разд. 5-8 и пример 6-18). Требуемый узловый вектор является объединением узловых векторов двух кривых. Любые кратные узловые значения для любой кривой включаются в окончательный узловый вектор. Для обеспечения идентичности обоих узловых векторов используется вставка узлов (см. разд. 5-12). Увеличение степени и вставка узлов обеспечивают равенство числа вершин характеристического многоугольника для обеих кривых. Получающаяся рациональная В-сплайн линейчатая поверхность описывается уравнением (6-89) с

$$P_1(w) = Q(0, w) = \sum_{j=1}^{m+1} B_{1,j} R_{j,l}(w) \quad \text{и} \quad P_2(w) = \sum_{j=1}^{m+1} B_{2,j} R_{j,l}(w) = Q(1, w).$$

На рис. 6-56 показан пример линейчатой поверхности, переводящей четверть окружности в рациональную В-сплайн кривую четвертого порядка. Кривые и их характеристические многоугольники изображены отодвинутыми от края поверхности. Данный метод лучше проиллюстрировать на примере.

Пример 6-18 Рациональные В-сплайн линейчатые поверхности

Найти точку с параметрами $u = w = 0.5$ на линейчатой поверхности, образованной смещением 120° дуги окружности и рациональной кривой четвертого порядка. Дуга окружности представляется рациональной В-сплайн кривой, заданной $B_{1,1}[0 \ 0 \ 0]$, $B_{1,2}[1 \ \sqrt{3} \ 0]$, $B_{1,3}[2 \ 0 \ 0]$ и $[H] = [1 \ 1/2 \ 1]$. Вторая кривая определяется $B_{2,1}[0 \ 0 \ 10]$, $B_{2,2}[1 \ 1 \ 10]$, $B_{3,2}[2 \ 0 \ 10]$, $B_{4,2}[3 \ 1 \ 10]$ с $[H] = [1 \ 3/4 \ 5 \ 1]$.

Сначала необходимо увеличить степень дуги окружности. В действительности дуга является рациональной кривой Безье. Обсуждавшийся в разд. 5-8

метод увеличения степени в рациональном случае применяется к 4D однородным координатам. В результате получаем:

$$\begin{aligned} B_1^{h*} &= B_1^h, \\ B_i^{h*} &= \alpha_i B_{i-1}^h + (1 - \alpha_i) B_i^h \quad \alpha_i = \frac{i}{n+1} \quad i = 2, \dots, n, \\ B_{n+1}^{h*} &= B_n^h. \end{aligned}$$

После проецирования обратно в трехмерное пространство имеем:

$$\begin{aligned} B_1^* &= B_1, \\ B_i^* &= \frac{\alpha_i h_{i-1} B_{i-1} + (1 - \alpha_i) h_i B_i}{\alpha_i h_{i-1} + (1 - \alpha_i) h_i} \quad i = 2, \dots, n, \\ h_i^* &= \alpha_i h_{i-1} + (1 - \alpha_i) h_i. \end{aligned}$$

Использование данных результатов для увеличения степени 120° дуги приведет к

$$\begin{aligned} h_{1,1}^* &= h_{1,1} = 1, \\ B_{1,1}^* &= B_{1,1} = [0 \quad 0 \quad 0], \\ h_{1,2}^* &= \left(\frac{1}{3}\right)(1) + \left(\frac{2}{3}\right)\left(\frac{1}{2}\right) = \frac{2}{3}, \\ B_{1,2}^* &= \left\{ \frac{1}{3}(1)[0 \quad 0 \quad 0] + \frac{2}{3}\left(\frac{1}{2}\right)[1 \quad \sqrt{3} \quad 0] \right\} / \left(\frac{2}{3}\right) = \left[\frac{1}{2} \quad \frac{\sqrt{3}}{2} \quad 0\right], \\ h_{1,3}^* &= \left(\frac{2}{3}\right)\left(\frac{1}{2}\right) + \left(\frac{1}{3}\right)(1) = \frac{2}{3}, \\ B_{1,3}^* &= \left\{ \left(\frac{2}{3}\right)\left(\frac{1}{2}\right)[1 \quad \sqrt{3} \quad 0] + \frac{1}{3}(1)[2 \quad 0 \quad 0] \right\} / \left(\frac{2}{3}\right) = \left[\frac{3}{2} \quad \frac{\sqrt{3}}{2} \quad 0\right], \\ h_{1,4}^* &= h_{1,3} = 1, \\ B_{1,4}^* &= B_{1,3} = [2 \quad 0 \quad 0]. \end{aligned}$$

Теперь каждая кривая имеет четыре вершины характеристического многоугольника. Узловой вектор для каждой кривой равен $[X] = [Y] = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1]$. Следовательно, включение узлов применять не нужно.

Для $u = w = 0.5$ из уравнения (5-84) получим

$$\begin{aligned} N_{1,2} &= 0.5; & N_{2,2} &= 0.5 \\ M_{1,4} &= 0.125; & M_{2,4} &= 0.375; & M_{3,4} &= 0.375; & M_{4,4} &= 0.125 \end{aligned}$$

Тогда уравнение (6-87) приведет к следующим результатам:

$$\begin{aligned} S_{1,1} &= 0.0396 & S_{1,2} &= 0.0792 & S_{1,3} &= 0.0792 & S_{1,4} &= 0.0396 \\ S_{2,1} &= 0.0396 & S_{2,2} &= 0.0891 & S_{2,3} &= 0.594 & S_{2,4} &= 0.0396 \end{aligned}$$

Координаты точки на поверхности

$$Q(0.5, 0.5) = [1.634 \quad 0.266 \quad 7.624].$$

Вся поверхность изображена на рис. 6-56.

С помощью В-сплайнов могут быть также представлены поверхности вращения. Предположим, что

$$P(w) = \sum_{j=1}^{m+1} B_j B_{j,l}(w)$$

с узловым вектором $[Y]$ является рациональной В-сплайн кривой, и вспомним, что полная окружность получается при объединении четырех четвертей окружности, заданных девятью вершинами многоугольника (см. разд. 5-13). Это приводит нас к рациональной В-сплайн поверхности вращения, задаваемой (см. [6-33])

$$Q(u, w) = \sum_{i=1}^9 \sum_{j=1}^{m+1} B_{i,j} S_{i,j}(u, w), \quad (6-90)$$

где узловой вектор $[X] = [0 \ 0 \ 0 \ 1 \ 1 \ 2 \ 2 \ 3 \ 3 \ 4 \ 4 \ 4]$. Предположим, что вращение происходит вокруг оси z , кривая $P(w)$ определена в плоскости xz , элементы $B_{i,j}$ задаются как $B_{1,j} = B_j$ для фиксированного j с $1 \leq i \leq 9$. Вершины характеристического многоугольника образуют угловые и средние точки квадрата, лежащего в плоскости, перпендикулярной оси z , и с размером стороны, равным удвоенному радиусу окружности вращения. Однородные весовые множители являются произведением множителей, необходимых для задания рациональной В-сплайн кривой и для задания окружности вращения. Конкретнее, для фиксированного j , $h_{1,j} = h_j$, $h_{2,j} = h_j\sqrt{2}/2$, $h_{3,j} = h_j$, $h_{4,j} = h_j\sqrt{2}/2, \dots, h_{9,j} = h_j$. На рис. 6-57 показаны характеристический многоугольник и сама кривая для окружности вращения и для вращаемой рациональной В-сплайн кривой. Также на рисунке приведены объединенный характеристический многогранник поверхности и сама поверхность.

На рис. 6-58 и 6-59 представлены обычные поверхности вращения — тор и сфера, вместе со своими характеристическими многогранниками. Тор генерируется путем вращения смещенной окружности вокруг одной из осей. Сфера генерируется путем вращения полуокружности, составленной из двух 90° дуг вокруг оси, проходящей через диаметр полуокружности.

Как было ранее упомянуто, одной из наиболее мощных характеристик рациональных в отличие от нерациональных В-сплайн поверхностей является их способность «упрятывать» (или включать) квадратичные элементы поверхности внутри обобщенной скульптурной поверхности. Например, как часть более общей поверхности, может быть включен цилиндрический элемент. На рис. 6-60 представлено три примера. Центральной частью каждой поверхности четвертого порядка является секция кругового цилиндра. Рис. 6-60a мог бы изображать переднюю кромку крыла или лопасти¹ турбины, рис. 6-60b — цилиндрический нос корабля. Обе поверхности генерируются с помощью задания дуги окружности третьего порядка (см. разд. 5-13), увеличения степени дуги (см. пример 6-18), создания из дуги линейчатой поверхности и ее включения между двумя крайними элементами поверхностей четвертого порядка. Случайно обе поверхности, показанные

¹ В аэродинамических профилях NASA для задания передней кромки используется дуга окружности.

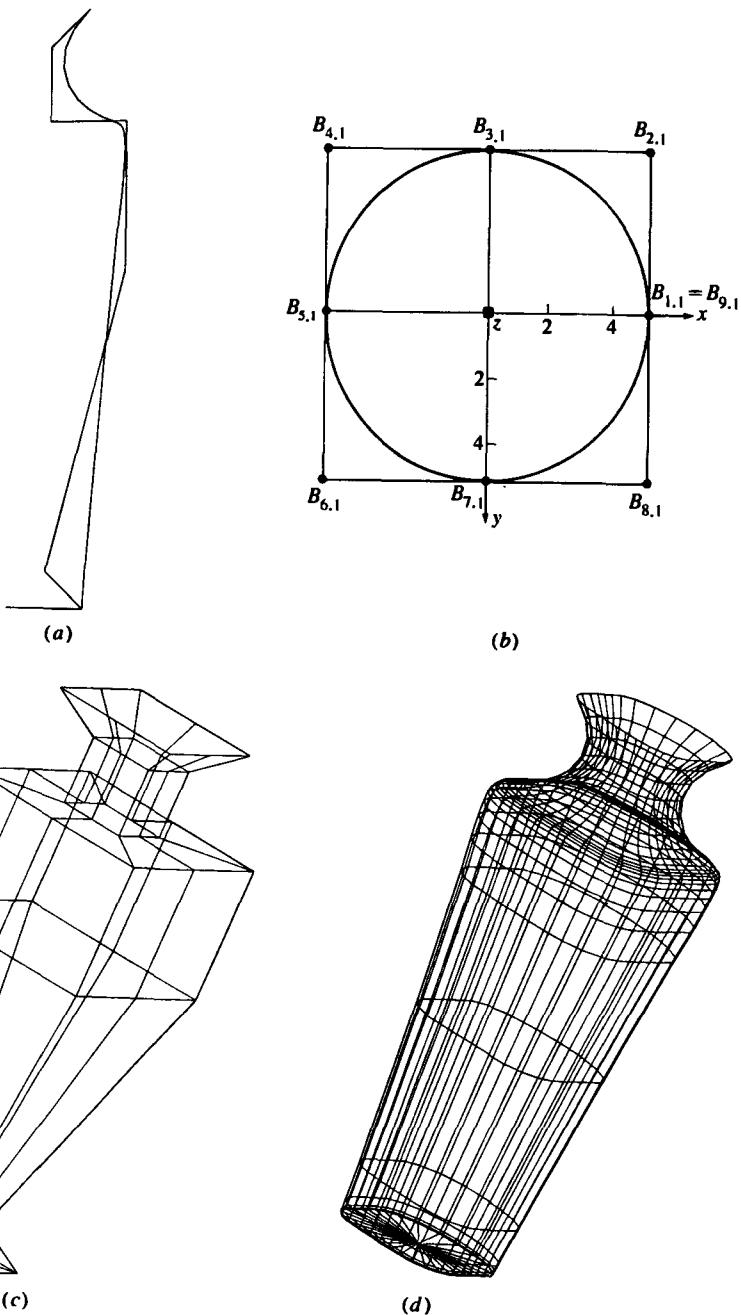


Рис. 6-57 Рациональная В-сплайн поверхность вращения. (а) Образующая кривая и задающая сетка; (б) окружность вращения; (с) характеристический многогранник; (д) поверхность вращения.

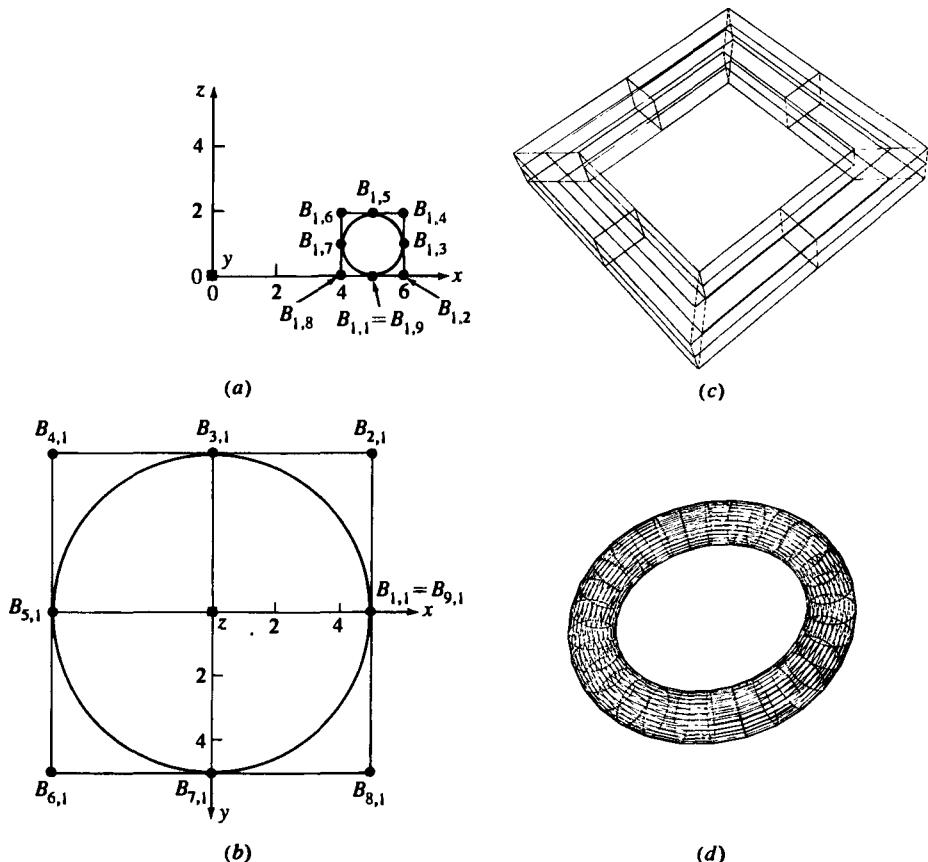


Рис. 6-58 Тор, сгенерированный как рациональная В-сплайн поверхность. (а) Смещенная окружность и задающий многоугольник; (б) окружность вращения и задающий многоугольник; (с) характеристический многогранник для тора; (д) тор.

на рис. 6-60 a и b , оказались линейчатыми развертывающимися поверхностями. На рис. 6-60 c изображен цилиндрический элемент, упакованный в более общую поверхность.

Производные рациональной В-сплайн поверхности получаются с помощью формального дифференцирования уравнения (6-86):

$$Q_u = \frac{\bar{N}}{D} \left(\frac{\bar{N}_u}{\bar{N}} - \frac{\bar{D}_u}{\bar{D}} \right), \quad (6-91a)$$

$$Q_w = \frac{\bar{N}}{D} \left(\frac{\bar{N}_w}{\bar{N}} - \frac{\bar{D}_w}{\bar{D}} \right), \quad (6-91b)$$

$$Q_{uw} = \frac{\bar{N}}{D} \left(\frac{\bar{N}_{uw}}{\bar{N}} - \frac{\bar{N}_u}{\bar{N}} \frac{\bar{D}_w}{\bar{D}} - \frac{\bar{N}_w}{\bar{N}} \frac{\bar{D}_u}{\bar{D}} + 2 \frac{\bar{D}_u}{\bar{D}} \frac{\bar{D}_w}{\bar{D}} - \frac{\bar{D}_{uw}}{\bar{D}} \right), \quad (6-91c)$$

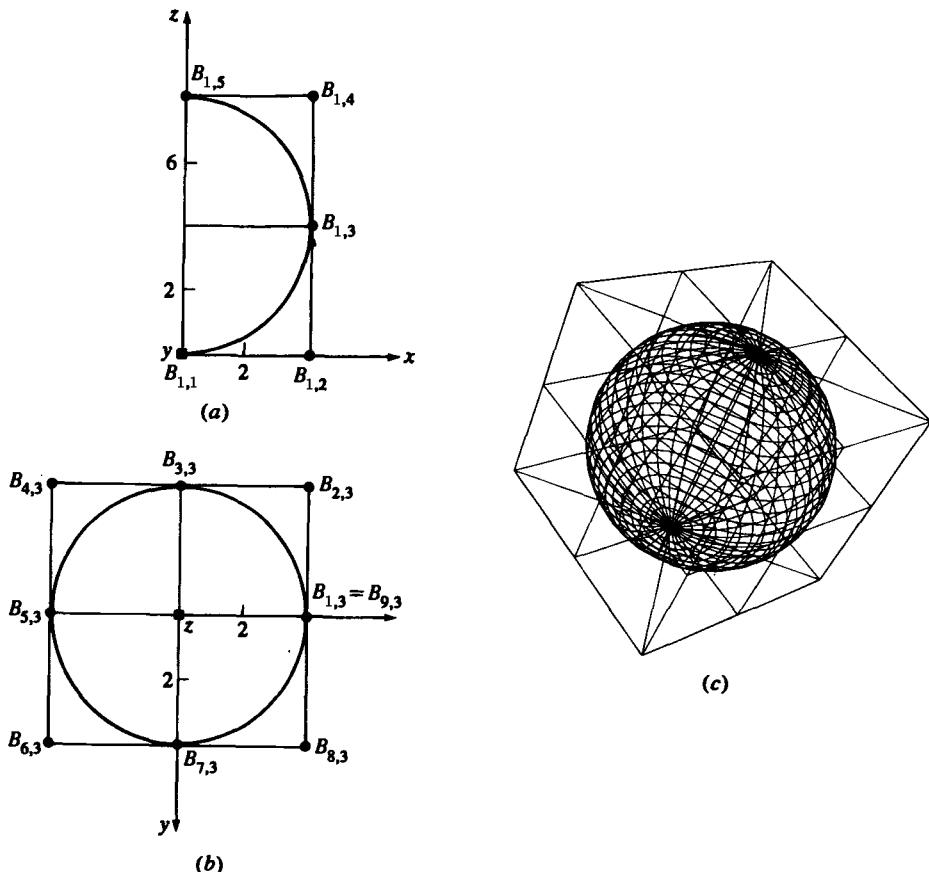


Рис. 6-59 Сфера, сгенерированная как рациональная В-сплайн поверхность. (а) Смещенная окружность и задающий многоугольник; (б) окружность вращения и задающий многоугольник; (с) характеристический многогранник и сфера.

$$Q_{uu} = \frac{\bar{N}}{D} \left(\frac{\bar{N}_{uu}}{\bar{N}} - 2 \frac{\bar{N}_u}{\bar{N}} \frac{\bar{D}_u}{D} + 2 \frac{\bar{D}_u^2}{D^2} - \frac{\bar{D}_{uu}}{D} \right), \quad (6-91d)$$

$$Q_{ww} = \frac{\bar{N}}{D} \left(\frac{\bar{N}_{ww}}{\bar{N}} - 2 \frac{\bar{N}_w}{\bar{N}} \frac{\bar{D}_w}{D} + 2 \frac{\bar{D}_w^2}{D^2} - \frac{\bar{D}_{ww}}{D} \right), \quad (6-91e)$$

где \bar{N} и \bar{D} являются числителем и знаменателем, соответственно, уравнения (6-86) с производными

$$\bar{N}_u = \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N'_{i,k}(u) M_{j,l}(w),$$

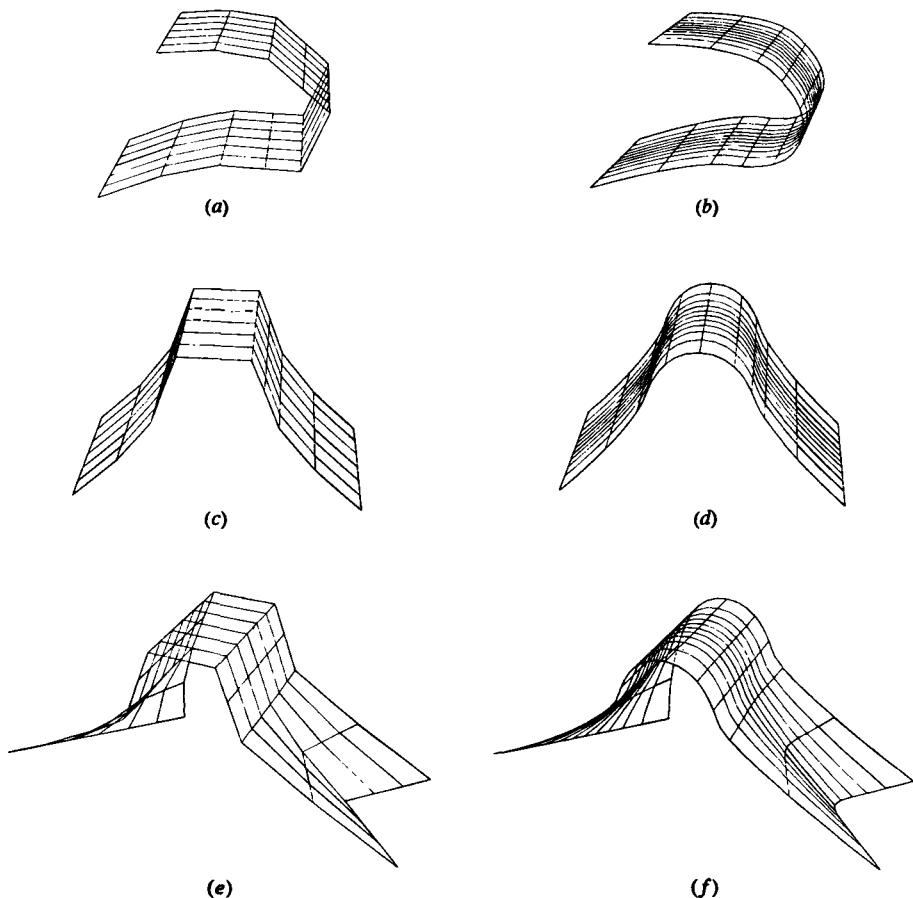


Рис. 6-60 Элемент квадратичной поверхности внутри более общей рациональной B-сплайн поверхности. Передняя кромка крыла. (a) Характеристический многогранник; (b) поверхность. Корма корабля. (c) Характеристический многогранник; (d) поверхность. Цилиндр как часть более общей поверхности. (e) Характеристический многогранник; (f) поверхность.

$$\begin{aligned}\bar{N}_w &= \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N_{i,k}(u) M'_{j,l}(w), \\ \bar{N}_{uw} &= \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N'_{i,k}(u) M'_{j,l}(w), \\ \bar{N}_{uu} &= \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N''_{i,k}(u) M_{j,l}(w), \\ \bar{N}_{ww} &= \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} B_{i,j} N_{i,k}(u) M''_{j,l}(w),\end{aligned}$$

$$\begin{aligned}\overline{D}_u &= \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N'_{i,k}(u) M'_{j,l}(w), \\ \overline{D}_w &= \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M'_{j,l}(w), \\ \overline{D}_{uw} &= \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N'_{i,k}(u) M'_{j,l}(w), \\ \overline{D}_{uu} &= \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N''_{i,k}(u) M_{j,l}(w), \\ \overline{D}_{ww} &= \sum_{i=1}^{n+1} \sum_{j=1}^{m+1} h_{i,j} N_{i,k}(u) M''_{j,l}(w).\end{aligned}$$

Штрих обозначает производную относительно соответствующего параметра. Функции $N'_{i,k}(u) M'_{j,l}(w)$, $N''_{i,k}(u)$, $M''_{j,l}(w)$ задаются уравнениями (5-97)–(5-100).

Эти производные полезны при определении гауссовой кривизны поверхности (см. разд. 6-15), а также других свойств поверхности.

6-17 ЛИТЕРАТУРА

- 6-1 Rogers, D. F., *Procedural Elements for Computer Graphics*, McGraw-Hill, New York, 1985. (См. [1-1] на с. 74.)
- 6-2 Bézier P. E., *Emploi des Machines à Commande Numérique*, Masson et Cie, Paris, France, 1970; Bézier, P. E., «Example of an Existing System in the Motor Industry: The Unisurf System,» *Proc. Roy. Soc. (London)*, Vol. A321, pp. 207–218, 1971.
- 6-3 Sabin, M. A., «An Existing System in the Aircraft Industry. The British Aircraft Corporation Numerical Master Geometry System,» *Proc. Roy. Soc. (London)*, Vol. A321, pp. 197–205, 1971.
- 6-4 Peters, C. J., «Interactive Computer Graphics Application of the Bicubic Parametric Surface to Engineering Design Problems,» McDonnell Douglas Automation Company, St. Louis, Missouri, presented at SIAM 1973 National Meeting, Hampton, Va., 18–21 June 1973.
- 6-5 Rogers D. F., and Satterfield S. D., «Dynamic B-Spline Surfaces,» *Proc. of the Fourth International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design (ICCAS 82)*, 7–10 June 1982, Annapolis, Maryland, pp. 189–196, North Holland, 1982.
- 6-6 Rogers, D. F., and Satterfield S. G., «B-Spline Surfaces for Ship Hull Design,» *Comp. Graph.*, Vol. 14, pp. 211–217, 1980, (SIGGRAPH 80).
- 6-7 Rogers, D. F., Rodriguez, F., and Satterfield, S. G., «Computer Aided Ship Design and the Numerically Controlled Production of Towing Tank Models,» *Proc. of 16th Des. Auto. Conf.*, San Diego, California, 24–27 June 1979.
- 6-8 Cohen, E., «Some Mathematical Tools for a Modeler's Workbench,» *IEEE Comp. Graph. & Appl.*, Vol. 3, pp. 63–66, October 1983.
- 6-9 Tan, S. T., Yuen, M. F., and Hui, K. C., «Modeling Solids with Sweep Primitives,» *Comp. Mech. Eng.*, Vol. 6, pp. 60–72, 1987.
- 6-10 Coquillart, S., «A Control-Point-Based Seeping Technique,» *IEEE Comp. Graph. & Appl.*, Vol. 7, pp. 36–45, November 1987.

- 6-11 Patton, D., Dept. of Mech. and Prod. Engr., Royal Melbourne Inst. of Tech., Melbourne, Australia, private communication, 1988.
- 6-12 Dresden, A., *Solid Analytic Geometry and Determinants*, Dover Publications, New York, 1964.
- 6-13 Sommerville, D. M. Y., *Analytical Geometry of Three Dimensions*, Cambridge Univ. Press, 1959.
- 6-14 Snyder, V., and Sisam, C. H., *Analytic Geometry of Space*, Henry Holt and Co., New York, 1914.
- 6-15 Goldman, R. N., «Two Approaches to a Computer Model for Quadric Surfaces,» *IEEE Comp. Graph. & Appl.*, Vol. 3, pp. 21–24, September 1983.
- 6-16 Wilson, P. R., «Conic Representation for Shape Description,» *IEEE Comp. Graph. & Appl.*, Vol. 7, pp. 23–30, 1987.
- 6-17 Miller, J. R., «Geometric Approaches to Non-planar Quadric Surfaces Intersection Curves,» *ACM Trans. on Graph.*, Vol 6, pp. 274–307, 1987.
- 6-18 Dill, J. C., «An Application of Color Graphics to the Display of Surface Curvature,» *Comp. Graph.*, Vol. 15, pp. 153–161, 1981, (SIGGRAPH 81).
- 6-19 Coons, S., «Surfaces for Computer-Aided Design of Space Forms,» M. I. T. Proj. MAC, MAC-TR-41, June 1967. (Also as AD 663 504).
- 6-20 Ferguson, J. C., «Multivariable Curve Interpolation,» *J. ACM*, Vol. 2, pp. 221–228, April 1964.
- 6-21 Barnhill, R. E., «Smooth Interpolation over Triangles,» in *Computer Aided Geometric Design*, Barnhill, R. E., and Riesenfeld, R. F. (eds.), Academic Press, New York, pp. 45–70, 1974.
- 6-22 Barnhill, R. E., «Coons Patches,» *Comp. in Ind.*, Vol. 3, pp. 37–43, 1982.
- 6-23 Charrot, P., and Gregory, J., «A Pentagonal Surface Patch for Computer Aided Geometric Design,» *Comp. Aid. Geom. Des.*, Vol. 1, pp. 87–94, 1984.
- 6-24 Faux, I. D., and Pratt, M. J., *Computational Geometry for Design and Manufacture*, Ellis Horwood (John Wiley & Sons), 1979.
- 6-25 Bézier, P. E., *The Mathematical Basis of the Unisurf CAD System*, Butterworth, London, 1986.
- 6-26 Barsky, B. A., and Greenberg, D. P., «Determining a Set of B-spline Control Vertices to Generate an Interpolating Surface,» *Comp. Graph. Imag. Proc.*, Vol. 14, pp. 203–226, 1980.
- 6-27 Rogers, D. F., and Fog, N. G., «Constrained B-spline Curve and Surface Fitting,» *CADJ*, Vol. 21, pp. 641–648, 1989.
- 6-28 Munchmeyer, F. C., Schubert, C., and Nowacki, H., «Interactive Design of Fair Hull Surfaces Using B-splines,» *Proc. of the Third International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design* (ICCAS 79), 18–21 June 1979, University of Strathclyde, Glasgow, Scotland, pp. 67–76, North Holland, 1979.
- 6-29 Munchmeyer, F. C., «The Gaussian Curvature of Coons Biquintic Patches,» *Proc. ASME Century 2 Inter. Comp. Tech. Conf.*, 12–15 August 1980.
- 6-30 Dill, J. C., «An Application of Color Graphics to the Display of Surface Curvature,» *Comp. Graph.*, Vol. 15, pp. 153–161, 1981, (SIGGRAPH 81).
- 6-31 Forrest, A. R., «On the Rendering of Surfaces,» *Comp. Graph.*, Vol. 13, pp. 253–259, 1979, (SIGGRAPH 79).
- 6-32 Dill, J. C., and Rogers, D. F., «Color Graphics and Ship Hull Surface Curvature,» *Proc. of the Fourth International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design* (ICCAS 82), pp. 197–205, North Holland, 1982.
- 6-33 Piegl, L., and Tiller, W., «Curve and Surface Constructions Using Rational B-splines,» *Comp. Aid. Des.*, Vol. 19, pp. 485–498, 1987.

Приложение А

ГРАФИЧЕСКОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

A-1 ВВЕДЕНИЕ

Разрабатывая концепции графического программного обеспечения, удобно мыслить в терминах схемы, показанной стрелками на рис. 1-63. Конвейер начинается с прикладной программы и соответствующей ей базы данных, проходит через пакет машинной графики и соответствующую ему графическую базу данных и далее через соответствующий программный интерфейс внешнего устройства, для генерации изображения на физическом графическом устройстве, с которым пользователь может или не может взаимодействовать. На протяжении такого конвейера существуют два критических программных интерфейса: интерфейс прикладной программы и интерфейс внешнего устройства. В настоящее время предпринимаются попытки привести оба интерфейса в соответствие с международными стандартами.

Интерфейс прикладного программиста обычно оформляется как пакет подпрограмм или функций, доступных из языка высокого уровня, такого, как Фортран, Си, Бейсик или Паскаль. Интерфейс пользователя обычно пишется с помощью подпрограмм и функций интерфейса прикладного программиста, который в настоящее время представлен четырьмя основными стандартами Графической корневой системой (the Graphical Kernel System GKS) [A-1–A-3], GKS-3D [A-4] и Иерархической интерактивной графической системой программиста (the Programmer's Hierarchical Interactive Graphics System PHIGS) [A-5, A-6] и PHIGS+ [A-7]. Многие программы и функции, предусмотренные для интерфейса прикладного программиста, являются основным предметом данной книги и работы [A-8]. Примеры: поворот, перенос, перспективные проекции, отсечения, удаление невидимых линий и невидимых поверхностей.

Интерфейс внешнего устройства работает на гораздо более низком уровне, чем интерфейс прикладного программиста. Это связано с генерацией кодов (двоичных битовых карт), требуемых для управления конкретным физическим или виртуальным графическим устройством. Обычно системные программисты и разработчики интерфейса прикладной программы имеют дело с интерфейсом устройства. Примеры стандартов интерфейса устройства: Графический метафайл компьютера (the Computer Graphics Metafile CGM) [A-9]–[A-11], Графический интерфейс компьютера (the Computer Graphics Interface CGI) [A-10]–[A-12] и Интерфейс виртуального устройства (the Virtual Device Interface VDI) [A-13].

Детальное обсуждение различных стандартов выходит за рамки данной книги и проводится в цитируемой литературе. Здесь дается только краткое общее описание.

GKS – двумерный стандарт визуализации, не включает моделирования, предусмотрены только двумерные примитивы, такие, как ломаная, поворот, перенос, масштабирование. Трехмерные преобразования должны выполняться за пределами стандарта, т. е. трехмерный объект преобразуется в двумерную проекцию (см. гл. 3) и результат отображается с использованием программ двумерной графики GKS. GKS поддерживает широкий спектр устройств ввода/вывода (см. гл. 1), таких, как монохроматические и цветные ЭЛТ-дисплеи, принтеры, графопостроители, мыши, планшеты, устройства для ввода скалярных значений, функциональные переключатели и т. д. Доступ к программам GKS обеспечен из ряда языков высокого уровня: Фортран, Паскаль, Ада, Си, GKS – официальный стандарт как Международной организации по Стандартам (ISO), так и Американского национального института стандартов (ANSI) [A-1].

GKS-3D – расширение GKS, обеспечивающее возможности отображения и определения трехмерных объектов. В нем также нет функций моделирования. Включает невидимые линии и невидимые поверхности, а также параллельные и перспективное проецирование. В настоящее время стандарт GKS-3D продолжает дорабатываться [A-4].

В отличие от GKS и GKS-3D, стандарт PHIGS был изначально разработан как двойной стандарт для трехмерного манипулирования и визуализации и для трехмерного геометрического моделирования. Особо выделяется поддержка высокоинтерактивного программного обеспечения, такого, как САПР/АСТПП, молекулярное моделирование, моделирование сплошных тел, а также имитация и управление. PHIGS позволяет строить, преобразовывать, модифицировать и хранить трехмерные геометрические модели. Стандарт PHIGS продолжает дорабатываться [A-5].

PHIGS+ – расширение PHIGS, включает модели освещения, обработку полутонов, глубину, а также кривые и поверхности, такие, как рациональные В-сплайны (см. разд. 5-13 и 6-16). PHIGS специально разработан применительно к возможностям высокоинтерактивных графических рабочих станций. Стандарт, например включает представление таких эффектов, как раскраска по методу Гуро и Фонга, а также новые геометрические атрибуты, например нормали вершины.

CGI обеспечивает функциональное преобразование независимой от аппаратуры информации, генерируемой стандартом интерфейса прикладного программиста, в аппаратно зависимые данные, требуемые для управления конкретным графическим устройством. CGI определяет виртуальный интерфейс устройства, т. е. идеализированное стандартное графическое устройство, которое поддерживает ввод от пользователя, генерирует графическую информацию и манипулирует ею. Существует строго регламентированное взаимодействие между GKS и CGI.

CGM перехватывает и хранит информацию на уровне CGI. Результатом является форматированный файл, содержащий «картинку», который может храниться на диске или ленте и передаваться по телефонным линиям или компьютерным сетям.

Прежде чем обсуждать каждый из стандартов во всех деталях и сравнивать их, полезно получить представление об их функциональной и концептуальной структуре.

А-2 ГРАФИЧЕСКИЕ ПРИМИТИВЫ МАШИННОЙ ГРАФИКИ

При создании любой системы графического программного обеспечения физическое (виртуальное) графическое устройство управляет первичными аппаратно зависимыми командами нижнего уровня (примитивами). Графический примитив выполняет единичное действие. Графические примитивы обычно распадаются на три группы: управление курсором, управление состоянием устройства и графический ввод. Здесь понятия курсора и состояния используются в наиболее общем смысле. Из этих графических примитивов формируются различные графические команды (множественные действия), называемые элементарными графическими функциями, они рассматриваются в разд. А-3. Графические элементарные функции аппаратно независимы.

Существуют два графических примитива управления курсором:

- переместить курсор – MOVE (x, y);
- включить/выключить курсор ONOFF (i).

Функции этих примитивов не нуждаются в комментариях.

Графические примитивы управления устройством более многочисленны и разнообразны. Они получили общее имя SET (установить). Некоторые примеры:

- установить алфавитно-цифровой режим;
- установить графический режим;
- установить режим абсолютных координат;
- установить режим относительных координат;
- установить цвет (красный, зеленый, синий);
- очистить экран
- и т. д.

В принципе существует только один графический примитив SET с различными аргументами. Перед тем как продолжить, кратко охарактеризуем функции каждого из этих различных состояний устройства.

Алфавитно-цифровой и графический режимы, как правило, относятся к устройствам, которые могут интерпретировать поток данных как в виде алфавитно-цифровых (ASCII) символов, так и в виде графических команд. Режимы абсолютных и относительных координат определяют, будут ли декартовы координаты, объявленные в графических примитивах, интерпретироваться как абсолютные или относительные. Команда установки цвета определяет цвет курсора для последующих команд, таких, как линия, точка и т. д. Команда очистки экрана перекрашивает экран дисплея в цвет фона.

Примитив графического ввода имеет общее имя GET (задать)

GET (значение).

Из сказанного ясно, что в принципе существуют только четыре графических примитива. В общем виде они могут быть представлены следующим образом:

ONOFF (управляющая переменная);
 MOVE (коорд., коорд.);
 SET (упр. переменная, {арг.1}, {арг.2}, ...);
 GET (значение).

Здесь угловые скобки $\langle \rangle$ используются для обозначения необязательных аргументов.

A-3 ЭЛЕМЕНТАРНЫЕ ГРАФИЧЕСКИЕ ФУНКЦИИ

Графические примитивы, которые обсуждались в разд. А-2, могут быть объединены в графические элементарные функции. Вообще говоря, это составные операции, т. е. они содержат более одного графического примитива. При аккуратной реализации они не зависят от конкретного физического устройства.

Так же как и графические примитивы, элементарные графические функции делятся на три группы: функции рисования, функции управления устройством или картинкой и функции управления интерактивным вводом. Функции рисования происходят от процесса рисования карандашом или перемещения карандаша по листу бумаги. Карандаш может быть перемещен над листом без рисования. Когда карандаш опускается на лист и перемещается, то он рисует линию. Карандаш может быть перемещен в некоторую позицию, в которой будет поставлена точка. Удобно иметь возможность задавать перемещения карандаша над бумагой и рисования им как в абсолютных так и в относительных координатах. Эти понятия проиллюстрированы шестью функциями рисования, или графическими элементарными функциями:

MOVE ABSOLUTE (x, y);
 MOVE RELATIVE (Δx , Δy);
 DRAW ABSOLUTE (x, y);
 DRAW RELATIVE (Δx , Δy);
 POINT ABSOLUTE (x, y);
 POINT RELATIVE (Δx , Δy).

Также нужна возможность вывести символ или текст на поверхности рисования, сказав:

TEXT (строка).

Всегда удобно предполагать, что действие начинается от текущей позиции курсора. Функционально каждая из этих графических элементарных функций может быть описана в терминах графических примитивов. Например, алгоритм MOVE ABSOLUTE можно представить следующим образом:

установить режим абсолютных координат;
 установить графический режим;

выключить курсор;
переместить курсор в указанную позицию.

Алгоритм POINT ABSOLUTE представляется в следующем виде:

Установить режим абсолютных координат.
Установить графический режим.
Выключить курсор.
Переместить курсор в указанную позицию.
Включить курсор.

Алгоритм DRAW ABSOLUTE:

Установить режим абсолютных координат.
Установить графический режим.
Включить курсор.
Переместить курсор в указанную позицию.

Относительные алгоритмы аналогичны приведенным, но с установкой режима относительных координат командой SET.

Элементарные функции управления устройством или картинкой инициируют устройство и начинают новый фрейм или картинку. Например, для этих целей могли бы служить функции:

Инициировать устройство.
Начать фрейм.

Функционально команда «Инициировать устройство» служит для установки предопределенного состояния устройства. Команда «Начать фрейм» служит для объявления регенерации графического устройства, при которой работа с текущей картинкой завершается, устройство заполняется цветом фона и начинается представление следующей картинки, очищается экран на запоминающей ЭЛТ, устанавливается новый чертеж для перьевых или струйных устройств и т. д.

Управление графическим вводом должно быть более сложным, чем простой графический примитив GET, обсуждавшийся в разд. А-2. Вот некоторые примеры элементарных функций графического ввода:

GET MOUSE (x, y).
GET VALUATOR (значение).
GET TABLET (x, y).
GET FUNCTION SWITCH (i).
и т. д.

Для выполнения этих функций графические примитивы можно объединить. Это иллюстрируют два примера алгоритмов. Первым является алгоритм GET VALUATER, который мог бы иметь следующий вид:

Установить режим абсолютных координат.
Установить графический режим.

Задать (значение).

Алгоритм команды GET MOUSE:

Установить режим абсолютных координат.

Установить графический режим.

Задать (значение x).

Задать (значение y).

Выше ничего не говорилось о типе графического устройства, пикселях и т.д. Читатель может удивиться, что не были включены команды SET PIXEL и GET PIXEL. Однако после минутного размышления он обнаружит, что в контексте графического устройства пиксел — это точка. Следовательно, команды для точек концептуально достаточны для команд пикселов.

Вообще говоря, элементарные графические функции — самый нижний уровень графических команд прикладного программиста. Во многих случаях прикладной программист ничего не знает о графических примитивах, и они могут быть ему фактически недоступны. Однако эти примитивы интересны для системного программиста. Если всю графическую систему представить в виде уровней или оболочек, то графические элементарные функции образуют первый уровень или первую оболочку, а графические примитивы — нулевой уровень или нулевую оболочку (нулевой, потому что они скрыты от прикладного программиста).

A-4 СИСТЕМЫ КООРДИНАТ

Для систем программного обеспечения машинной графики интерес представляют три координатные системы: мировые координаты, или координаты пользователя, псевдопространство с нормализованными координатами устройства и координаты устройства. Мировые координаты служат для описания прикладной области. Они обычно задаются в виде чисел с плавающей точкой. Координаты устройства используются для работы с конкретным графическим устройством: пиксели, адресуемые точки, дюймы, сантиметры и т. д. Они задаются целыми числами. Нормализованным пространством координат устройства называется псевдопространство, используемое для обеспечения независимости от устройства. Нормализованные координаты задаются числами с плавающей точкой из первого квадранта в интервалах $0 \leq ndcx \leq 1.0$, $0 \leq ndcy \leq 1.0$. Таким образом, начало координат находится в левом нижнем углу.

Как показано на рис. А-1 и обсуждалось в гл. 1, описание, манипулирование и отсечение базы данных происходит в мировых координатах. Сформированное таким образом в базе данных «окно» преобразуется в нормализованные координаты, а затем в координаты устройства. Размеры «окна» определяются в мировых координатах. Размеры экранного окна задаются в нормализованных координатах, чтобы обеспечить независимость от устройства. Оба преобразования из мировых в нормализованные координаты и из нормализованных координат в координаты устройства включают только простые перемещения и масштабирование¹.

¹ Предполагается, что преобразования перспективных проекций были выполнены перед отсечением.

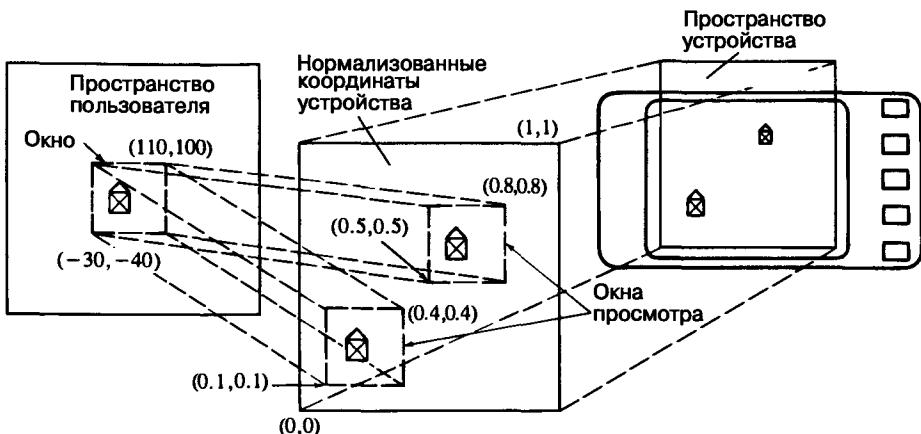


Рис. А-1 Преобразования в графическом конвейере.

Пример служит для иллюстрации различных систем координат и преобразований.

Пример А-1 Преобразование координат

Отобразить квадрат с центром в начале координат, со стороной 36 единиц в верхний левый квадрант устройства с разрешением 1024×1024 .

Чтобы задать некоторое рабочее пространство вокруг квадрата, определим окно со следующими левым, правым, верхним и нижним углами: $wleft = -20$, $wright = 20$, $wbottom = -20$, $wtop = 20$. Это действительно окно, а не изображение, которое определяет преобразование.

В нормализованных координатах верхний левый квадрант определяется так: $vleft = 0$, $vright = 0.5$, $vbottom = 0.5$, $vtop = 1.0$.

Преобразование из мировых координат в нормализованные требует масштабирования оконных координат и затем приведения результата к первому квадранту. Масштабные коэффициенты:

$$xwscale = \frac{1}{wright - wleft} = \frac{1}{20 - (-20)} = \frac{1}{40},$$

$$ywscale = \frac{1}{wtop - wbottom} = \frac{1}{20 - (-20)} = \frac{1}{40}.$$

Коэффициенты преобразования:

$$xwtran = \frac{-wleft}{wright - wleft} = \frac{-(-20)}{20 - (-20)} = \frac{1}{2}.$$

$$ywtran = \frac{-wbottom}{wtop - wbottom} = \frac{-(-20)}{20 - (-20)} = \frac{1}{2}.$$

Результирующее преобразование плоского окна в нормализованные координаты:

$$\begin{aligned}[T_w] &= [S_w][Tr_w] = \begin{bmatrix} xwscale & 0 & 0 \\ 0 & ywscale & 0 \\ xwtran & ywtran & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 1/40 & 0 & 0 \\ 0 & 1/40 & 0 \\ 1/2 & 1/2 & 1 \end{bmatrix}. \end{aligned}$$

Преобразование из нормализованных координат в координаты устройства вновь включает масштабирование и перенос. Когда координаты устройства описаны в целых числах, требуется особое внимание при определении масштабного и преобразующего множителей.

Хотя разрешение адресуемого в целых числах устройства составляет $xresolution \times yresolution$, адресуемый диапазон целых чисел расположен между 0 и $xresolution - 1$ и между 0 и $yresolution - 1$, т. е. если разрешающая способность равна 1024×1024 , то адресуемый интервал по обеим осям изменяется от 0 до 1023. Обозначив через $xmaxdev$ и $ymaxdev$ максимальные числа соответственно по осям x и y , получим масштабный множитель и величину переноса из нормализованных координат в координаты устройства:

$$\begin{aligned}xvscale &= \text{int}\{(vright - vleft)xmaxdev\} = \text{int}\{(0.5 - 0)(1023)\} = 511, \\yvscale &= \text{int}\{(vtop - vbottom)ymaxdev\} = \text{int}\{(1 - 0.5)(1023)\} = 511, \\xvtran &= \text{int}\{(vleft)(xmaxdev + 1)\} = \text{int}\{(0)(1024)\} = 0, \\yvtran &= \text{int}\{(vbottom)(ymaxdev + 1)\} = \text{int}\{(0.5)(1024)\} = 512.\end{aligned}$$

Результирующие преобразование нормализованных координат к координатам устройства:

$$[T_v] = [S_v][Tr_v] = \begin{bmatrix} xvscale & 0 & 0 \\ 0 & yvscale & 0 \\ xvtran & yvtran & 1 \end{bmatrix} = \begin{bmatrix} 511 & 0 & 0 \\ 0 & 511 & 0 \\ 0 & 512 & 1 \end{bmatrix}.$$

Использование этих результатов для координат углов квадрата с центром в начале координат дает:

$$\begin{bmatrix} -18 & -18 & 1 \\ 18 & -18 & 1 \\ 18 & 18 & 1 \\ -18 & 18 & 1 \end{bmatrix} \begin{bmatrix} 1/40 & 0 & 0 \\ 0 & 1/40 & 0 \\ 1/2 & 1/2 & 1 \end{bmatrix} = \begin{bmatrix} 1/20 & 1/20 & 1 \\ 19/20 & 1/20 & 1 \\ 19/20 & 19/20 & 1 \\ 1/20 & 19/20 & 1 \end{bmatrix}$$

и

$$\begin{bmatrix} 1/20 & 1/20 & 1 \\ 19/20 & 1/20 & 1 \\ 19/20 & 19/20 & 1 \\ 1/20 & 19/20 & 1 \end{bmatrix} \begin{bmatrix} 511 & 0 & 0 \\ 0 & 511 & 0 \\ 0 & 512 & 1 \end{bmatrix} = \begin{bmatrix} 25.55 & 537.55 & 1 \\ 485.45 & 537.55 & 1 \\ 485.45 & 997.45 & 1 \\ 25.55 & 997.45 & 1 \end{bmatrix} = \\ = \begin{bmatrix} 25 & 537 \\ 485 & 537 \\ 485 & 997 \\ 25 & 997 \end{bmatrix}.$$

Целочисленная функция использована здесь для получения результирующих целых адресуемых точек.

А-5 ПРОСТОЙ ГРАФИЧЕСКИЙ МЕТАФАЙЛ

К концепции хранения картинки в файле данных мы обращались при обсуждении графических стандартов. Применимым для этого графическим стандартом является Графический метафайл компьютера (CGM). Однако CGM более сложен и детализован, чем это требуется для простых разработок и студенческих работ. Простая альтернатива этому стандарту, называемая здесь стандартным дисплейным файлом, дана в таблице А-1¹.

Первая строка стандартного дисплейного файла служит для идентификации. Каждая последующая строка файла начинается с номера кода операции, за которым следует информация, требуемая для данной операции. Все значения координат определяются как числа с плавающей точкой в нормализованных координатах, т. е. в интервалах $0 \leq x \leq 1,0$, $0 \leq y \leq 1,0$. Все элементы данных разделены запятыми.

Простой стандартный дисплейный файл для квадрата из примера А-1 имеет вид:

```
0, 0.05, 0.05
2, 0.95, 0.05
2, 0.95, 0.05
2, 0.05, 0.95
2, 0.05, 0.05
```

Такой формат настолько прост, что позволяет создавать картинку с помощью текстового редактора. Обычно файл создается путем вызова графической процедуры, образованной из обращений к командам установки либо отмены, например:

```
Set sdf on/off filename.
```

Программы вывода сохраненной в метафайле картинки на некоторое графическое устройство достаточно элементарны. Такой формат допускает доступное расширение за счет включения локальной функциональности.

Возможности нормализованных координат и графического метафайла иллюстрирует рис. А-2. Изображения, включающие символы, были разработаны и отложены на устройстве с запоминающей трубкой. Результирующая картинка заполнена в стандартном дисплейном файле, который был передан компьютером, используемым как контроллер реального времени для обрабатывающего центра с ЧПУ (рис. 1-1). Программа читает стандартный дисплейный файл и генерирует код для драйвера обрабатывающего центра. Полученный результат показан на рис. А-2.

А-6 ПРОСТЫЕ ФОРМАТЫ ГРАФИЧЕСКОГО ВВОДА

В некоторых системах графического программного обеспечения необходимы стандарты форматов файла ввода. Для сложных или очень больших баз данных требуются формализованные структуры данных и систем управления. Однако для

¹ Подобная концепция используется более десятилетия в институте, где работают авторы.

Таблица А-1 Код стандартного дисплейного файла (.sdf)

Код	Операция	Параметры*
0	Абсолютное перемещение	x, y
1	Абсолютная точка	x, y
2	Абсолютное рисование	x, y
3	Относительное перемещение	Δx , Δy
4	Относительная точка	Δx , Δy
5	Относительное рисование	Δx , Δy
6	Текст	Строка текста
7	Установить цвет	красный, зеленый, синий
8	Установить таблицу просмотра	красные биты, зеленые биты, синие биты
9	Установить пиксель	x, y, красный, зеленый, синий
10	Новый фрейм	
11	Пауза	
12	Установить разрешение	xresolution, yresolution
99	Печать сообщения	Строка сообщения

* x, y – числа с плавающей точкой в NDC, т. е. в диапазоне $0 \leq x, y \leq 1.0$. Цвета красный, зеленый, синий являются числами с плавающей точкой в диапазоне $0 \leq$ красный, зеленый, синий ≤ 1.0 ; при этом равенство нулю означает отсутствие интенсивности, а равенство единице – полную интенсивность. Биты красного, зелено-го и синего цветов есть степени 2 (см. разд. 1-9, где обсуждаются соответствующие таблицы).

относительно небольших баз данных вполне достаточны простейшие форматы, основанные на файле с последовательным доступом. Приведем два таких формата.

Первый формат файла ввода, называемый «стандартный графический формат» (sgf) (табл. А.2), лучше всего подходит для простого рисования и для дву-и трехмерного каркасного представления изображения. Второй, определяемый в табл. А-3, называется «стандартный полигональный формат» и особенно полезен для растровых дисплеев или для описания входной информации для алгорит-мов невидимых линий и невидимых поверхностей. Заметим, что когда формат .spf используется для генерации каркасных изображений, каждая линия рису-



Рис. А-2 Гравюра, полученная из стандартного дисплейного файла.

Таблица А-2 Стандартный графический формат (.sgf)

Стандартный графический формат — это очень простой и общеупотребительный формат для описания любых дву- или трехмерных данных, которые могут быть представлены как список пар x, y или троек x, y, z . В файле допускаются пустые строки. Общая форма:

строка описания	строка описания
x_1, y_1	x_1, y_1, z_1
x_2, y_2	x_2, y_2, z_2
.	.
x_n, y_n	x_n, y_n, z_n
1e37, 1e37	1e37, 1e37, 1e37

Разделителем группы пар или троек является 1e37, 1e37 или 1e37, 1e37*. Последняя строка файла не должна быть разделителем.

Обычно этот формат интерпретируется следующим образом: графическое устройство, начиная с первой точки группы, рисует все входящие в эту группу точки и соединяющие линии, пока не встречает разделитель.

* Значение разделителя сложилось исторически.

ется дважды. В обоих случаях такие атрибуты представления, как цвет линии или многоугольника, коэффициент отражения многоугольника или тела и т. д., хранятся в отдельной группе файлов.

Оба формата заключают в себе уже готовый алгоритм рисования или черчения.¹ Рисование в соответствии с форматом .sgf:

В абсолютных координатах переместиться к первой точке и затем рисовать каждую следующую точку, пока не будет обнаружен разделитель.

Повторять до конца файла.

Алгоритм рисования для .spf формата приводится как часть описания многоугольника:

Переместиться в абсолютных координатах к точке, определенной первым указателем.

Соединять точки, определенные каждым следующим указателем в списке.

Соединить точку, определенную последним указателем списка, с точкой, определенной первым указателем, чтобы замкнуть многоугольник.

Вообще удобно иметь согласованное направление (по часовой стрелке или против) для спецификации вершин многоугольника.

A-7 ПРОСТЫЕ АЛГОРИТМЫ СОЕДИНЕНИЯ

Оба формата (.sgf и .spf) определяют простой и ограниченный алгоритм соединения или рисования. Практический интерес представляют два альтернативных, но

¹ И стандартный графический формат (.sgf), и стандартный формат многоугольника (.spf) более десятилетия используются в институте, где работают авторы.

Таблица А-3 Простой полигональный формат файла (.spf)

Прежде всего мы приведем общее описание формата, а затем подробный пример. Файл естественным образом делится на две части. Первая часть образует описание всех координатных векторов в изображении, вторая — описание характеристик многоугольника. В файле допускаются пустые строки. Полстрочное описание файла имеет следующую форму:

описательная (идентифицирующая) строка

число координатных векторов, многоугольников

список векторов x, y, z

.

.

список многоугольников

Каждая строка содержит два типа данных: число вершин многоугольника, за которым следует список вершин в виде соответствующих номеров строк из списка координатных векторов.

Характерный пример:

многоугольник с отверстием

14,2

4,4

4,26

20,26

28,18

28,4

21,4

21.8

10,8

10,4

10,12

10,20

17,20

21,16

21,12

9,1,2,3,4,5,6,7,8,9

5,10,11,12,13,14

Это простой многоугольник с отверстием.

Даны 14 координатных векторов или вершин многоугольника и 2 многоугольника.

Первый многоугольник содержит 9 вершин.

Второй многоугольник содержит 5 вершин.

Предполагается, что многоугольники замкнуты, т.е. первая точка не должна быть переопределена в качестве последней точки списка, чтобы замкнуть многоугольник.

отдельно описанных алгоритма рисования. Допустим, что база данных трехмерных координатных векторов представлена в виде матрицы $n \times 4$. Также определен одномерный управляющий массив $C(j)$, задающий алгоритм рисования следующим образом:

Значение $C(j)$ определяет номер строки в матрице координатных векторов.

Знак $C(j)$ используется для определения оператора рисования.

$C(j) < 0$ — перемещение в абсолютных координатах;

$C(j) > 0$ — рисование в абсолютных координатах.

Хотя такая схема достаточна, в некоторых отношениях она ограничена. Более гибкая и мощная схема использует двумерный управляющий массив $C(i, j)$. Алгоритм рисования определяется следующим образом:

значение $C(i, 1)$ служит для определения строки в матрице координатных векторов;

значение $C(i, 2)$ определяет оператор рисования (табл. А-1):

$C(i, 2) = 0$ — перемещение в абсолютных координатах;

$C(i, 2) = 1$ — перемещение в относительных координатах;

$C(i, 2) = 2$ — рисование в абсолютных координатах;

$C(i, 2) = 3$ — рисование в относительных координатах;

$C(i, 2) = 4$ — точка с абсолютными координатами;

$C(i, 2) = 5$ — точка с относительными координатами.

A-8 ПРОСТОЙ ГРАФИЧЕСКИЙ ПАКЕТ

В методике преподавания и изучения нового предмета, как правило, лучше начинать с простых понятий и тем. Приведение здесь минимального множества программ с использованием концепций, развивавшихся в этой книге и в работе [А-8], преследует цели обучения. Графический пакет растрового дисплея (RDGP) [А-14] представляет собой пример такого минимального множества графических программ. Программы разработаны для четырех языков: ANSI Бейсик, Фортран 77, Си, Паскаль. Описание каждой программы дано для всех четырех языков, сначала для ANSI Бейсика, потом для Фортрана 77, Си, Паскаля. Спецификации каждой программы отделены горизонтальными линиями. Типы переменных определены для каждого параметра каждой программы.

RDGP — ГРАФИЧЕСКИЙ ПАКЕТ РАСТРОВОГО ДИСПЛЕЯ¹

Программы естественным образом распадаются на три группы: управления состоянием дисплея, манипулирование пикселем и рисование линии. В таком порядке они и обсуждаются.

¹ Copyright © 1985 by David F. Rogers and Stephen D. Rogers. Пакет программ RDGP на TRUE Бейсик, Фортран 77 и Си для IBM PC распространяется автором. За информацией обращаться по адресу: Prof. D. F. Rogers, 817 Holly Dr. E., Rt. 10, Annapolis, MD 21401, USA.

имя:	<code>rdinit</code> (Инициализировать графический дисплей)
назначение:	Инициализировать графический пакет растрового дисплея и графическое устройство.
описание:	<pre>sub rdinit subroutine rdinit rdinit () procedure rdinit;</pre>
комментарии:	<code>rdinit</code> вызывается только один раз в начале программы.

имя:	<code>erase</code> (Очистить экран)
назначение:	Очистить графический экран указанным цветом.
описание:	<pre>sub erase (r, g, b) subroutine erase (r, g, b) real r, g, b erase (r, g, b) double r, g, b; procedure (r, g, b: real);</pre>
комментарии:	<code>r, g, b</code> — красная, зеленая и синяя составляющие цвета. Их значения задаются из интервала от 0 до 1. $(r, g, b) = (0, 0, 0)$ — черный и $(r, g, b) = (1., 1., 1.)$ — белый.

имя:	<code>setrsl</code> (Установить разрешение)
назначение:	Установить псевдоразрешающую способность растрового дисплея.
описание:	<pre>sub setrst (xres, yres) subroutine, setrsl (xres, yres) integer xres, yres setrsl (xres, yres) int xres, yres; procedure setrsl (xres, yres: integer);</pre>
комментарии:	<code>xres</code> — псевдоразрешающая способность раstra по оси x или по горизонтали. Она может изменяться от 1 до максимального значения физической разрешающей способности раstra по горизонтали. <code>yres</code> — псевдоразрешающая способность раstra по оси y или по вертикали. Может изменяться от 1 до максимального значения физической разрешающей способности раstra по вертикали. Пиксели располагаются в центре физического пространства дисплея. Пиксели имеют квадратную форму, насколько это позволяют ограничения физического дисплея. Программа <code>setrsl</code> вызывается только один раз в начале программы.

имя:	<code>setrgb</code> (Установить rgb)
назначение:	Установить текущий цвет для рисования.
описание:	<pre>sub, setrgb (r, g, b) subroutine setrgb (r, g, b) real r, g, b</pre>

```
setrgb (r, g, b)
double r, g, b;
procedure setrgb (r, g, b: real);
```

комментарии: r, g, b — красная, зеленая и синяя составляющие цвета. Их значения задаются из интервала от 0 до 1. (r, g, b)=(0., 0., 0.) — черный и (r, g, b)=(1., 1., 1.) — белый. Эта команда обычно используется, чтобы установить цвет для команд рисования ta, pa, da, mrg, pr, dr и text, данных ниже. Заметим, однако, что вызов команд setpxl и setscn может изменить цвет рисования.

имя:	setlut (Установить таблицу просмотра)
назначение:	Установить таблицу цветов просмотра.
описание:	<pre>sub setlut (table, rbits, gbits, bbits) subroutine setlut (table, rbits, gbits, bbits) character table (*) integer rbits, gbits, bbits setlut (table, rbits, gbits, bbits) rdString = lstring (255); procedure setlut (table: rdString; rbits, gbits, bbitg: integer);</pre>
комментарии:	<p>Таблица — это строка, которая определяет специфику установленной таблицы просмотра. Возможны следующие значения: по умолчанию (таблица просмотра отсутствует), серый и цветной.</p> <p>rbits, gbits и bbits — число бит, которые предполагается отвести под красную, зеленую и синюю составляющие буфера кадра. Число градаций каждого цвета 2^{bits}. Каждый из параметров rbits, gbits, bbits должен быть меньше или равен ширине таблицы цветов просмотра. Их сумма должна быть меньше числа физических бит, отведенных под буфер кадра. Общее число отображаемых в некоторый момент времени цветов, равное $2^{(rbits+gbits+bbits)}$, выйдет за допустимую палитру, значение которой равно 2^{3*w}, если $w \geq n$, где n — ширина таблицы цветов просмотра. В случае использования таблицы серых тонов просмотра значение градации серого цвета равно rbits. Максимальное число градаций серого цвета — это w.</p>

имя:	setdm (Установить режим дисплея)
назначение:	Управление режимом дисплея.
описание:	<pre>sub setdm (str \$) subroutine setdm (str) character str (*) setdm (str) char * str; rdString = lstring (255); procedure setdm (str: rdString);</pre>
Комментарии:	Предопределеными являются строки monitor (стандартное алфавитно-цифровое устройство) и graphics (стандартное графическое устройство).

имя:	setpxl (Установить пиксель)
назначение:	Установить единственный пиксель на линии развертки.
описание:	<pre>sub setpxl (x, y, r, g, b) integer x, y real r, g, b setpxl (x, y, r, g, b) int x, y; double r, g, b; procedure setpxl (x, y: integer; r, g, b: real);</pre>
комментарии:	Координаты x и y указывают левый нижний угол псевдопикселя, который был определен программой setrls. Пиксель покрывает пространство правее и выше позиции, определенной x и y. r, g, b — красная, зеленая и синяя составляющие цвета пикселя. Их значения задаются из интервала от 0 до 1. (r, g, b) = (0., 0., 0.) — черный и (r, g, b) = (1., 1., 1.) — белый.

имя:	setscn (Установить линию развертки)
назначение:	Установить группу пикселов на линии сканирования.
описание:	<pre>sub setscn (y, ra, ga, ba, xstart, xstop) subroutine setscn (y, ra, ga, ba, xstart, xstop) integer y, xstart, xstop real ra (*), ga (*), ba (*) setscn (y, ra, ga, ba, xstart, xstop) int y, xstart, xstop; double ra [], ga [], ba []; rgScanline = array [0... x max] of real; procedure setscn (y: integer; ra, ga, ba: rdScanline; xstart, xstop: integer);</pre>
комментарии:	у — устанавливаемая псевдолиния развертки. xstart — начальная позиция пикселов. xstop — позиция последнего пикселя. ra, ga, ba — массивы, содержащие красную, зеленую и синюю составляющие цветов пикселов. Значения элементов ga, ga, ba задаются из интервала от 0 до 1, где (0., 0., 0.) — черный и (1., 1., 1.) — белый. Первый элемент каждого массива используется для цвета пикселя, соответствующего xstart. Следующие пиксели линии развертки используют значения следующих элементов массивов.

имя:	getpxl (Задать пиксель)
назначение:	Дать характеристики единственного пикселя линии развертки.
описание:	<pre>sub getpxl (x, y, r, g, b) subroutine getpxl (x, y, r, g, b) integer x, y real r, g, b getpxl (x, y, r, g, b) int x, y; double *r, *g, *b; procedure getpxl (x, y: integer; var r, g, b: real);</pre>

комментарии: Координаты x и y указывают левый нижний угол псевдопикселя, который был определен программой setrl. Пиксел покрывает пространство правее и выше позиции, определенной x и y. r, g, b (для Си – указатели на них) – красная, зеленая и синяя составляющие цвета пикселя. Их значения находятся в интервале от 0 до 1. (r, g, b)=(0., 0., 0.) – черный и (r, g, b)=(1., 1., 1.) – белый.

имя: getscn (Дать линию развертки)
назначение: Дать характеристики группы пикселов линии развертки.
описание:

```
sub getscn (y, ra, ga, ba, xstart, xstop)
subroutine getscn (y, ra, ga, ba, xstart, xstop)
integer y, xstart, xstop
real ra (*), ga (*), ba (*)
getscn (y, ra, ga, ba, xstart, xstop)
int y, xstart, xstop;
double ra [], ga [], ba [];
rdScanline = array [0..x max] of real;
procedure getscn (y: integer; var ra, ga, ba: rdScanline; xstart, xstop: integer);
```

комментарии: y – псевдолиния развертки, информацию о которой требуется прочитать.
xstart – начальная позиция группы пикселов.
xstop – позиция последнего пикселя.
ga, ga, ba – массивы, которые получают значения красного, зеленого и синего компонентов цвета пикселов, о которых читается информация. Их значения находятся в интервале от 0 до 1. (ra, ga, ba)=(0., 0., 0.) – черный и (1., 1., 1.) – белый. Первый элемент каждого массива используется для цвета пикселя, соответствующего xstart. Следующие элементы массива для следующих пикселов линии развертки.

имя: text (Текст)
назначение: Отобразить текст.
описание:

```
sub text (str $)
subroutine text (str)
character str (*)
text (str)
char *str;
rdString = lstring (255);
procedure txt (str: rdString)
```

комментарии: Текст отображается, начиная с последней позиции курсора с использованием предопределенного размера символов. Позиция курсора указывает на левый нижний угол ячейки первого символа. Курсор остается в нижнем правом углу ячейки последнего символа строки.
Замечание. В Фортране 77 текстовая/символьная строка должна заканчиваться символом \$, т.е. HELLO\$ служит для отображения на экране HELLO.
Замечание. Так как text – это зарезервированное в Паскале слово, используется аббревиатура txt.

имя:	ma (Абсолютное перемещение)
назначение:	Переместить курсор в позицию с абсолютным значением x и y.
описание:	<pre>sub ma (x, y) subroutine ma (x, y) real x, y ma (x, y) double x, y; procedure ma (x, y: real);</pre>
комментарии:	x и y — действительные числа, x и y — абсолютные координаты для псевдорастра, установленного программой setrls. 0, 0 — это левый нижний угол псевдорастра.
имя:	mr (Относительное перемещение)
назначение:	Переместить курсор на dx, dy.
описание:	<pre>sub mr (dx, dy) subroutine mr (dx, dy) real dx, dy mr (dx, dy) double dx, dy; procedure mr (dx, dy: real);</pre>
комментарии:	dx и dy — действительные числа. Перемещение происходит относительно последней позиции курсора в размерах псевдорастра, установленного программой setrls.
имя:	da (Абсолютное рисование)
назначение:	Нарисовать тонкую линию в абсолютных координатах.
описание:	<pre>sub da (x, y) subroutine da (x, y) real x, y da (x, y) double x, y; procedure da (x, y: real);</pre>
комментарии:	x и y — действительные числа, заданные в размерах псевдорастра, установленного программой setrls. Тонкая (шириной в один физический пикセル) линия рисуется от текущей позиции курсора до позиции x, y. Значения x, y — это абсолютные координаты, причем начальная пара 0, 0 соответствует левому нижнему углу псевдорастра, заданного программой setrls. Для рисования линии используется текущий цвет, который устанавливается с помощью программы setrgb.

имя:	dr (Относительное рисование)
назначение:	Нарисовать тонкую линию, используя относительные координаты.
описание:	<pre>sub dr (dx, dy) subroutine dr (dx, dy) real dx, dy</pre>

```

dr (dx, dy)
double dx, dy;
procedure dr (dx, dy: real);

```

комментарии: dx и dy — действительные числа, заданные в размерах псевдорастра, установленного программой setrsl. Тонкая (ширина в один физический пиксел) линия рисуется от последней позиции курсора x0, y0 до позиции x0+dx, y0+dy. Значения dx и dy — это относительные координаты, где x0, y0 — последняя позиция курсора. Для рисования линии используется текущий цвет. Текущий цвет устанавливается с помощью программы setrgb.

имя: pa (Абсолютная точка)
назначение: Отобразить точку в абсолютных координатах.
описание:

```

subroutine pa (x, y)
real x, y
pa (x, y)
double x, y;
procedure pa (x, y: real);

```

комментарии Значения x и y — действительные числа, заданные в размерах псевдорастра, который установлен программой setrsl. Маленькая точка (размером в один физический пиксел) отображается в позиции x, y. Значения x, y представляют собой абсолютные координаты, причем пара 0, 0 соответствует левому нижнему углу псевдорастра, заданного программой setrsl. Для отображения точки используется текущий цвет. Текущий цвет устанавливается с помощью программы setrgb.

имя: pr (Относительная точка)
назначение: Отобразить точку, используя относительные координаты.
Описание:

```

sub pr (dx, dy)
subroutine pr (dx, dy)
real dx, dy
pr (dx, dy)
double dx, dy;
procedure pr (dx, dy: real);

```

комментарии: Значения dx и dy — это действительные числа, заданные в размерах псевдорастра, установленного программой setrsl. Маленькая точка (размером в один физический пиксел) отображается в позиции x0 + dx, y0 + dy. dx, dy — относительные координаты, а x0, y0 — последняя позиция курсора. Для отображения точки используется текущий цвет. Текущий цвет устанавливается с помощью программы setrgb.

A-9 ЛИТЕРАТУРА¹

- A-1 «Graphical Kernel System (GKS)», ISO 7942, 15 August 1985; also, ANSI X3.124-1985, 24 June 1985.
- A-2 Bono, P., and Herman, I. (eds.), *GKS Theory and Practice*, Springer-Verlag, Heidelberg, 1987.
- A-3 Enderle, G., Kansy, K., and Pfaff, G., *Computer Graphics Programming, GKS – The Graphics Standard*, Springer/Verlag, Heidelberg, 2nd ed., 1987.
- A-4 «Graphical Kernel System for Three Dimensions (GKS-3D)», ISO DIS 8805, July 1987.
- A-5 «Programmer's Hierarchical Interactive Graphics System (PHIGS)», ISO DIS 9592, December 1987; also, dpANS X3.144.
- A-6 Brown, M., and Heck, M., *Understanding PHIGS*, Megatek Corp., San Diego, 1985.
- A-7 Van Dam, A., «PHIGS+, Functional Description Revision 3.0», *Comp. Graph.*, Vol. 22, pp. 125–218, 1988.
- A-8 Rogers, D.F., *Procedural Elements for Computer Graphics*, McGraw-Hill, New York, 1985. (См. [1-1] на с. 74.)
- A-9 «Computer Graphics Metafile for the Storage and Transfer of Picture Description Information (CGM)», ISO 8632-Parts 1 through 4: 1987; also, ANSI/X3.122-1986.
- A-10 Arnold, D.B., and Bono, P.R., *CGM and CGI*, Springer-Verlag, Heidelberg, 1988.
- A-11 McKay, L., *CGI/CGM PRIMER*, Nova Graphics International, Austin, TX 1987.
- A-12 «Computer Graphics Interfacing Techniques for Dialogues with Graphical Devices (CGI)», ISO DP 9636, 6 December 1986; also, dpANS X3.161.
- A-13 International Organization for Standardization: Information Processing Systems—Computer Graphics—Interface Techniques for Dialogues with Graphical Devices. Baseline Document. ISO/TC97 N1511 (1985).
- A-14 Rogers, D.F., and Rogers, S.D., «A Raster Display Graphics Package for Education», *IEEE Comp. Graph. & Appl.*, Vol. 6, pp. 51–58, April 1986.

¹ Эти документы имеются в различных источниках:

Опубликованные ANSI-стандарты (а именно GKS и CGM) и ISO-документы (CGI, CGM, GKS, GKS-3D, PHIGS) можно получить в ANSI (1430 Broadway, New York, NY 10018). Помимо США ISO-документы можно получить из национальных организаций по стандартам (например, BSI в Великобритании, AFNOR во Франции, DIN в ФРГ и JIS в Японии).

Проекты ANSI-стандартов (CGI, GKS-3D и PHIGS) можно получить в секретариате X3 фирмы СВЕМА (311 First Street, Suite 500, Washington, DC 20001).

Документы IGES и PDES можно получить в фирме IGES (IGES Committee Chairman, Mr. Bradford Smith, National Bureau of Standards, Gaithersburg, ND 20760).

Приложение В

МАТРИЧНЫЕ ОПЕРАЦИИ

Это приложение содержит некоторые несложные правила из матричной алгебры. Матрица — это прямоугольный массив чисел, операции над которыми подчинены приведенным правилам.

B-1 ТЕРМИНОЛОГИЯ

Числа, которые составляют матрицу, называются ее элементами. Эти элементы образуют строки и столбцы внутри матрицы. Если число строк равно числу столбцов, матрица называется квадратной. Рассмотрим *квадратную* матрицу 4×4 :

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}.$$

Обозначим элементы матрицы через a_{ij} . Первый индекс — номер строки, второй — номер столбца. Таким образом, a_{34} — элемент из третьей строки, четвертого столбца. Матрица, состоящая из m строк и n столбцов, имеет размерность $m \times n$. Для квадратной матрицы выполняется равенство $m = n$. Элементы a_{11}, a_{22}, a_{33} и a_{44} при $i = j$ называются диагональными элементами. Сумма диагональных элементов квадратной матрицы называется ее следом.

Диагональной матрицей называют матрицу, в которой отличны от нуля только диагональные элементы, т. е. $a_{ij} = 0$, если $i \neq j$. Единичной матрицей называют матрицу, у которой $a_{ij} = 0$ для $i \neq j$ и $a_{ij} = 1$ для $i = j$. Пример единичной 3×3 матрицы:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Каждый элемент нулевой матрицы равен 0, т. е. $a_{ij} = 0$. Две матрицы равны, только когда каждый элемент первой матрицы равен соответствующему элементу второй матрицы, т. е. $a_{ij} = b_{ij}$.

Симметричной называется квадратная матрица, у которой $a_{ij} = a_{ji}$. Например,

$$\begin{bmatrix} 2 & 3 & 6 \\ 3 & 17 & 5 \\ 6 & 5 & 12 \end{bmatrix}.$$

Строка матрицы является $1 \times n$ -матрицей. Строки матрицы иногда называют векторами строк или просто векторами. Столбец матрицы является $n \times 1$ -матрицей. Столбцы матрицы иногда называют векторами столбцов или просто векторами.

Скаляр — это 1×1 -матрица.

B-2 СЛОЖЕНИЕ И ВЫЧИТАНИЕ

Если две матрицы имеют одинаковую размерность, то для них определены операции сложения и вычитания. Чтобы сложить или вычесть две матрицы, надо сложить или вычесть соответствующие элементы, т. е. $c_{ij} = a_{ij} + b_{ij}$ или $c_{ij} = a_{ij} - b_{ij}$ соответственно. Например:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} + \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} = \begin{bmatrix} 2 & 6 & 10 \\ 6 & 10 & 14 \\ 10 & 14 & 18 \end{bmatrix}$$

и

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} - \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} = \begin{bmatrix} 0 & -2 & -4 \\ 2 & 0 & -2 \\ 4 & 2 & 0 \end{bmatrix}.$$

Матрица $[B]$ называется отрицанием матрицы A , если $b_{ij} = -a_{ij}$.

B-3 УМНОЖЕНИЕ

Умножение матриц — это очень полезная операция для машинной графики. Рассмотрим матрицу размера $n \times m$ и вторую матрицу размера $r \times s$. Для умножения этих матриц необходимо, чтобы m было равно r . Это значит, что число столбцов первой матрицы должно быть равно числу строк второй матрицы. Правила умножения матриц лучше всего описывает следующий пример. Если матрица $[A]$ имеет размерность 4×3 , а $[B]$ — квадратная 3×3 -матрица, то произведение этих матриц определяется следующим образом:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \\ = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} & a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} & a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} \\ a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31} & a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32} & a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33} \\ a_{41}b_{11} + a_{42}b_{21} + a_{43}b_{31} & a_{41}b_{12} + a_{42}b_{22} + a_{43}b_{32} & a_{41}b_{13} + a_{42}b_{23} + a_{43}b_{33} \end{bmatrix}.$$

Иначе говоря, каждая строка в матрице $[A]$ поэлементно умножается на соответствующий столбец матрицы $[B]$, результаты складываются и записываются в качестве элемента результирующей матрицы $[C]$. Заметим, что умножение 4×3 -матрицы на 3×3 -матрицу, дает матрицу размерности 4×3 . Вообще говоря, в результате умножения $n \times m$ -матрицы на $r \times s$ -матрицу при $m = r$ мы получим $n \times s$ -матрицу. Рассмотрим численный пример:

$$\begin{bmatrix} 1 & 3 \\ 4 & 2 \\ 1 & 1 \\ 6 & 4 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 1+9 & 2+12 \\ 4+6 & 8+8 \\ 1+3 & 2+4 \\ 6+12 & 12+16 \\ 3+6 & 6+8 \end{bmatrix} = \begin{bmatrix} 10 & 14 \\ 10 & 16 \\ 4 & 6 \\ 18 & 28 \\ 9 & 14 \end{bmatrix}.$$

Операция умножения матриц не является коммутативной, т. е. в общем случае произведение $[A][B]$ не равно $[B][A]$. Таким образом, важен порядок (последовательность) умножения. Для операции умножения матриц верны первый и второй дистрибутивные законы, т. е. $A(B+C) = AB + AC$ и $(A+B)C = AC + BC$. Также выполняется ассоциативный закон $A(BC) = (AB)C$.

В-4 ОПРЕДЕЛИТЕЛЬ КВАДРАТНОЙ МАТРИЦЫ

Определитель квадратной матрицы $[A]$ обозначается $|A|$. В теории матриц определитель имеет много полезных свойств. Для наших целей достаточно проиллюстрировать метод вычисления определителя на примере квадратной 3×3 -матрицы. Для матриц

$$M = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

определитель равен

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

$$= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

т. е. определитель имеет единственное численное значение. В качестве примера подсчитаем определитель

$$\begin{vmatrix} 3 & 9 & 4 \\ 6 & 1 & 8 \\ 2 & 5 & 3 \end{vmatrix}.$$

В результате получим

$$3(3 - 40) - 9(18 - 16) + 4(30 - 2) = 3(-37) - 9(2) + 4(28) = -17.$$

B-5 ТРАНСПОНИРОВАНИЕ МАТРИЦЫ

Транспонированная матрица — это исходная матрица, в которой строки заменены на столбцы, т. е. $b_{ij} = a_{ji}$. Например,

$$\begin{bmatrix} 1 & 6 & 3 & 7 \\ 2 & 8 & 11 & 4 \\ 6 & 3 & 2 & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 2 & 6 \\ 6 & 8 & 3 \\ 3 & 11 & 2 \\ 7 & 4 & 1 \end{bmatrix},$$

где специальный символ T определяет операцию транспонирования. Заметим, что если исходная матрица имела размерность $m \times n$, то размерность транспонированной матрицы будет $n \times m$.

B-6 ОБРАЩЕНИЕ КВАДРАТНОЙ МАТРИЦЫ

В алгебре скаляров справедливо следующее утверждение, если $ax = y$, то $x = a^{-1}y$, где a^{-1} это просто обратная величина к a , т. е. $a^{-1} = 1/a$. В матричной алгебре операция деления не определена и обратной матрицы в смысле скалярной алгебры не существует, однако если $[A][X] = [Y]$, то $[X] = [A]^{-1}[Y]$, где $[A]^{-1}$ — матрица, обратная к квадратной матрице $[A]$.

Обратная матрица существует для квадратной и неединичной матрицы. Однако не каждая квадратная матрица имеет обратную. Но если определитель квадратной матрицы не равен нулю, то обратная матрица существует и единственна. Если обратная матрица существует, то $[A][A]^{-1} = [I]$, где $[I]$ — единичная матрица.

Рассмотрим произведение двух квадратных матриц, размерностью 3×3 .

$$\begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} 6 & -2 & -3 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Если произведение двух матриц $[A][B] = [I]$, то $[B] = [A]^{-1}$, т. е. $[B]$ — это матрица, обратная к $[A]$.

Существует много способов вычисления обратной матрицы. В большинстве языков программирования есть функции или подпрограммы, позволяющие вычислять значения элементов матрицы, обратной к данной. Описание таких алгоритмов выходит за рамки нашего краткого изложения.

Приложение С

ПСЕВДОКОД

Приведенный псевдокод призван помочь в понимании и реализации представленных в тексте алгоритмов. Он не задумывался как точный, синтаксически корректный и полный язык. Элементы псевдокода заимствованы из нескольких общепринятых языков программирования: Бейсика, Фортрана, Паскаля, Си и т. д. Псевдокод содержит структурные конструкции, например **if-then-else** и **while**. Для удобства включен оператор безусловного перехода **go to**. Оператор цикла **for-next** взят из Бейсика. Псевдокод включает также подпрограммные модули и матричные функции. Специальные функции и программы, такие, как **Min**, **Max**, **Push**, **Pop**, определены в алгоритмах в индивидуальном порядке.

Кратко охарактеризуем общие соглашения, используемые при описании алгоритмов. Все ключевые слова набираются жирным шрифтом строчными буквами. Все операторы внутри тела оператора **if-then-else** или цикла **for-next** выделяются отступом. Комментарии набираются курсивом и печатаются с таким же отступом, как и операторы, к которым они относятся. Многолитерные имена переменных, как правило, набираются строчными буквами. Однолитерные переменные могут представляться как прописной, так и строчной буквами. Функции выделяются жирным шрифтом и начинаются с прописной буквы. Далее следует детальное описание этих соглашений.

C-1 КОММЕНТАРИИ

Комментарии набираются курсивом. Они располагаются с тем же отступом, что и операторы, к которым они относятся. В начале алгоритма помещаются комментарии, кратко описывающие назначение этого алгоритма и определяющие используемые переменные.

C-2 КОНСТАНТЫ

Все константы являются десятичными числами, если в комментариях не оговорено иное. Например: 9, -3, 6.732, 1×10^{-9} , 3.

C-3 ПЕРЕМЕННЫЕ

Переменная — это имя, используемое для хранения значения. Это значение может изменяться. В длинных именах переменных используются, в основном, строчные

буквы; прописные используются в тексте книги лишь в тех случаях, когда это способствует лучшему пониманию программы. Однолитерные переменные могут набираться как строчной, так и прописной буквой. Для большей наглядности можно употреблять литеры с индексами. Примеры переменных: flag, P'_2 , x , y .

C-4 МАССИВЫ

Массив — имя для индексированной совокупности значений. Для имен массивов действуют те же соглашения, что и для переменных. Обращение ко всему массиву осуществляется по его имени, а к отдельным элементам — по имени массива, за которым в скобках следует список индексов. Примеры массивов: window, window(1, 3).

C-5 ОПЕРАТОР ПРИСВАИВАНИЯ

Знак «равно» используется для присвоения переменной, расположенной в левой части оператора, значения выражения из правой части этого оператора.

C-6 АРИФМЕТИЧЕСКИЕ ВЫРАЖЕНИЯ

Обычные арифметические операции умножения, деления, сложения, вычитания и возведения в степень обозначаются знаками *, /, +, -, ^.

C-7 ЛОГИЧЕСКИЕ ОПЕРАЦИИ И ОПЕРАЦИИ ОТНОШЕНИЯ

Логические операции **and** и **or** выделяются жирным шрифтом, как это здесь показано. Операции отношения «равно», «не равно», «меньше», «больше», «меньше или равно», «больше или равно» обозначаются =, ≠, <, >, ≤, ≥ соответственно. Эти операции используются для проверки условий. Результатом проверки являются логические значения «истина» или «ложь».

C-8 ОПЕРАТОР finish

Оператор **finish** используется для обозначения конца работы алгоритма.

C-9 ОПЕРАТОРЫ while И end while

Операторы, расположенные внутри блока **while-end while**, выполняются циклически, до тех пор, пока истинно некоторое условие. Условие проверяется в начале блока. Когда условие перестает быть истинным, выполнение программы продолжается с оператора, следующего за **end while**. Оператор **end while** используется для обозначения конца блока. Все операторы внутри блока **while-end while** располагаются с отступом. Общая форма оператора

```
while (условие)
  [операторы, которые необходимо выполнить]
end while
```

Пример

```
i = 0
while (i < 5)
    x = x + 5
    i = i + 1
end while
finish
```

C-10 ОПЕРАТОР if-then

Оператор **if-then** используется для выбора альтернативы пути выполнения программы либо для присвоения переменной альтернативного значения в зависимости от истинности некоторого условия.

Если аргументом **then** служит номер оператора и условие истинно, то управление передается на оператор с этим номером. В противном случае выполняется следующий по порядку оператор. Номера операторов являются их метками.

Если аргументом **then** служит оператор присваивания и условие истинно, то присваивание выполняется. В противном случае присваивание не выполняется и управление передается сразу на следующий оператор. Общая форма оператора:

```
if (условие) then (номер оператора)
if (условие) then (оператор присваивания)
```

Примеры

```
if (i < 10) then 3
if (i < 10) then x = x + 1
```

C-11 ОПЕРАТОРЫ if-then-else И end if

Оператор **if-then-else** используется для выбора одного из двух альтернативных блоков операторов в зависимости от истинности некоторого условия. Для обозначения конца блока оператора **if-then-else** используется оператор **end if**. Оператор не подразумевает повторения. Выполняется только один из альтернативных блоков. После этого выполнение программы продолжается с оператора, следующего за **end if**. Все операторы внутри блока **if-then-else** располагаются с отступом. Общая форма оператора

```
if (условие) then
    [операторы, выполняемые, если условие истинно]
else
    [операторы, выполняемые, если условие ложно]
end if
```

Пример

```
if (i ≥ 0) then
    x = x + 1
else
    x = x - 1
end if
```

Если оператор **if-then-else** записывается в одной строке, то оператор **end if** опускается. Заметим, что если опускается **else** и вторая группа операторов, то в результате получается блок оператора **if-then**.

C-12 ОПЕРАТОР **for-next**

Управление циклом можно организовать с помощью оператора **for-next**, как и с помощью оператора **while**. Циклическое выполнение операторов, расположенных внутри тела цикла **for-next**, происходит до тех пор, пока значение переменной цикла находится в заданных пределах. Все операторы внутри блока **for-next** располагаются с отступом. Общая форма оператора:

```
for (пер.цикла) = (нач.знач) to (кон.знач) step(шаг.цикла)
    [операторы, которые надо выполнить]
next (пер.цикла)
```

Если шаг цикла отсутствует, то он полагается равным единице. Шаг цикла может быть отрицательным. Начальное, конечное значение переменной цикла и шаг цикла могут быть переменными.

Пример

```
for x = 1 to n step a
    y = y + x
next x
```

C-13 ОПЕРАТОР **go to**

Оператор **go to** вызывает безусловный переход на оператор, определяемый аргументом. Общая форма оператора:

```
go to (номер оператора)
```

Номера операторов являются метками. Они располагаются в крайней левой позиции строки оператора.

C-14 ПОДПРОГРАММНЫЕ МОДУЛИ

Подпрограмма — отдельный программный модуль, который вызывается с помощью оператора **call**. Начало подпрограммы определяет оператор **subroutine**. Выход из подпрограммы определяется оператором **return**. После выхода из подпрограммы управление возвращается оператору, следующему за оператором **call** в вызывающей программе. Оператор **subroutine** содержит список входных и список выходных переменных. Связь между вызывающей программой и подпрограммным модулем осуществляется только через эти переменные. Все остальные переменные подпрограммного модуля локализованы в этом модуле. Общая форма операторов **call**, **subroutine** и **return**:

```
call имя (входные переменные; выходные переменные)
subroutine имя (входные переменные; выходные переменные)
return
```

Списки входных и выходных переменных операторов **call** и **subroutine** разделяются точкой с запятой и должны соответствовать друг другу. Имена подпрограмм состоят из строчных букв, набранных жирным шрифтом. Пример подпрограммного модуля:

```
subroutine check (x, y; flag)
  if x < y then
    flag = 0
  else
    flag = 1
  end if
return
```

С-15 ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ

Тригонометрические функции, такие, как **Sin** (θ), **Cos** (θ), **Tan** (θ) и обратные к ним **Arcsin** (x), **Arccos** (s), **Arctan** (x) принимают и возвращают значения в градусах, если установлен переключатель **option angle degrees**. Общепринятые гиперболические функции и их обратные функции **Sinh** (θ), **Cosh** (θ), **tanh** (θ), **Arcsinh** (x), **Arccosh** (x), **Arctanh** (x) также принимают и возвращают значения в градусах, если установлен этот переключатель. В противном случае они принимают и возвращают значения в радианах. Имена этих функций начинаются с прописной буквы и выделяются жирным шрифтом.

С-16 МАТРИЧНЫЕ ФУНКЦИИ

Матричные функции служат для выполнения типичных операций матричной алгебры, например сложения, вычитания, умножения, обращения и транспонирования матриц. Матричные функции манипулируют целыми массивами. Размерность каждого массива, используемого в качестве аргумента матричной функции, должна соответствовать выполняемой операции. Размерность результирующей матрицы также соответствует выполненной операции. Например, в результате умножения 3×2 -матрицы на 2×3 -матрицу получается 3×3 -матрица. Массив матрицы читается и записывается по строкам. Размерность матрицы может быть изменена во время выполнения операций. Массив матрицы может быть использован в обеих частях оператора присваивания. Имена матричных функций начинаются с прописной буквы и выделяются жирным шрифтом. Матричные функции:

сложение	Mat	$A = B + C$
вычитание	Mat	$A = B - C$
умножение	Mat	$A = B * C$
обращение	Mat	$A = \text{Inv}(B)$
транспонирование	Mat	$A = \text{Trn}(B)$
вычисление определителя	Значение	$= \text{Det}(B)$
единичная матрица	Mat	$A = \text{Idn}$
нулевая матрица	Mat	$A = \text{Zer}(n1, n2)$
постоянная матрица	Mat	$A = \text{Con}(n1)$
скалярное произведение	Значение	$= \text{Dot}(A, B)$

Функция **Zet** формирует массив матрицы размерностью $[n1 \times n2]$ и заполняет его нулями. Функция **Con** заполняет массив заранее заданной размерности константы $n1$.

C-17 СПЕЦИАЛЬНЫЕ ФУНКЦИИ

В алгоритмах, представленных в данной книге, определены разнообразные специальные функции. Имена этих функций выделяются жирным шрифтом и начинаются с прописной буквы. В качестве примера приведем функцию

Max(x_1, x_2),

результатом которой является большее из значений x_1 и x_2 .

Приложение D

ФОРМАТ ФАЙЛА В-СПЛАЙН ПОВЕРХНОСТИ

В связи с возникшим значительным интересом к В-сплайн поверхностям, разработан ряд значительных форматов файлов для обмена их описания. Еще один формат предложен в табл. D-1. Четырехмерные однородные координаты используются для обозначения определяемых точек прямоугольной сетки. Это позволяет объединить трехмерные физические позиции точек прямоугольной сетки и однородный координатный (весовой) коэффициент в общем описании. Предлагаемый формат, вообще говоря, достаточен для управления рациональными однородными и неоднородными, периодическими и открытыми В-сплайн поверхностями. Нерациональные сплайны поверхностей определяются установкой всех однородных координатных коэффициентов (весов) равными единице. Рациональные поверхности Безье определяются посредством однородных открытых связующих векторов в виде $[k \text{ нулей}, k \text{ единиц}]$ с подходящими однородными координатными коэффициентами. Нерациональные поверхности Безье определяются с помощью соответствующих связующих векторов и установкой всех однородных координатных коэффициентов (весов) равными единице. Составные поверхности определяются простым повторением полного описания для каждой поверхности. Заметим, что такой подход позволяет описывать различные типы В-сплайнов и поверхностей Безье в одном файле. Запятые должны быть простиавлены, как показано в таблице. В файле допустимы пробелы и пустые строки.

Таблица D-1 Формальное описание формата файла В-сплайн поверхности

Строка идентификации поверхности	например, номер корпуса (рис. 6-50)
нерациональная рациональная	рациональная, нерациональная в <i>u</i> , <i>w</i> соответственно
открытая, периодическая	открытая, периодическая, неоднородная в <i>u</i> , <i>w</i> соответственно
порядок базиса, порядок базиса	порядок в <i>u</i> , порядок в <i>w</i>
кол-во узлов сетки	кол-во узлов сетки в направлении <i>u</i>
кол-во узлов сетки	кол-во узлов сетки в направлении <i>w</i>
вектор связи в <i>u</i>	последовательность чисел 1, 2, 3, ...
вектор связи в <i>w</i>	последовательность чисел 1, 2, 3, ...
список 4D-точек сетки	массив $B(i, j)$ размерностью ((кол-во узлов сетки в направлении <i>u</i>) \times (кол-во узлов сетки в направлении <i>w</i>), 4). Элементы массива $B(i, j)$ располагаются с опережающим изменением индекса <i>j</i> , т.е. по строкам. Каждая строка содержит компоненты <i>x</i> , <i>y</i> , <i>z</i> , <i>h</i> , где <i>h</i> — однородный (весовой) коэффициент для каждой точки сетки.
	Замечание. Для нерациональных В-сплайнов все <i>h</i> = 1.

Пример файла

Пример поверхности (см. рис. 6-50)
нерац., нерац.

открыта, открыта

4, 4

8, 8

0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5

0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5

Поверхность, нерациональная в обоих направлениях.

В обоих направлениях используются функции открытого базиса.

Четвертого порядка в обоих направлениях.

Используется прямоугольная сетка 8×8 .

Открытый однородный связующий вектор для направления u .

Открытый однородный связующий вектор для направления w .

Точки сетки $B(i, j)$, расположенные по строкам.

0	,	62.7896	,	-14.2345	,	1	37.8884	,	50.5464	,	108	,	1
0	,	58.3546	,	-11.7746	,	1	33.861	,	47.4467	,	108	,	1
0	,	50.7321	,	-7.7246	,	1	27.1525	,	36.9685	,	108	,	1
0	,	37.2752	,	-0.7445	,	1	23.2976	,	19.6017	,	108	,	1
0	,	24.6114	,	0.2266	,	1	19.832	,	11.186	,	108	,	1
0	,	12.2841	,	-0.7321	,	1	12.7107	,	2.5924	,	108	,	1
0	,	0.9638	,	-4.2096	,	1	5.5457	,	-0.4150	,	108	,	1
0	,	-0.02351	,	-0.06207	,	1	-0.01007	,	-0.04313	,	108	,	1
13.2994	,	57.2518	,	0	,	1	41.0213	,	48.1632	,	162	,	1
8.6079	,	50.7918	,	0	,	1	39.6319	,	39.2661	,	162	,	1
7.2922	,	52.4931	,	0	,	1	38.6344	,	30.4192	,	162	,	1
-0.6518	,	33.3121	,	0	,	1	33.8422	,	16.8768	,	162	,	1
1.5411	,	14.3331	,	0	,	1	28.2826	,	6.96425	,	162	,	1
3.7183	,	13.4063	,	0	,	1	18.2607	,	-1.0973	,	162	,	1
7.87419	,	1.8224	,	0	,	1	4.4469	,	0.2564	,	162	,	1
0.9025	,	-0.4589	,	0	,	1	0.1344	,	0.01421	,	162	,	1
17.0482	,	57.6166	,	13.5	,	1	40.4748	,	47.8315	,	216	,	1
14.0925	,	55.5365	,	13.5	,	1	41.1403	,	43.5986	,	216	,	1
11.067	,	48.282	,	13.5	,	1	40.9666	,	30.6183	,	216	,	1
3.0433	,	40.1302	,	13.5	,	1	41.4111	,	14.9213	,	216	,	1
0.2289	,	21.5281	,	13.5	,	1	34.093	,	0.2635	,	216	,	1
3.5539	,	7.0775	,	13.5	,	1	16.9322	,	0.9311	,	216	,	1
3.0443	,	0.7293	,	13.5	,	1	5.55319	,	0.167384	,	216	,	1
-0.2092	,	0.08977	,	13.5	,	1	0.06208	,	-0.07885	,	216	,	1
29.2903	,	53.3467	,	54	,	1	40.536	,	47.9986	,	270	,	1
30.7639	,	54.1187	,	54	,	1	40.8065	,	41.4721	,	270	,	1
21.1234	,	48.1403	,	54	,	1	39.8169	,	33.6058	,	270	,	1
12.0645	,	35.461	,	54	,	1	41.8009	,	13.1977	,	270	,	1
9.0546	,	17.3787	,	54	,	1	37.0837	,	0.3062	,	270	,	1
8.6407	,	6.64932	,	54	,	1	13.1546	,	1.2999	,	270	,	1
5.1127	,	-0.7182	,	54	,	1	7.0730	,	-0.2726	,	270	,	1
0.1572	,	-0.03143	,	54	,	1	0.07475	,	-0.04795	,	270	,	1

Приложение Е

ЗАДАЧИ

Изучение компьютерной графики невозможно без ее практического использования, поэтому здесь приводится ряд задач и спецификаций программ. Задачи в основном направлены на то, чтобы подтвердить или развить концепции, представленные в тексте, или на то, чтобы накопить определенный опыт работы в соответствии с предлагаемой технологией. Как правило, задачи акцентируют внимание на каких-то отдельных конкретных аспектах материала. Задачи могут требовать, а могут и не требовать программирования.

Спецификации программ разработаны не только для иллюстрации концепций и технологий, представленных в книге, но также чтобы передать некоторый опыт создания достаточно сложных программ машинной графики. Хорошо реализованные, они могут составить инструментарий полезных графических программ. Результатом законченных программных проектов являются полезные программы общего назначения.

Это приложение содержит задачи. Спецификации программ приводятся в приложении F. Задачи и спецификации объединены в соответствии с главами книги.

ГЛАВА 2

2-1. Показать, что зеркальное отражение в плоскости относительно оси x , следующее за зеркальным отражением относительно прямой $y = -x$, эквивалентно полному повороту вокруг точки начала координат.

2-2. Для плоского прямоугольника, заданного координатными векторами своих вершин $P_1 [1 \ 1]$, $P_2 [3 \ 1]$, $P_3 [4 \ 2]$, $P_4 [2 \ 3]$, определить единственную матрицу преобразования, которая

- 1) зеркально отражает его относительно прямой $x = 0$;
- 2) осуществляет параллельный перенос на -1 в направлениях x и y ;
- 3) осуществляет поворот на 180° относительно точки начала координат.

Используя эти преобразования, определить новые координатные векторы вершин прямоугольника. Построить исходный и преобразованный прямоугольники на каком-либо графическом устройстве.

2-3. Показать, что 2×2 -матрица

$$[T] = \begin{bmatrix} \frac{1-t^2}{1+t^2} & \frac{2t}{1+t^2} \\ \frac{-2t}{1+t^2} & \frac{1+t^2}{1-t^2} \end{bmatrix}$$

осуществляет полный поворот.

2-4. Единичный квадрат преобразуется 2×2 -матрицей. Результирующие координатные векторы:

$$\begin{bmatrix} 0 & 0 \\ 2 & 3 \\ 8 & 4 \\ 6 & 1 \end{bmatrix}.$$

Найти матрицу преобразования.

2-5. Показать, что для $x = t^2$, $y = t$ результатом преобразования

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 0 & -2 & 2 \\ -2 & 2 & -2 \\ 1 & 0 & 1 \end{bmatrix}$$

является множество точек, образующих единичную окружность.

ГЛАВА 3

3-1. Показать, что для достижения некоторого эффекта преобразования объекта путем конкатенации некоторых матриц слева требуется, чтобы они были применимы в обратном порядке, когда используется конкатенация справа, т. е.

$$[Tr][Rx][Ry][Data] = [Data][Ry][Rx][Tr],$$

где $[Data]$ — матрица данных для объекта.

3-2. Вывести результирующую матрицу преобразования для перевода m , n , l в x , y , z соответственно; поворота вокруг оси y на угол φ ; поворота вокруг оси x на угол θ , следующих за преобразованием центрального перспективного проецирования на плоскость $z = 0$, с центром проекции на оси z в точке $z = k$. Сравнить результат с (3-63).

3-3. Изменить порядок выполнения поворотов в задаче 3-2. Сравнить результаты.

3-4. Для куба из примера 3-23 определить координатные векторы после преобразования, использующего конкатенацию матриц из задачи 3-2 при $m = l = n = 0$, $\varphi = 60^\circ$, $\theta = 30^\circ$ и $k = 2.5$. Сравнить полученный результат с примером 3-23 после построения обоих результатов рядом на одном листе графической бумаги.

3-5. Изменить порядок выполнения поворотов в задаче 3-4. Сравнить полученный результат с результатом задачи 3-4 и с примером 3-23 путем построения этих результатов на одном листе графической бумаги.

3-6. Получить матрицу преобразования размера 4×4 для объекта, заданного координатными векторами

$$\begin{bmatrix} 0 & 0 & 2 & 1 \\ 2 & 0 & 2 & 1 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 \\ 2 & 2 & 1 & 1 \end{bmatrix}$$

с алгоритмом соединения (разд. А-7)

$$C(j) = -1, 2, 3, 4, 5, 1, -6, 7, 8, 9, 6, -1, 6, -2, 7, -5, 9, -4, 10, 8, -10, 3$$

для вида из точки наблюдения (центр проекции), расположенной на 50 единиц правее, 60 единиц выше и выдвинутой на 150 единиц вперед от центра объекта. Помните, что мы смотрим в перспективу.

3-7. Повернуть куб с отсеченным углом (задача 3-6) на 30° относительно оси, проходящей через точки $P_1[-3 \quad -3 \quad 2]$ и $P_2[-3 \quad -2 \quad -2]$. Определить преобразованные координатные векторы. Нарисовать результат в диметрической проекции с коэффициентом искажения 0.625. Оба диметрических угла вращения считать положительными.

3-8. Показать, что любые комбинации полных поворотов и/или четное число зеркальных отражений могут быть представлены не более чем тремя поворотами вокруг координатных осей.

3-9. Продолжить анализ, приведенный в разд. 3-20 для случая, когда вектор зрения из центра проекции не перпендикулярен плоскости проекции.

ГЛАВА 4

4-1. Получить отношение между числом точек, радиусом и толерантностью (точностью приближения), в случае когда круг представляется вписанным в него многоугольником. Толерантность определяется как наибольшее расстояние между дугой и хордой, соединяющей две точки многоугольника на внешней окружности.

4-2. Кривая Ламе описывается формулой

$$\frac{x^n}{c_1} + \frac{y^n}{c_2} = 1.$$

Определить овал, проходящий через три точки $P_1[0 \quad 6]$, $P_2[5 \quad 5]$, $P_3[6 \quad 0]$. Определить точки пересечения овала и круга с радиусом 3 и центром в точке $x = 6$, $y = 0$.

4-3. Вычислить равное число точек на внешней дуге четверти круга в первом квадранте, используя непараметрический явно заданный (уравнение (4-1)), параметрический тригонометрический (уравнение (4-2)) и параметрический чебодящийся (уравнение (4-3)) анализ. Построить и сравнить результаты.

4-4. Получить уравнение (4-13).

4-5. Разработать и реализовать алгоритм для определения вершин параметрической параболы, заданной уравнением 4-13.

4-6. Доработать метод, предложенный в разд. 4-9, для случая круга, касающегося заданной прямой.

4-7. Доработать метод, предложенный в разд. 4-9, для случая прямой, касающейся круга заданного радиуса с центром в заданной позиции.

4-8. Доработать метод, предложенный в разд. 4-9, для случая окружности заданного радиуса, касающейся двух окружностей заданного радиуса с центрами в заданных позициях. Небольшая подсказка: существует 16 возможных решений. Построение дерева возможных вариантов решения зависит от радиуса касательной окружности и расстояния между двумя заданными окружностями.

4-9. Доработать метод, предложенный в разд. 4-9, для случая окружности заданного радиуса, одновременно касающейся прямой и окружности заданного радиуса с центром в заданной позиции. Небольшая подсказка: существует 8 возможных решений. Прямая может пересекать или не пересекать окружность.

4-10. Доработать метод, предложенный в разд. 4-9, для случая эллипса с заданными большой и малой полуосами, касающегося двух пересекающихся прямых. Какая дополнительная информация могла бы потребоваться?

4-11. Решить задачи 4-5–4-8, заменив все окружности эллипсами с заданными большими и малыми полуосами. В каждом случае определить, если это необходимо, какая дополнительная информация требуется.

4-12. Какой тип конического сектора представлен каждым из следующих уравнений:

$$\begin{aligned}x^2 + y^2 + 4x - 2y &= 10, \\x^2 + y^2 + xy - x + y &= 3, \\x^2 - 4xy + 4y^2 + 3x &= 6, \\2x^2 - y^2 + 4xy + 2x - 3y &= 6, \\x^2 - 4xy + 3y^2 + 5y &= 7, \\x^2 - y^2 &= 1.\end{aligned}$$

4-13. Преобразовать каждое из уравнений задачи 4-12 к стандартной форме.

4-14. Показать, что уравнение (4-33) эквивалентно следующему:

$$A(x + m)^2 + B(y + n)(x + m) + C(y + n)^2 + F(x + m) + G(y + n) + H = 0.$$

4-15. Показать, что следующее квадратное уравнение определяет параболу, и нарисовать сегмент между $x = 1$ и $x = 3$ для y_{\min} :

$$16x^2 - 24xy + 9y^2 - 60x - 80y + 20 = 0.$$

4-16. Первый квадрант единичного круга параметрически представляется в виде

$$x = \frac{(1 - \sqrt{2})t^2 + (\sqrt{2} - 2)t + 1}{(2 - \sqrt{2})t^2 + (\sqrt{2} - 2)t + 1},$$

$$y = \frac{(1 - \sqrt{2})t^2 + \sqrt{2}t}{(2 - \sqrt{2})t^2 + (\sqrt{2} - 2)t + 1}.$$

Определить значение y для $x = 0,5$. Указание: использовать метод итераций и сначала определить t для $x = 0,5$.

ГЛАВА 5

5-1. Сгенерировать параметрическую кривую, показанную на рис. 5-3.

5-2. Заданы четыре координатных вектора $P_1 [0 \ 0 \ 0]$, $P_2 [1 \ 1 \ 1]$, $P_3 [2 -1 -1]$, $P_4 [3 \ 0 \ 0]$, определить кусочно непрерывный кубический сплайн кривой, проходящей через них, используя аппроксимацию хордами для t_k . Касательные векторы на концах: $P'_1 [1 \ 1 \ 1]$ и $P'_4 [1 \ 1 \ 1]$. Вычислить 7 точек каждого сегмента сплайна. Сравнить результат для $\tau = 1/3, 2/3$ с примером 5-2, двумерным аналогом данной задачи на плоскости.

5-3. Используя координатные векторы из примера 5-2, вычислить и сравнить формы кривой для кубического сплайна для касательных векторов в точке P_1 для $P'_1 [\sqrt{2} \cos \theta, \sqrt{2} \sin \theta]$ при $\theta = 0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ$, сохраняя $P_2 [1 \ 1]$. Сравнить эти результаты с получаемыми в случае, когда оба касательных вектора — переменные $P'_1 [\sqrt{2} \cos \theta, \sqrt{2} \sin \theta]$ и $P'_2 [\sqrt{2} \cos \theta, \sqrt{2} \sin \theta]$. Заметим, что величина касательных векторов постоянна, изменяется только их направление.

5-4. Используя координатные векторы из примера 5-2, вычислить и сравнить формы кривой кубического сплайна для касательных векторов в точках P_1 и P_2 для $P'_1[a \ a]$ и $P'_2[a \ a]$ при $a = 1/4, 1/2, 1, 2, 4, 8$. Заметим, что направление касательных векторов постоянно, изменяется только их величина.

5-5. Используя координатные векторы из примера 5-2, вычислить и сравнить формы кривых кубического сплайна при условии незакрепленного конца. Сравнить результаты с теми, которые были получены в задаче 5-2 и в примере 5-2.

5-6. Заданы пять координатных векторов на плоскости

$$[P] = \begin{bmatrix} 0 & 0 \\ 0.5 & 0.6 \\ 1.0 & 1.0 \\ 1.5 & 0.8 \\ 1.0 & 0 \end{bmatrix}$$

Положив касательные векторы концов равными $P'_1 [1 \ 1]$ и $P'_2 [0 \ -1]$, определить $P(0.5)$ для каждого из четырех сегментов кубического сплайна, используя нормализованную аппроксимацию, т. е. приняв $0 \leq t \leq 1$.

5-7. Повторить задачу 5-2, используя нормализованную аппроксимацию для t_k , т. е. приняв $t_k = 1.0$. Сравнить результат с результатами, которые были получены в задаче 5-2, и с примером 5-2, аналогом этой задачи для плоскости.

5-8. Определить кубический сплайн кривой через пять координатных векторов $P_1[1 \ \pi/4 \ 0], P_2[1 \ 3\pi/4 \ 1], P_3[1 \ 5\pi/4 \ 2], P_4[1 \ 7\pi/4 \ 3], P_5[1 \ \pi/4 \ 4]$, где компоненты координатных векторов заданы в цилиндрических координатах, т. е. $P_i[r \ \theta \ z]$. Использовать условия незакрепленного конца.

5-9. Вычислить параболически смешанную кривую, проходящую через точки, данные в задаче 5-2. Использовать технологию, описанную в разд. 5-6. Сравнить результаты с результатами задачи 5-2 и с примером 5-2.

5-10. Вывести матрицу $[A]$ (см. равенство (5-60)) для параболически смешанной кривой, используя длины хорд как максимальные значения параметров, т. е. полагая

$$\begin{aligned} p(r) &= [r^2 \ r \ 1][B], & 0 \leq r \leq d_{13}, \\ q(s) &= [s^2 \ s \ 1][D], & 0 \leq s \leq d_{24}, \\ C(t) &= [t^3 \ t^2 \ t \ 1][A][G], & 0 \leq t \leq d_{23}, \end{aligned}$$

где d_{12}, d_{23}, d_{34} — длины хорд для P_1P_2, P_2P_3, P_3P_4 и $d_{13} = d_{12} + d_{23}, d_{24} = d_{23} + d_{34}$.

Заметим, что это не совсем точно соответствует результату Оверхаузера.

5-11. Вывести результат Оверхаузера, используя матричные методы из разд. 5-7.

5-12. Определить кривую Безье, используя координатные векторы, заданные в примере 5-2, как определение многоугольника. Вычислить результат для следующих значений параметров: 0, 1/3, 2/3, 1.

5-13. Вычислить координатный вектор P_3 , задающий касательный вектор, проходящий вдоль кривой, соединяющей два сегмента кривой Безье, иначе определяемых как $P_1[0 \ 0 \ 0], P_2[2 \ 2 \ -2], P_4[4 \ 0 \ 0]$ и $Q_1[4 \ 0 \ 0], Q_2[6 \ -2 \ 1], Q_3[8 \ -3 \ 2], Q_4[10 \ 0 \ 1]$.

5-14. Для поверхности Безье, заданной вершинами многоугольника из примера 5-7, найти определяющие вершины многоугольника, требуемые для генерации поверхности Безье шестого порядка, эквивалентной поверхности четвертого порядка из примера 5-7. Порядок наращивать последовательными шагами.

5-15. Найти определяющие вершины многоугольника, которые разбивают поверхность Безье, определенную в примере 5-7, на две кубические поверхности Безье.

5-16. Применить пошагово технологию разбиения к поверхности Безье и определяющим ее вершинам многоугольника из примера 5-7 и показать на этом примере, что определяющий многоугольник и поверхность сходятся.

5-17. Повторить пример 5-11 для узлового вектора $[0 \ 0 \ 0 \ 2 \ 2 \ 3 \ 3 \ 3]$. Результат показан на рис. 5-38e. Сравнить этот результат с примером 5-11 и с рис. 5-38d.

5-18. Определить алгебраические отношения для функций неоднородного базиса, полученного с помощью узловых векторов $[0 \ 0 \ 0 \ 0.4 \ 2.6 \ 3 \ 3 \ 3]$ и $[0 \ 0 \ 0 \ 1.8 \ 2.2 \ 3 \ 3 \ 3]$. Результаты показаны на рис. 5-38b и с. Сравнить эти результаты с рисунками 5-38d и 5-38e.

5-19. Проиллюстрировать зависимость базисных функций кривых В-сплайна четвертого и пятого порядков, определенных шестью вершинами многоугольника. Использовать технологию, подобную рассмотренной в разд. 5-9.

5-20. Определить В-сплайн кривой четвертого порядка, используя координатные векторы примера 5-2 как определяющий многоугольник. Сравнить результат с поверхностью Безье, полученной в задаче 5-12.

5-21. Определить В-сплайн кривой третьего порядка, используя координатные векторы примера 5-2 как определяющий многоугольник. Использовать параметрические значения $0, 1/2, 1, 3/2, 2$.

5-22. Для определяющего многоугольника, заданного в примере 5-13, определить псевдовершины, необходимые для совпадения конечных точек периодического В-сплайна с B_1 и B_4 соответственно. Сравнить результат с рис. 5-54.

5-23. Для определяющего многоугольника из примера 5-13 определить псевдовершины, необходимые, чтобы сделать касательные векторы концов кривой равными $[-1 \ 1]$ и $[1 \ -1]$ соответственно. Вычислить также начальную и конечную точки кривой. Результаты сравнить с рис. 5-54.

5-24. Используя аппроксимацию хордами, обсуждавшуюся в разд. 5-9, найти открытый В-сплайн поверхности четвертого порядка для определяющего многоугольника, заданного в примере 5-14. Сравнить результат с примером 5-14.

5-25. Используя аппроксимацию хордами, обсуждавшуюся в разд. 5-9, найти периодический В-сплайн поверхности третьего порядка для определяющего многоугольника, заданного в примере 5-14. Сравнить результат с примером 5-14.

5-26. Используя аппроксимацию хордами, обсуждавшуюся в разд. 5-9, найти периодический В-сплайн поверхности четвертого порядка для определяющего многоугольника, заданного в примере 5-14. Сравнить результат с примером 5-14 и результатом задачи 5-24.

5-27. Показать, что две кривые В-сплайна, заданные в примере 5-18 вершинами многоугольника $B_1[0 \ 0], B_2[1 \ 1], B_3[2 \ 1], B_4[3 \ 0]$ с альтернативными узловыми векторами $[X'] = [0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2]$ и $[X] = [0 \ 0 \ 0 \ 2 \ 4 \ 4 \ 4]$, идентичны. Указание: вычислите число точек на первой кривой $0 \leq t' \leq 2$ и на второй кривой $0 \leq t = 2t' \leq 4$ и сравните результаты.

5-28. Показать, что кривые В-сплайна разбиения из примера 5-18 такие же, как и исходная кривая. Указание: вычислите несколько точек двух кривых и постройте их на такой же шкале.

5-29. Показать, что кривые В-сплайна разбиения из примера 5-19 такие же, как и исходная кривая. Указание: вычислите несколько точек двух кривых и постройте их на такой же шкале.

5-30. Показать, что если после разбиения исходной кривой из примера 5-19 новую вершину многоугольника C_3 переместить в позицию вершины C_2 , то результирующая поверхность будет иметь угловую форму в точке $C_2 = C_3$.

5-31. Для лучшего понимания сущности функции оценки вектора, такой, как $P(t) = [x(t) \ y(t)]$, построить графики функций $x(t)$ от t , $y(t)$ от t и $y(t)$ от $x(t)$ для В-сплайна кривой четвертого порядка ($k = 4$), определенной $B_1[0 \ 0], B_2[2 \ 2], B_3[4 \ 2], B_4[6 \ 0]$ с открытым узловым вектором $[0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1]$. Решить эту же задачу для В-сплайна кривой третьего порядка ($k = 3$) с открытым узловым вектором $[0 \ 0 \ 0 \ 1 \ 2 \ 2 \ 2]$.

5-32. Вывести аналитическое выражение для второй производной вдоль кривой рационального В-сплайна. Оценить результат для $t = 0$ и $t = n + k - 2$.

5-33. Сгенерировать кривые рационального В-сплайна для определяющих вершин многоугольника $B_1[0 \ 0]$, $B_2[1 \ 2]$, $B_3[2.5 \ 0]$, $B_4[2.5 \ 0]$, $B_5[2.5 \ 0]$, $B_6[4 \ 2]$, $B_7[5 \ 0]$, при $[H] = [1 \ 1 \ h_3 \ h_4 \ h_5 \ 1 \ 1]$ для значений $h_3 = h_4 = h_5 = 0, 0.25, 1, 5$. Сравнить результаты с рис. 5-66.

5-34. Рассмотреть пример 5-20 при $k = 4$.

5-35. Найти первую и вторую производные при $t = 0$ и $t = 3$ для определяющего многоугольника из примера 5-20.

5-36. Найти первую и вторую производные при $t = 0$ и $t = 2$ для $k = 4$ для определяющего многоугольника из примера 5-20.

5-37. Сгенерировать тор как поверхность рационального В-сплайна, полученную вращением окружности радиуса 1 с центром в точке $x = 5$, $z = 1$ относительно оси z (рис. 6-58).

5-38. Сгенерировать сферу радиуса 1 как поверхность рационального В-сплайна, полученную вращением полуокружности радиуса 1 с центром в точке $x = 0$, $z = 1$ относительно оси z (рис. 6-59).

ГЛАВА 6

6-1. Сгенерировать поверхности, показанные на рис. 6-6.

6-2. Сгенерировать поверхности вращения для кривых, вычисленных в задачах 5-6 и 5-9.

6-3. Сгенерировать «кружку», показанную на рис. 6-10.

6-4. Добавить точку к «кружке», показанной на рис. 6-10, используя изогнутую поверхность.

6-5. Сгенерировать изогнутую поверхность, показанную на рис. 6-16.

6-6. Определить тип квадратичной поверхности, заданной каждым из этих уравнений:

$$\begin{aligned} 2x^2 - 2yz + 2zx - 2xy - x - 2y + 3z - 2 &= 0, \\ x^2 + z^2 + 2xy + 2zx - 2yz - 2x + 4y - 4 &= 0, \\ x^2 + 4y^2 + z^2 - 4yz - 12zx + 24xy + 4x + 16y - 26z - 3 &= 0, \\ 4x^2 + y^2 + z^2 - 4xy + 6z + 8 &= 0, \\ 4y^2 + 4z^2 + 4xy - 2x - 14y - 22z + 33 &= 0. \end{aligned}$$

6-7. Используя параметрическое представление, сгенерировать гиперболический параболоид.

6-8. Нарисовать поверхности, составляющие отдельные компоненты, т. е. $Q(u, v, x)$, $Q(u, v, y)$, $Q(u, v, z)$ (рис. 6-24) и полную поверхность $Q(x, y, z)$ для точки, прямой и плоскости.

6-9. Для линейчатой поверхности с ребрами в виде кривых, заданных как

$$\begin{aligned} z &= 0, \quad y = 0.25 \cos(2\pi x), \quad 0 \leq x \leq 1, \\ z &= 1, \quad y = 0.25 \sin(2\pi x), \quad 0 \leq x \leq 1, \end{aligned}$$

определить поверхность в точке $Q(0.5, 0.5)$.

6-10. Построить билинейную поверхность, соответствующую четырем точкам $P(0,0) = [0.25 \quad 0]$, $P(1,0) = [0.75 \quad 0]$, $P(0,1) = [0.75 \quad 0.9]$, $P(1,1) = [0.25 \quad 0.8]$.

6-11. Построить билинейную поверхность, которая соответствует четырем точкам, определяемым противоположными концами диаметров двух единичных окружностей, лежащих на параллельных плоскостях, отстоящих друг от друга на единицу и поворачиваемых на 180° с интервалом 15° .

6-12. Сгенерировать полную линейчатую поверхность, показанную на рис. 6-28 и обсуждавшуюся в примере 6-10. Является ли такая поверхность развернутой?

6-13. Сгенерировать полную линейчатую поверхность, показанную на рис. 6-30 и обсуждавшуюся в примере 6-12. Показать, что угловая часть этой поверхности развернута.

6-14. Показать на примере, что плоская бикубическая поверхность Кунса получается в том случае, когда координатные, касательные векторы и векторы вращения лежат в одной плоскости.

6-15. Найти матрицу $[P]$, которая задает кусок цилиндрической бикубической поверхности Кунса.

6-16. Найти матрицу $[P]$, которая задает кусок общей бикубической поверхности Кунса.

6-17. Определить точку на поверхности с координатами $u = w = 0.5$ для бикубической поверхности Кунса со следующими координатными векторами: $P(0,0) = [-100 \quad 0 \quad 100]$, $P(0,1) = [100 \quad -100 \quad 100]$, $P(1,0) = [-100 \quad 0 \quad -100]$, $P(1,1) = [-100 \quad -100 \quad -100]$, u -касательные векторы $P^{1,0}(0,0) = [10 \quad 10 \quad 0]$, $P^{1,0}(0,1) = [-1 \quad -1 \quad 0]$, $P^{1,0}(1,0) = [-1 \quad 1 \quad 0]$, $P^{1,0}(1,1) = [1 \quad -1 \quad 0]$, w -касательные векторы $P^{0,1}(0,0) = [0 \quad -10 \quad -10]$, $P^{0,1}(0,1) = [0 \quad 1 \quad -1]$, $P^{0,1}(1,0) = [0 \quad 1 \quad 1]$, $P^{0,1}(1,1) = [0 \quad 1 \quad 1]$, и векторы вращения $P^{1,1}(0,0) = [0 \quad 0 \quad 0]$, $P^{1,1}(0,1) = [0.1 \quad 0.1 \quad 0.1]$, $P^{1,1}(1,0) = [0 \quad 0 \quad 0]$, $P^{1,1}(1,1) = [-0.1 \quad -0.1 \quad -0.1]$. Систематически изменить векторы вращения и касательные векторы и сравнить результаты.

6-18. Продолжить анализ параболически смешанной кривой на случай параболической смешанной поверхности [5-10]. Указание: использовать параболически смешанные кривые для четырех граничных кривых и стиль задания линейной поверхности Кунса со смешанными функциями, заданными как $F_1(t) = 1 - 3t^2 + 2t^2$, $F_2(t) = 3t^2 - 2t^3$. Заметим, что результирующая поверхность контролируется четырьмя угловыми точками и двумя присоединенными векторами в концах граничных кривых. Что является продолжением отдельных кусков? Какой порядок продолжения может поддерживаться через границы кусков? Каково продолжение влияния перемещения единственной точки поверхности, построенной из множества кусков?

6-19. Показать для бикубической поверхности Безье 4×4 , что для $u = w = 0$

$$Q_{uw}(0,0) = 9[(B_{1,1} - B_{0,1}) - (B_{1,0} - B_{0,0})].$$

Сравнить результат с уравнением (6-60).

6-20. Найти (4×4) определяющие вершины прямоугольной сетки Безье, которые задают кусок бикубической поверхности Кунса из примера 6-13.

6-21. Сгенерировать поверхность Безье из примера 6-14, не используя матричных методов.

6-22. Показать, что ни гиперболический параболоид, ни параболический лист не являются развертывающимися поверхностями.

6-23. Для поверхности из примера 6-12 показать, что кривизна Гаусса равна нулю для плоской области и положительна в остальной области.

6-24. Определить кривизну Гаусса при $u = w = 0.5$ для поверхности из примера 6-14, показанной на рис. 6-39a-d. Сравнить результаты.

6-25. Сгенерировать закрытые периодические поверхности В-сплайна, показанные на рис. 6-46.

6-26. Сгенерировать комбинированную открытую и периодическую поверхность В-сплайна, показанную на рис. 6-47.

6-27. Сгенерировать закрытую периодическую тороидальную поверхность В-сплайна, показанную на рис. 6-48.

6-28. Рассчитать поверхность В-сплайна четвертого порядка для определяющей прямоугольной сетки 4×5 , заданной так:

$$\begin{array}{lll}
 B_{1,1}[0 \ 0 \ 100] & B_{2,1}[25 \ 0 \ 150] & B_{3,1}[50 \ 0 \ 100] \\
 & B_{4,1}[75 \ 0 \ 50] & B_{5,1}[100 \ 0 \ 100] \\
 B_{1,2}[0 \ 33 \ 150] & B_{2,2}[25 \ 33 \ 200] & B_{3,2}[50 \ 33 \ 100] \\
 & B_{4,2}[75 \ 33 \ 50] & B_{5,2}[100 \ 33 \ 50] \\
 B_{1,3}[0 \ 66 \ 50] & B_{2,3}[25 \ 66 \ 25] & B_{3,3}[50 \ 66 \ 100] \\
 & B_{4,3}[75 \ 66 \ 150] & B_{5,3}[100 \ 66 \ 150] \\
 B_{1,4}[0 \ 100 \ 100] & B_{2,4}[25 \ 100 \ 50] & B_{3,4}[50 \ 100 \ 100] \\
 & B_{4,4}[75 \ 100 \ 150] & B_{5,4}[100 \ 100 \ 100]
 \end{array}$$

Определить вычислением вручную точку в центре параметрической сетки поверхности 11×15 .

Написать программу для вычисления двух диагональных параметрических прямых $u = 2w$ и $u = 2(1 - w)$. Составить список точек от $(0, 0)$ до $(2, 1)$ для $u = 2w$ и от $(0, 1)$ до $(2, 0)$ для $u = 2(1 - w)$. Составить список значений $15u$ и $11w$.

Вычислить и изобразить параметрическую поверхность 11×15 . Использовать подходящее зрительное преобразование. Указание: вывести результаты в файл в стандартном графическом формате (.sgf) или в стандартном полигональном формате (.spf) и использовать трехмерные преобразования, разрабатываемые в спецификации 3-1 для изображения поверхности.

Приложение F

СПЕЦИФИКАЦИИ ПРОГРАММ

Спецификации программ разработаны не только для пояснения концепций и методов, представленных в книге, но также для передачи некоторого опыта создания достаточно сложных программ машинной графики. Хорошо выполненные, они могут составить инструментарий полезных графических программ. Законченные программные проекты дают в результате полезные программы общего назначения. Спецификации сгруппированы по главам книги.

Особенно рекомендуются две из этих спецификаций. Первая — это научная программа черчения (спецификация 2-4). Написание хорошей научной программы черчения — нетривиальная задача, однако она вполне доступна студентам старших курсов и дипломникам. Ее можно рекомендовать в качестве одного из двух курсовых проектов для студентов выпускных курсов (другим проектом может быть трехмерный манипулятор (спецификация 3-1)). Студенты старших курсов могут выполнить подобную разработку примерно в течение трех недель.

Вторая спецификация — это трехмерный манипулятор (спецификация 3-1). Трехмерный манипулятор позволяет изображать и совершать различные преобразования (вращение, перемещение, масштабирование, проецирование и т.д.) над трехмерными объектами. Трехмерный манипулятор значительно облегчает задачу визуализации объектов. Кроме того, он улучшает возможности визуализации в трех измерениях. В довершение всего, он полезен для лучшего усвоения материала по трехмерным кривым и поверхностям (гл. 5 и 6). Для студентов старших курсов он может служить вторым из двух курсовых проектов. Дипломник должен справиться с этой работой в течение примерно трех недель.

При проектировании и реализации этих программ особое внимание должно уделяться концепциям структурного программирования и особенно модуляризации программ. Каждая из предложенных спецификаций достаточно ясно распадается на умеренное количество графических инструментальных модулей. Например, большинство графических инструментов для трехмерного манипулятора даны в виде псевдокода в приложении G. Язык реализации несуществен. К настоящему времени все спецификации реализованы студентами на различных вычислительных машинах и различных языках программирования.

Необходима хорошая документация. Пример хорошего документирующего заголовка показан в табл. F-1. Необходимо также подробное документирование внутри программы.

Таблица F-1 Заголовок документации программы

имя:
программист:
дата:
история:
назначение:
метод:
система:
язык:
вызываемые подпрограммы:
файлы ввода:
команда компилятора:
команда компоновщика:
аргументы вызова:
список переменных:

Так как ряд спецификаций задуман для работы в интерактивном режиме, уместно сказать несколько слов об интерфейсе пользователя. В графической среде, не рассчитанной на работу в реальном времени, в качестве интерактивного устройства наиболее часто используется обычная алфавитно-цифровая клавиатура. Для такой среды простым в изучении, быстрым и эффективным является хорошо разработанный интерфейс клавиатуры.

Если не требуется специфический числовый ввод, интерфейс клавиатуры принимает для ввода буквенные слова. Цифровые списки ответов неприемлемы вследствие того, что пользователь мыслит в терминах слов и фраз, а не цифровых элементов или списков. Например, если желаемое действие — это поворот объекта, то пользователь думает о повороте, а не об элементе 3 из списка опций.

Далеко не все пользователи хорошо печатают на машинке. Поэтому число символов, необходимое для ввода команды, должно быть сведено к минимуму. Однако те, которые хорошо печатают, не должны быть поставлены в неудобное положение, раздражены использованием странных последовательностей символов. Одно из возможных решений состоит в том, чтобы допустить полное слово в качестве ввода и затем усечь его до минимального количества символов, необходимого для уникального определения команды. Например команда *rotate* (поворот) усекается до первых двух символов, т. е. *го*. Допустимое усечение, которое дает уникальную последовательность символов, выделяется скобками, например (*го*)*tate*. Численный ввод должен допускаться в свободном формате. Например, в числах с плавающей точкой не обязательно присутствие десятичной точки, т. е. числа 30 и 30. эквивалентны.

Следовало бы обеспечить командное меню. Важный вопрос — как и когда оно поддерживается. Его следовало бы высвечивать один раз при инициировании выполнения программы и затем по запросу. Нецелесообразно высвечивать меню после приема каждой очередной команды, так как это раздражает и к тому же поглощает много времени. Полезно, чтобы командное меню полностью помещалось на экране. На каждом уровне программы, обрабатывающей команды, должно быть обеспечено соответствующее командное меню. Требуется, чтобы для

высвечивания командного меню использовалась единственная командная строка. Такая командная строка состоит из единственного слова или символа. Например «?» или «Option?» или «Command?».

Существует столько же способов запроса командного меню или меню (экрана) помощи, сколько существует программистов. Например, можно набрать h или help или ? или ?? или все, что угодно. Подбор подходящей команды часто оказывается довольно интересным занятием. Простое решение — принимать любой из этих запросов или возвращать простой carriage (символ возврата каретки). Инструментарий прост. Если при разборе командного ввода не обнаруживается допустимой команды, то высвечивается подходящее для этого уровня командное меню (экран подсказки).

При использовании концепции, называемой «командный стек», эффективность интерфейса значительно возрастает. Лучше всего ее смысл проясняется из примера. В спецификации трехмерного манипулятора (спецификация 3-1) команда поворота запрашивает описание оси поворота (ось x, y, z или произвольная) и угла поворота в градусах. Процессор команд верхнего уровня принимает команду rotate (поворот) и вызывает процессор команд нижнего уровня, который запрашивает и вводит оси и угол. Командный стек позволяет опытному пользователю набрать команду в виде

го × 30.

Здесь параметры выделены пробелами. Командный процессор верхнего уровня разбирает строку ввода, определяет, является ли первая команда go допустимой, и, если это так, передает оставшиеся команды командному процессору нижнего уровня. Командный стек позволяет опытному пользователю выполнять серии манипуляций, минимизируя ввод и время ожидания. Начинающий пользователь имеет доступ к таким же возможностям. Например, с помощью командного стека

```
sc o 2
го x 30
го у -45
dr
```

объект масштабируется по всем координатам с коэффициентом 2, поворачивается вокруг оси x на 30°, поворачивается вокруг оси y на 45°, а результат отображается в плоскости z = 0 с использованием ортографической проекции.

В завершение отметим, что интерфейс пользователя должен быть согласован на всех уровнях. Следует придерживаться «принципа наименьшего удивления». Иначе говоря, при наличии множества ответов на команду выбор должен основываться на том, «какой ответ вызовет наименьшее удивление пользователя».

ГЛАВА 2

Первые две спецификации были разработаны для того, чтобы ознакомить студентов с локальным графическим пакетом и заложить основу базисного инструментария графических модулей. Третья спецификация служит для приобретения опыта работы с оконной техникой на графическом устройстве и является основой

для разного рода управляющих преобразований. Четвертая относится к развитию программы визуализации научных результатов.

2-1. Разработать и реализовать программу рисования замкнутого многоугольника с вершинами $P_1[0 \ 0], P_2[2 \ -2], P_3[2 \ 1], P_4[0 \ 3], P_5[-2 \ 1], P_6[-2 \ -2]$, выбрав направление обхода против часовой стрелки. Использовать команды окна и/или области вывода графического устройства, чтобы многоугольник заполнял все пространство экрана. Будьте внимательны при построении геометрических отношений, одна единица по оси x должна быть равна одной единице по оси y . Включить оси, проходящие через начало координат. Разбить оси на единичные интервалы, расставить маркировку на осях.

Разработать и включить в программу инструментальные модули для:

Рисования осей через точку $x0, y0$, от $x\text{-min}$ до $x\text{-max}$ и от $y\text{-min}$ до $y\text{-max}$.

Размещения левых, правых или центральных осевых маркеров на осях x и y , определенной длины, расположенных через определенный интервал, начиная от определенной точки.

Разметки осевых маркеров подходящими значениями.

Рисования произвольного многоугольника. (Приложение A-7.)

2-2. Разработать и усовершенствовать программу генерации и отображения кривых, описанных параметрическими уравнениями

$$\text{a. } x = \{2 + 7 \cos(\sin(\theta) + \sin(121\theta))\} \cos(\theta) \quad \pi/2500 \leq \theta \leq 2\pi$$

$$y = \{2 + 7 \cos(\sin(\theta) + \sin(121\theta))\} \sin(\theta)$$

$$\text{b. } x = \{11/10 + \cos(8\theta)^{19}\} \cos(\theta) \quad \pi/2000 \leq \theta \leq 2\pi$$

$$y = \{1.1 + \cos(8\theta)^{19}\} \sin(\theta)$$

$$\text{c. } x = \{1 + 2 \sin(17\theta/4)\} \cos(\theta) \quad \pi/500 \leq \theta \leq 8\pi$$

$$y = \{1 + 2 \sin(17\theta/4)\} \sin(\theta)$$

$$\text{d. } x = \{-\theta \sin(\pi\theta)/8\} \cos(\theta) \quad 9\pi/1000 \leq \theta \leq 36\pi$$

$$y = \{-\theta \sin(\pi\theta)/8\} \sin(\theta)$$

$$\text{e. } x = \{\sin(13\theta/4)\} \cos(\theta) \quad \pi/100 \leq \theta \leq 100\pi$$

$$y = \{\sin(13\theta/4)\} \sin(\theta)$$

Используя команды работы с окнами и/или управления областями вывода, увеличить эти фигуры до размеров экрана. Будьте внимательны при построении геометрических отношений, одна единица по оси x должна быть равна одной единице по оси y . Включить оси, проходящие через начало координат. Разбить оси на единичные интервалы, расставить маркировку на осях.

Разработать и включить в программу инструментальные модули для:

Рисования осей через точку $x0, y0$, от $x\text{-min}$ до $x\text{-max}$ и от $y\text{-min}$ до $y\text{-max}$.

Размещения левых, правых или центральных осевых маркеров на осях x и y , определенной длины, расположенных через определенный интервал, начиная от определенной точки.

Разметки осевых маркеров подходящими значениями.

Рисования фигуры. (Приложение A-7.)

После преобразования

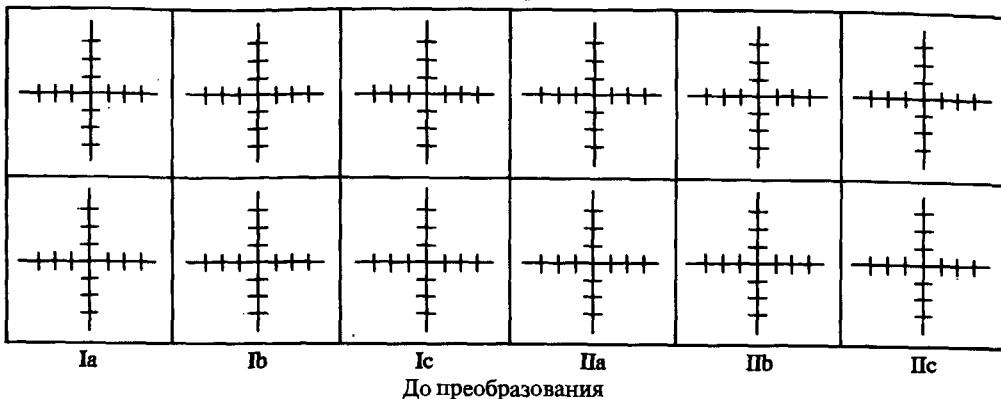


Рис. F-1

2-3. Выполнить следующие преобразования для двух прямых P_1P_2 и P_3P_4 , заданных $P_1[-2 \quad -1]$, $P_2[1 \quad 2]$, $P_3[-1 \quad -1]$, $P_4[-1 \quad 2]$:

- поворот на -60° ;
- зеркальное отражение относительно прямой $y = x$;
- масштабирование с коэффициентом 2.

Выполнить следующие преобразования для второй пары прямых P_5P_6 и P_7P_8 , определенных через $P_5[-1 \quad -1]$, $P_6[2 \quad 2]$, $P_7[-3 \quad -1]$, $P_8[0 \quad 2]$:

- поворот на 60° , а затем зеркальное отражение относительно оси x ;
- зеркальное отражение, относительно оси x , после поворота на 60° ;
- преобразования $\begin{bmatrix} -1 & 3 \\ 2 & 4 \end{bmatrix}$.

Получить указанные результаты с помощью вычислений вручную.

Разработать и реализовать программу отображения результатов на один экран в формате, показанном на рис. F-1. Три левые клетки в нижнем ряду содержат первую пару прямых до преобразования. Три правых клетки содержат вторую пару прямых до преобразования. В верхнем ряду находятся соответствующие преобразованные прямые. Каждая клетка содержит оси с разметкой. Так как пространство ограничено, может оказаться желательным не наносить метки значений маркеров. Однако необходимо указать значение каждого маркера в каждой клетке. Соответствующее изображение должно быть помечено как индикатор.

При разработке и реализации этих программ добросовестный пользователь обратится к концепции окон и областей вывода графического устройства (разд. 1-4 и приложение А-4). Если абстрагироваться от размера окна, размещения клеток на экране и конкретных операций преобразования, то действия, требуемые для генерации каждого бокса, представляются идентичными. Таким образом, главная программа выглядит так:

```

for i = 1 to 6
    применить преобразование, если надо
    определить размер окна
    call window
    for j = 1 to 2
        call viewport(j)
        call box
        call axes
        call tick
        call label
        call drawlines(j)
    next j
next i

```

Разработать и включить в программу инструментальные модули для

- определения требуемого размера окна;
- рисования клетки вокруг окна;
- рисования осей через точку $x0, y0$, от $x\text{-min}$ до $x\text{-max}$ и от $y\text{-min}$ до $y\text{-max}$;
- размещения левых, правых или центральных осевых маркеров на осях x и y определенной длины, расположенных через определенный интервал, начиная от определенной точки;
- разметки осевых маркеров подходящими значениями;
- рисования прямых.

Позаботьтесь также о включении общепринятых модулей двумерного поворота, отражения и масштабирования на плоскости (приложение G). Используйте эти модули в программах для генерации матриц необходимых преобразований. Когда требуются комбинированные преобразования (задания a и b для второй пары линий), нужно перемножить матрицы перед применением их к исходной паре прямых.

2-4. Разработать и реализовать программу визуализации научных данных, которая принимает входной поток данных в форме двумерного файла .sgf (табл. А-2 из приложения А-6). При решении данной задачи воспользуйтесь комментариями относительно интерфейса пользователя, данными в начале этого приложения. Программа должна удовлетворять следующим характеристикам и условиям:

ввод	Ввод данных из двумерного файла .sgf.
чертение	Отображение графиков в соответствии с пределами, метками, аннотациями и другими спецификациями. По умолчанию применяется автоматическая установка пределов, данные присоединяются в виде единой линии, нет меток, клетка без сетки и аннотация отсутствуют. (См. ниже.)
метки	Помечаются график, ось x и ось y .
оси	Обе оси линейные, одна или обе оси логарифмические. Требуются точные значения пределов по осям. Допустимые пределы по осям начинаются или заканчиваются в $0, \pm 1, \pm 2, \pm 5$. Таким образом, оси могут иметь пределы $0 \rightarrow 1, 0 \rightarrow 2, 0 \rightarrow 5, -1 \rightarrow 1, -1 \rightarrow 5, -2 \rightarrow 5$ и т. д. Случайные пределы, например $0 \rightarrow 1.736$, не допускаются, за исключением специально установленных командой range (пределы). Оси помечаются в соответствии с порядком значения 1. Величина метки указана

	зывается как множитель, расположенный следом за последней меткой оси, который задается в форме $\times 10^2$. Логарифмические оси размечаются только в интервалах, соответствующих 1, 5 и 10.
диапазон	Изменение пределов графика. Запрашиваются минимальное и максимальное значения x и y .
автодиапазон	Автоматически определяются приближенные значения пределов по осям x и y . (См. оси.)
тип линии	Выбирается тип линий для определяемой кривой. Представлено меню допустимых типов. Минимально необходимое множество — это штриховая линия, непрерывная линия, точка в каждой позиции графика, звездочка в каждой позиции графика, плюс в каждой позиции графика или \times в каждой позиции графика. Допустимы комбинации позиций графика и типов линий.
пометки	Выбор типа маркера пометки. Представлено меню возможных типов. Минимально необходимое множество — это пометки на левой или правой стороне осей, если смотреть из начала координат в положительном направлении осей, пометки на обеих сторонах осей, длинные пометки в более важных точках деления и короткие в менее важных.
у-метка	Способ высвечивания метки оси u : метка расположена по вертикали так, что читается сбоку, или так, что читается сверху вниз.
сетка	Включение/выключение сетки.
квадрат	Построение графического квадрата, сохраняющего геометрические отношения, т.е. единица по оси x равна единице по оси y .
клетка	Переключение способа отображения графика на экране: график отображается в клетке с разметкой по всем четырем сторонам или график отображается только с двумя координатными осями.
масштаб	Изменение размеров графика путем указания масштабных множителей по осям x и y .
цвет	Задает цвет отдельно каждой кривой графика из смешанной палитры красного, зеленого и синего цвета. Значения цвета задаются из интервала $0 \leq r, g, b \leq 1$, где 0 — отсутствие данного цветового компонента, 1 — максимальный вклад.
аннотация	Размещает соответствующую текстовую аннотацию в произвольной позиции графика. Возможно как автоматическое, так и интерактивное размещение аннотации. Опция автоматического размещения служит для ввода строк и отображения их в левом верхнем углу графика. Опция интерактивного размещения позволяет позиционировать каждую введенную строку аннотации запросом координат x, y для левого нижнего угла строки. Представлено меню возможных шрифтов для вывода аннотации. (См. шрифты)
отмена	Позволяет отменить некоторые предыдущие команды, например метка, аннотация, пределы и т. д., в частности ввод последней команды.
переустановить	Перезадает всем опциям значения по умолчанию. (См. черчение.)
шрифт	Выбор одного из типов шрифта. Представлено меню возможных шрифтов.
твердая	Делает твердую копию изображения графика.
копия	Представлено меню доступных опций, включающее вывод в файл в формате .sdf (табл. А-1 из приложения А-5).
сохранить	Сохраняет текущий график в файле.
восстановить	Загружает из файла ранее сохраненный график. После этого возможны любые модификации этого графика.
округлить	Генерирует кривую, проходящую через заданные точки кривой. Представлено меню возможных методов построения таких кривых, которое должно включать следующие методы: полиномов, степенной и экспо-

	ненциальный наименьших квадратов, кубический сплайн, сплайн Безье и В-сплайн.
выход	Прекращает работу программы и возвращает управление операционной системе.

ГЛАВА 3

Здесь приведена единственная спецификация — разработка и развитие трехмерного манипулятора. Предполагается, что объекты отображаются с использованием каркасного представления. Разрешен ввод множественных (составных) объектов. Для успешного выполнения задания требуется разработка необходимых модулей для трехмерного поворота, переноса, масштабирования и т. д., а также для получения ортографической, косоугольной, перспективной проекций и т. д.

3-1. Разработать и реализовать программу трехмерного манипулятора, который вводит данные из файла как в формате .sgf, так и в формате .spf, описанных в табл. А-2 и А-3 из приложения А-6. Комментарии по поводу интерфейса пользователя, приведенные в начале этого приложения, справедливы и при разработке этой программы. Программа должна удовлетворять следующим характеристикам и условиям:

ввод	Ввод данных из файла как в формате .sgf, так и в формате .spf.
разделитель	Спецификации объектов разделяются с помощью знака #.
поворот	Поворот объектов относительно координатных осей x , y , z или относительно произвольной оси вращения. Произвольная ось вращения определяется как с помощью точки и трех направляющих косинусов, так и двумя точками на осях координат. Произвольный поворот получается из поворотов относительно координатных осей. Поворот может задаваться как относительно мировых координатных осей, так и относительно локальных осей тела.
перенос	Параллельный перенос объектов в направлении x , y или z .
отражение	Зеркальное отражение объектов относительно плоскостей $x = 0$, $y = 0$ или $z = 0$.
масштаб	Масштабирование объектов локально, т. е. отдельно в направлении x , y или z , или глобально, т. е. одинаково по всем направлениям.
оси	Определены два типа трехмерных осей: оси тела и фиксированные оси. В любой момент времени могут отображаться как одни из них, так и те, и другие сразу. Размеры переменные. Маркеры разметки могут быть включены или выключены. Оси тела фиксированы в центре объекта или группы объектов. Они перемещаются, т. е. вращаются, переносятся, отражаются вместе с объектом. Фиксированные оси располагаются в точке начала мировой системы координат. Они являются субъектом преобразования кадра, т. е. изометрической, диметрической, косоугольной или перспективной проекции.
проекция	Проекция объектов на плоскость $z = 0$ — изометрическая, диметрическая, косоугольная, кабине, кавалье или центральная перспективная проекция. Изометрические и диметрические преобразования поддерживают выбор положительного или отрицательного поворота относительно осей x , y . Диметрическая и косоугольная проекции допускают переменный множитель искажения. Косоугольная, а также проекции кабине и кавалье допускают переменный угол наклона относительно горизонтали. Центральная перспективная проекция на плоскость $z = 0$

	с переменным центром выполняется только для центров, расположенных на оси z .
трансфокация	Трансфокация (наезд) камеры относительно кадра изображения. Замечание: это не масштабирование объектов.
исходный	Восстанавливает исходные данные объекта.
рисовать	Рисует преобразованные объекты, используя ортографическую проекцию на плоскость $z = 0$.
вывод	Сохраняет текущие преобразованные данные точек в файле формата .sgf или .spf.
сохрматр	Сохранить текущую матрицу преобразования в файле.
тень	Создать изображение с тенями. ¹ Требуется, чтобы данные вводились в формат .spf.
невидимые	Создать изображение с невидимыми линиями/поверхностями. ² Требуется, чтобы данные вводились в формате .spf.
твердая копия	Делает твердую копию изображения. Представлено меню доступных устройств вывода. Требуется возможность вывода в файл формата .sdf. (Табл. А-1 из приложения А-5.)
центр	Изображение объектов центрируется относительно геометрического центра данных.
объект #	Включает/выключает отдельный объект или группу объектов, задаваемую как #.
4 вида	Создает четыре вида объектов: ортографические фронтальные проекции, виды сбоку и сверху в нижнем левом, нижнем правом и верхнем левом квадрантах соответственно, плюс текущую проекцию в верхнем правом квадранте.
окружение	Удаляет с экрана любые элементы окружения, например меню.
выход	Прекращает работу программы и возвращает управление операционной системе.

Программа структурирована так, чтобы все матрицы преобразований применялись до выполнения команды рисования или какой-либо команды проецирования. Перед выполнением любой из этих команд комбинированная матрица преобразования применяется к базе данных, а результирующие преобразованные двумерные координатные векторы рисуются на устройстве вывода. Исходная трехмерная база данных не модифицируется командами рисования или проекции.

Главная программа состоит из условного цикла в виде:

```

Выход = произвольная строка
while Выход <> q then
    print Опцию ? на экран
    input Ответ от клавиатуры
    truncate Ответ до первых двух символов
    convert Ответ в регистр строчных букв
    if Ответ = ro then call подменю поворота
    if Ответ = re then call подменю отражения
    if Ответ = pr then call подменю проекции
    :
    print главный экран помощи
end while

```

¹ См. Д. Роджерс, Алгоритмические основы машинной графики. — М.: Мир, 1989.

² См. предыдущую ссылку.

Типичная программа подменю могла бы выглядеть так:

```

Выход = произвольная строка
while Выход <> q then
    print Вращение относительно какой оси?
    input Ответ от клавиатуры
    truncate Ответ до первых двух символов
    convert Ответ в регистр строчных букв
    if Ответ = x then call программу x-поворота
    if Ответ = y then call программу y-поворота
    if Ответ = z then call программу z-поворота
    if Ответ = g then call программу произвольного поворота
    print локальный экран помощи
end while

```

Опираясь на эту базовую концепцию, легко понять, что программа имеет по крайней мере три уровня действий. Первый — выбор основной операции, например поворот, второй — выбор ее параметров, например ось поворота, третий — выполнение операции.

При необходимости возможности, описанные в спецификации, можно сократить путем удаления требований к программе.

В качестве примера рассмотрим Т-блок, определенный ниже

0	0	4	A
4	0	4	B
4	2	4	C
3	2	4	C
3	4	4	E
1	4	4	F
1	2	4	G
0	2	4	H
0	0	0	I
4	0	0	J
4	2	0	K
3	2	0	L
3	4	0	M
1	4	0	N
1	2	0	O
0	2	0	P

Разработать единую матрицу преобразования, которая:

- сокращает размер на 1/2;
- поворачивает относительно оси x на 30° ;
- поворачивает относительно оси y на -30° ;
- переносит на -2 единицы в направлении y ;
- выполняет центральную перспективную проекцию на плоскость $z = 0$ с центром проекции $z = 10$.

Сначала разработать, потом применить комбинированную матрицу преобразования для этих операций к базе данных путем вычисления вручную. Построить

результат на графической бумаге. Вычислить и построить некоторые невидимые точки. Использовать результаты этих ручных вычислений для проверки операций трехмерного манипулятора.

ГЛАВА 4

Единственная приведенная здесь спецификация основывается на расширении материала разд. 4-9. Предлагаемое расширение дано в задачах с 4-6 по 4-10. Подобные конструкции требуются в системах компьютерной поддержки черчения и во многих управляющих приложениях.

4-1. Разработать и реализовать программу интерактивного задания геометрии, определенной в разд. 4-9, и/или ее расширения, определенного в задачах с 4-6 по 4-10. Будьте внимательны при построении корректных геометрических отношений для отображения результатов; например, окружность должна выглядеть как окружность, а не как эллипс. Комментарии по поводу интерфейса пользователя, приведенные в начале этого приложения, справедливы и при разработке этой программы. Программа должна удовлетворять следующим характеристикам и условиям:

line	Определяет отрезок по его двум конечным точкам.
circle	Определяет окружность по ее центру и радиусу (h, k).
add	Добавляет окружность или отрезок к геометрии.
delete	Удаляет окружность или отрезок из геометрии.
move	Перемещает ранее определенные окружность или отрезок.
oxes	Включает/выключает оси.
ticks	Включает/выключает маркеры разметки осей.
tables	Включает/выключает метки маркеров разметки осей.
solution	Выбирает подходящее из возможных решений, используя такие концепции, как левый, правый, ближний, дальний, внутри, вне, см. разд. 4-9.

В качестве частей программы разработать и включить инструментальные модули для

- рисования осей, проходящих через точку $x0, y0$, от $x\text{-min}$ до $x\text{-max}$ и от $y\text{-min}$ до $y\text{-max}$;
- левого, правого или центрального расположения маркеров разметки осей x и y , определенной длины, начиная с заданной позиции через заданный интервал;
 - нанесения на маркеры разметки подходящих значений;
 - генерации и рисования окружности и дуги окружности.

Обязательно также разработайте модули необходимых двумерных преобразований, например поворота, переноса, зеркального отражения и т.д., необходимых для размещений в стандартной ориентации. Когда требуются множественные преобразования, перемножайте матрицы перед их применением к геометрической базе данных. Разработайте необходимые модули для поддержки движения по дереву решений, для выбора необходимого решения, для определения, является ли выбранное решение физически возможным, и выполнения необходимых вычислений для поиска некоторых неизвестных геометрических величин.

ГЛАВА 5

Здесь приведена спецификация программы генерации и манипуляции с кривыми. Каждая из этих кривых обсуждалась в гл. 5. В частности, речь шла о кубическом сплайне, параболически смешанном, сплайнe Безье и В-сплайне как рациональных, так и нерациональных. Они используются в качестве базиса программы. Включение возможностей генерации составной кривой нескольких типов делает особенно удобным сравнение.

5-1. Разработать и реализовать программу генерации и манипуляции трехмерными кривыми, которая принимает входные данные в формате трехмерного файла .sgf (табл. А-2 из приложения А-6). Сгенерированная трехмерная кривая строится с использованием ортографической проекции на плоскость $z = 0$. Комментарии по поводу интерфейса пользователя, приведенные в начале этого приложения, справедливы и при разработке этой программы. Программа должна удовлетворять следующим характеристикам и условиям:

ввод	Ввод данных из файла в трехмерном формате .sgf.
тип	Выбор типа кривой. Представлено меню возможных типов кривых, например кубический сплайн, параболически смешанный, сплайн Безье, В-сплайн. Для кубического сплайна реализованы как нормализованная, так и хордовая аппроксимации и, по крайней мере, условия жесткого и свободного конца. Для параболически смешанной кривой реализована аппроксимация для α и β с переменным значением α , $\alpha = 0.5$ по умолчанию. Для кривых Безье и В-сплайна реализованы рациональный и нерациональный варианты, а также возможности повышения степени и подразбиения.
многоугольник	Рисует прямые линии между введенными точками данных. Для кривых Безье и В-сплайна это задание определяющего многоугольника. Отображает метки следом за отдельными точками входных данных, например P_1, P_2, P_3 и т. д.
метка	Добавляет точку данных. Обязательна возможность добавить точку перед первой точкой данных.
добавить	Добавляет точку данных. Обязательна возможность добавить точку перед первой точкой данных.
удалить	Удаляет точку данных.
переместить	Перемещает точку данных.
очистить	Определяет, будет или нет экран очищаться перед рисованием первой кривой.
оси	Включает/выключает графические оси. Оси включают подходящие маркеры разметки и метки.
диапазон	Изменяет пределы графика. Запрашиваются минимальное и максимальное значения для x и y .
автодиапазон	Автоматически определяет подходящие пределы для x и y .
тип линии	Выбор типа линии для заданной кривой. Представлено меню возможных типов. Как минимум, оно должно включать штриховую и сплошную линии.
кривая #	Включает/выключает отдельную кривую или группу кривых, определенных как #.
восстановить #	Восстанавливает кривые, определяемые как #, из базы данных программы.
исходные	Восстанавливает исходные данные кривой.
печатать	Печатает на экране конструктивные данные кривой, например определяющий многоугольник для кривой Безье или В-сплайна.

твёрдая копия	Делает твердую копию текущего изображения. Представлено меню доступных устройств вывода. Необходима возможность вывода в файл в формате .sdf. (Табл. А-1 из приложения А-5.)
вывод	Сохраняет трехмерные точки данных сгенерированной кривой в файле формата .sgf для последующего отображения на экран и преобразований с помощью трехмерного манипулятора. (Спецификация 3-1.)
выход	Прекращает работу программы и возвращает управление операционной системе.

При необходимости возможности, описанные в спецификации, можно сократить путем удаления требований к программе.

Используйте вычисления из примера 5-2 и задач 5-2, 5-5, 5-7, 5-9, 5-10, 5-12, 5-20 и 5-21 для проверки работы программы генерации и преобразований трехмерных кривых.

5-2. Добавить к спецификации 5-1 возможности для кривых Безье и В-сплайна, как нерациональных, так и рациональных. Данные вводятся из файла в формате .sgf.

ГЛАВА 6

Здесь приведена спецификация программы генерации и манипуляции для поверхности. Она очень похожа на программу генерации и преобразований для кривой из гл. 5 и обладает многими аналогичными характеристиками. Она также может быть построена как расширение программы трехмерного манипулятора (спецификация 3-1). В этом случае генерация и манипуляция поверхностью является просто дополнительной функцией.

Поверхности, обсуждавшиеся в гл. 6, например билинейные, правильные, линейные или бикубические поверхности Кунса, поверхности Безье или В-сплайны, как нерациональные, так и рациональные, используются в качестве базиса программы. Включение возможностей генерации и отображения составных поверхностей делает особенно удобным исследование эффектов воздействия различных вариантов входных данных.

6-1. Разработать и реализовать программу генерации и манипуляции трехмерными поверхностями, которая получает входные данные в формате трехмерного файла .sgf, описанного в табл. А-2 приложения А-6 или точки многоугольной сетки в форме файла .prp для поверхности Безье или В-сплайна (табл. D-1 приложения D). Сгенерированная поверхность строится с использованием подходящей проекции на плоскости $z = 0$, например изометрической, диметрической, кабине или кавалье. Результаты выводятся в формате .sgf или в формате .spf и могут быть использованы впоследствии. Комментарии по поводу интерфейса пользователя, приведенные в начале этого приложения, справедливы и при разработке данной программы. Программа должна удовлетворять следующим характеристикам и условиям:

ввод поверхность	Ввод данных из файла в трехмерном формате .sgf. Генерирует поверхность. Для поверхностей Безье и В-сплайна реализованы как нерациональный, так и рациональный варианты, а также возможности увеличения порядка и степени разбиения. (Задачи 6-17 и 6-28 предлагаются в качестве тестовых поверхностей.)
---------------------	--

границы диагонали параметризация	Вычисляет и отображает четыре граничные кривые поверхности. Вычисляет и отображает две диагональные кривые поверхности. Вычисляет и отображает ортогональную сетку изопараметрических линий на поверхности. Запрашивает число линий в каждом параметрическом направлении.
рисовать	Отображает поверхность с использованием подходящей проекции, например изометрической, диметрической, кабине или кавалье.
многоугольник	Рисует прямые линии между точками входных данных. Применяется для поверхностей Безье и В-сплайна, чтобы показать определяющий многоугольник.
оси	Определены два типа трехмерных осей: оси тела и фиксированные оси. В любой момент времени могут отображаться как одни из них, так и те и другие сразу. Размеры переменные. Маркеры разметки могут быть включены или выключены. Оси тела фиксированы в центре объекта или группы объектов. Они перемещаются, т. е. врачаются, переносятся, отражаются вместе с объектом. Фиксированные оси располагаются в точке начала мировой системы координат. Они являются субъектом преобразования кадра, т. е. изометрической, диметрической, косоугольной или перспективной проекции.
метка	Отображает метки вслед за отдельными точками входных данных, например P_1 , P_2 , P_3 и т.д.
модифицировать	Модифицирует характеристики поверхности. Представлено меню возможных модификаций, одной из которых является перемещение координатных векторов. Для бикубической поверхности Кунса также включены модификации касательных векторов и векторов вращения.
трансформация	Трансформация камеры относительно кадра изображения. Замечание: это не масштабирование поверхности.
очистить	Определяет, будет ли очищаться экран перед рисованием текущей поверхности.
исходные печать	Восстанавливает исходные данные поверхности.
твердая копия	Печатает на экране конструктивные данные поверхности, например определяющий многоугольник для поверхности Безье и В-сплайна.
вывод	Делает твердую копию текущего изображения.
выход	Представлено меню доступных устройств вывода. Необходима возможность вывода в файл в формате .sdf. (Табл. A-1 из приложения A-5.)

При необходимости возможности, описанные в спецификации, можно сократить путем удаления требований к программе.

Использовать вычисления из примеров 6-9, 6-10, с 6-12 по 6-15 и 6-18 и задач с 6-9 по 6-11 для проверки операций программы генерации и манипуляций с поверхностями.

6-2. Добавить к спецификации 5-1 возможности для трехмерных поверхностей. Данные вводятся из файла в формате .sgf. (Табл. A-2 из приложения A-6.)

Приложение G

АЛГОРИТМЫ

В этом приложении приведено несколько полезных алгоритмов. Все они представлены с помощью псевдокода, описанного в приложении С. Выбрать язык для представления алгоритмов было непросто. Перспективными кандидатами выглядели Паскаль, Фортран, Си и ANSI Бейсик. С точки зрения машинной графики каждый из этих языков имеет сильные и слабые стороны. Окончательное решение выбрано на основе педагогических соображений. Представление с помощью псевдокода требует, чтобы алгоритмы были переведены на «реальный» язык программирования. Есть надежда, что этот процесс улучшит понимание алгоритмов, а следовательно, и лежащей в их основе математики.

Версии алгоритмов на псевдокоде алгоритмически построены на основе работающих программ. Мы надеемся, что такой способ привел к получению более точных алгоритмов. Однако в процессе печати могли быть внесены отдельные ошибки. Поэтому при реализации следует еще раз внимательно сверить алгоритмы с математическим представлением в тексте.

Алгоритмы разрабатывались для обучения, т. е. они задуманы в качестве иллюстраций математических методов, обсуждавшихся в книге, но не как эффективно работающие программы для промышленной эксплуатации. Во многих случаях признавалась приемлемой вычислительная неэффективность, если она окупается тем, что алгоритм более полно соответствует основному тексту книги. Вычислительная неэффективность допускалась и для сокращения требуемой памяти. Таким образом, программы, разработанные в соответствии с предложенными алгоритмами, применимы для микрокомпьютерных операционных систем в условиях ограниченной доступной оперативной памяти. Примером служит алгоритм *bspolsurf* построения поверхности В-сплайна. В предложенном ниже алгоритме на псевдокоде все базовые функции для каждого значения параметра вычисляются вне главного цикла. Это сокращает требования к размерам памяти. Однако с точки зрения минимизации времени выполнения было бы более эффективно вычислять все базовые функции в главном цикле, получать результаты и запоминать их в большом массиве. Элементы этого массива используются затем в основном цикле для вычисления координатных векторов на поверхности. Через несколько минут работы становится ясно, что для достаточно сложных поверхностей размеры массива получаются довольно большими. Однако алгоритм выполняется в несколько раз быстрее, чем представленный алгоритм на псевдокоде. При реализации алгоритмов предполагается, что первоначально они будут реализованы

в том виде, в котором они даны в книге. Сначала надо удостовериться в правильности программы, а уже затем следует заниматься исследованием алгоритма и его математического обоснования с целью обнаружения методов повышения эффективности.

Алгоритмы сгруппированы по главам книги и расположены в алфавитном порядке для каждой главы.

ГЛАВА 2

Здесь приведены алгоритмы, выполняющие базовые двумерные манипуляции. Вот они в алфавитном порядке: **reflt2d**, **rot2d**, **scale2d**, **tran2d**.

reflt2d

Подпрограмма формирования и конкатенации матрицы отражения в плоскости к текущей преобразующей матрице (уравнения 2-34-2-36).

reflt(,) = матрица отражения
rficode = код отражения
 1 = относительно оси *x*
 2 = относительно оси *y*
t(,) = текущая преобразующая матрица, с которой будет конкатенирована матрица отражения

subroutine reflt2d (rficode; t(),)

dimension reflt(3,3)

Mat reflt = Idn

if rficode = 1 then reflt(2,2) = -1 установить соответствующий элемент равным -1
if rficode = 2 then reflt(1,1) = -1

Mat t = t*reflt

конкатенировать матрицы

return

rot2d

Подпрограмма формирует матрицу поворота в плоскости и конкатенирует ее с текущей матрицей преобразования (уравнение (2-29)).

rot(,) = матрица поворота
theta = угол поворота в градусах
t(,) = матрица, с которой будет конкатенирована матрица поворота

subroutine rot2d (theta; t(),)

option angle degrees

dimension rot(3,3)

Mat rot = Idn

rot(1,1) = Cos(theta)

установка значений матрицы поворота

rot(1,2) = Sin(theta)

rot(2,2) = rot(1,1)

rot(2,1) = -rot(1,2)

```
Mat t = t*rot
return
```

конкатенация матриц

scale2d

Подпрограмма формирует матрицу масштабирования в плоскости и конкатенирует ее с текущей матрицей преобразования (уравнения (2-37)–(2-58)).

a, d = масштабирующие множители в направлениях *x* и *y*;
a, d > 1 – растяжение; *a, d < 1* – сжатие
s = общий масштабирующий множитель; *s > 1* – увеличение;
s < 1 – сокращение
Scale() = матрица масштабирования
t() = текущая преобразующая матрица, с которой будет конкатенирована матрица масштабирования

```
subroutine scale2d (a, d, s; t(),)
```

```
dimension scale(3,3)
```

```
Mat scale = Idn
```

установка значений матрицы масштабирования

```
scale(1,1) = a
```

```
scale(2,2) = d
```

```
scale(3,3) = 1/s
```

```
Mat t = t*scale
```

конкатенация матриц

```
return
```

tran2d

Подпрограмма формирует матрицу параллельного переноса в плоскости и конкатенирует ее с текущей матрицей преобразования (уравнение (2-50)).

m, n = множитель переноса в направлениях *x* и *y*, соответственно
t() = матрица, с которой будет конкатенирована
 матрица переноса
tran() = матрица переноса

```
subroutine tran2d (m, n; t(),)
```

```
dimension tran(3,3)
```

```
Mat tran = Idn
```

установка значений матрицы переноса

```
tran(3,1) = m
```

```
tran(3,2) = n
```

```
Mat t = t*tran
```

конкатенация матриц

```
return
```

ГЛАВА 3

Здесь приведены алгоритмы, выполняющие базовые трехмерные манипуляции. Также предлагается ряд алгоритмов для проецирования на плоскость. В алфавитном порядке это: **cabinet**, **cavalier**, **dimetric**, **genrot**, **isometric**, **oblique**, **perspective**, **project**, **reflt3d**, **scale3d**, **tran3d**, **xrot**, **yrot**, **zrot**.

cabinet

Подпрограмма формирует матрицу проекции кабине и конкатенирует ее с текущей матрицей преобразования (с. 187 и пример 3-16).

*alpha = угол между горизонталью и осью проекции z
 $t(,)$ = текущая преобразующая матрица, с которой будет конкатенирована матрица проекции кабине*

```
subroutine cabinet (alpha; t(,))
option angle degrees
выполнить проекцию кабине, вызвав oblique при  $f = 0.5$ 
call oblique(alpha,0.5; t(,))
return
```

cavalier

Подпрограмма формирует матрицу проекции кавалье и конкатенирует ее с текущей матрицей преобразования (с. 187 и пример 3-16).

*alpha = угол между горизонталью и осью проекции z
 $t(,)$ = текущая преобразующая матрица, с которой будет конкатенирована матрица проекции кавалье*

```
subroutine cavalier (alpha; t(,))
option angle degrees
выполнить проекцию кавалье, вызвав oblique при  $f = 1$ 
call oblique(alpha,1; t(,))
return
```

dimetric

Подпрограмма формирует матрицу диметрической проекции и конкатенирует ее с текущей матрицей преобразования (уравнения (3-32), (3-39), (3-40) и пример 3-14).

*f = коэффициент искажения в направлении z
 $t(,)$ = текущая преобразующая матрица, с которой будет конкатенирована матрица диметрической проекции
xcode = код x поворота
 0 = отрицательный угол поворота
 1 = положительный угол поворота
ycode = код y поворота
 0 = отрицательный угол поворота
 1 = положительный угол поворота*

```
subroutine dimetric (ycode,xcode,f; t(,))
option angle degrees
найти углы поворота
phi = Arcsin(f/(Sqrt(2 - f*f)))
theta = Arcsin(f/Sqrt(2))
установить значения направлений поворота
```

```

if ycode = 0 then phi = -phi
if xcode = 0 then theta = -theta
выполнить диметрические преобразования, вызвав уrot, xrot и project
call yrot (phi; t)
call xrot (theta; t)
call project (3; t)
return

```

genrot

Подпрограмма формирует матрицу трехмерного поворота относительно произвольной оси в пространстве и конкатенирует ее с текущей матрицей преобразования методом совмещения оси поворота с осью z (уравнения (3-21) и (3-23)–(3-26)).

*a(,) = массив, содержащий две точки оси поворота
 delta = угол поворота относительно произвольной оси в градусах
 t(,) = текущая преобразующая матрица, с которой будет
 конкатенирована матрица поворота*

```

subroutine genrot (a(,),delta; t())
option angle degrees
dimension rx(4,4),ry(4,4),rdelta(4,4),rxinv(4,4),ryinv(4,4)
dimension tran(4,4),traninv(4,4)
dimension m(4,4),minv(4,4)
dimension templ(4,4),temp2(4,4)
Mat rx = Idn
Mat ry = Idn
Mat rdelta = Idn
Mat tran = Idn
Mat rxinv = Idn
Mat ryinv = Idn
Mat traninv = Idn
вычислить направляющие косинусы
cx = a(2,1) – a(1,1)
cy = a(2,2) – a(1,2)
cz = a(2,3) – a(1,3)
nfactor = Sqrt(cx*cx + cy*cy + cz*cz)
cx = cx/nfactor
cy = cy/nfactor
cz = cz/nfactor
d = Sqrt(cy*cy + cz*cz)
установить значения переменных матриц
заметим, что, так как rx и ry – абсолютные повороты, обращение
является транспонированием
tran(4,1) = -a(1,1)
tran(4,2) = -a(1,2)
tran(4,3) = -a(1,3)
traninv(4,1) = a(1,1)
traninv(4,2) = a(1,2)
traninv(4,3) = a(1,3)

```

```

rx(2,2) = cz/d
rx(2,3) = cy/d
rx(3,2) = -rx(2,3)
rx(3,3) = rx(2,2)
Mat rxinv = Trn(rx)
ry(1,1)=d
ry(1,3)=cx
ry(3,1)=-cx
ry(3,3)=d
Mat ryinv = Trn(ry)
rdelta(1,1)=Cos(delta)
rdelta(2,2)=rdelta(1,1)
rdelta(1,2)=Sin(delta)
rdelta(2,1)=-rdelta(1,2)
конткатенировать переменные матрицы
Mat temp1 = rx*ry
Mat m = tran*temp1
Mat temp2 = rxinv*traninv
Mat minv = ryinv*temp2
Mat temp1 = rdelta*minv
Mat temp2 = m*temp1
Mat t = t*temp2
return

```

imsometric

Подпрограмма формирует матрицу изометрической проекции и конкатенирует ее с текущей матрицей преобразования (уравнения (3-32), (3-41) и пример 3-15).

t(,) = текущая преобразующая матрица, с которой будет конкатенирована матрица изометрической проекции

xcode = код x поворота

0 = отрицательный угол поворота

1 = положительный угол поворота

ycode = код y поворота

0 = отрицательный угол поворота

1 = положительный угол поворота

subroutine isometric (ycode,xcode; t(,))

option angle degrees

установить значения направлений поворота

if ycode = 0 then phi = -45

if ycode = 1 then phi = 45

if xcode = 0 then pheta = -35.26439

if xcode = 1 then pheta = 35.26439

выполнить изометрические преобразования, вызвав yrot, xrot и project

call yrot(phi; t)

call xrot(pheta; t)

call project(3; t)

return

oblique

Подпрограмма формирует матрицу косоугольной проекции и конкатенирует ее с текущей матрицей преобразования (уравнение (3-44)).

alpha = угол между горизонталью и осью проекции *z*
f = коэффициент искажения в направлении *z*
t(,) = текущая преобразующая матрица, с которой будет конкатенирована матрица косоугольной проекции

```
subroutine oblique (alpha,f; t())
option angle degrees
dimension oblique(4,4)
Mat oblique = Idn
oblique(3,3) = 0          установить значения преобразующей матрицы
oblique(3,1) = -f*Cos(alpha)
oblique(3,2) = -f*Sin(alpha)
Mat t = t*oblique         конкатенировать матрицы
return
```

perspective

Подпрограмма формирует матрицу центральной перспективной проекции на плоскость *z* = 0 и конкатенирует ее с текущей матрицей преобразования (уравнение (3-47)).

cp = центр проекции на оси *z*
t(,) = текущая преобразующая матрица, с которой будет конкатенирована матрица перспективной проекции

```
subroutine perspective (cp; t())
option angle degrees
dimension perspective(4,4)
Mat perspective = Idn
perspective(3,3) = 0        установить значения матрицы центральной
                           перспективной проекции
perspective(3,4) = -1/cp
Mat t = t*perspective     конкатенировать матрицы
return
```

project

Подпрограмма формирует матрицу простой ортографической проекции на одну из координатных плоскостей и конкатенирует ее с текущей матрицей преобразования (уравнения (3-27)–(3-29)).

proj(,) = матрица проекции
projcode = код проекции
 1 = на плоскость *x* = 0
 2 = на плоскость *y* = 0
 3 = на плоскость *z* = 0
t(,) = текущая преобразующая матрица, с которой будет конкатенирована матрица проекции

```

subroutine project (projcode; t(),)
dimension proj(4,4)
Mat proj = Idn
if projcode = 1 then proj(1,1) = 0 установить соответствующий элемент
if projcode = 2 then proj(2,2) = 0 матрицы равным 0
if projcode = 3 then proj(3,3) = 0
Mat t = t*proj конкатенировать матрицы
return

```

reflt3d

Подпрограмма формирует матрицу зеркального отражения в пространстве и конкатенирует ее с текущей матрицей преобразования (уравнения (3-11)–(3-13)).

refltcode = код отражения
 $1 = \text{относительно плоскости } x = 0$
 $2 = \text{относительно плоскости } y = 0$
 $3 = \text{относительно плоскости } z = 0$
reflt(,) = матрица отражения
 $t(,) = \text{текущая преобразующая матрица, с которой будет}$
 $\text{конкатенирована матрица отражения}$

```

subroutine reflt3d (rltcode; t(),)
dimension reflt(4,4)
Mat reflt = Idn
if rltcode = 1 then reflt(1,1) = -1 установить соответствующий терм
if rltcode = 2 then reflt(2,2) = -1 в матрице отражения равным  $-1$ 
if rltcode = 3 then reflt(3,3) = -1
Mat t = t*reflt конкатенировать матрицы
return

```

scale3d

Подпрограмма формирует матрицу масштабирования в пространстве и конкатенирует ее с текущей матрицей преобразования (уравнения (3-3) и (3-4)).

$a = \text{масштабирующий множитель в направлении } x a > 1 -$
 $\text{растяжение; } a < 1 - \text{сжатие}$
 $e = \text{масштабирующий множитель в направлении } y e > 1 -$
 $\text{растяжение; } e < 1 - \text{сжатие}$
 $j = \text{масштабирующий множитель в направлении } z j > 1 -$
 $\text{растяжение; } j < 1 - \text{сжатие}$
 $s = \text{общий масштабирующий множитель } s > 1 - \text{увеличение;}$
 $s < 1 - \text{сокращение}$
scale = масштабирующая матрица
t(,) = текущая преобразующая матрица, с которой будет
конкатенирована масштабирующая матрица

```
subroutine scale3d (a,e,j,s; t(),)
```

```
option angle degrees
dimension scale(4,4)
Mat scale = Zer
scale(1,1) = a
scale(2,2) = e
scale(3,3) = j
scale(4,4) = 1/s
Mat t = t*scale
return
```

установить значения масштабирующей матрицы

конкатенировать матрицы

tran3d

Подпрограмма формирует матрицу параллельного переноса в пространстве и конкatenирует ее с текущей матрицей преобразования (уравнение (3-14)).

<i>l</i>	= множитель переноса в направлении <i>x</i>
<i>m</i>	= множитель переноса в направлении <i>y</i>
<i>n</i>	= множитель переноса в направлении <i>z</i>
<i>tran(,)</i>	= матрица переноса
<i>t(,)</i>	= текущая преобразующая матрица, с которой будет конкатенирована матрица переноса

subroutine **tran3d(l,m,n; t(,))**

dimension `tran(4,4)`

Mat tran = Idn

$$\text{tran}(4,1) = 1$$

$\text{tran}(4,2) = m$

tran(4,3) = n

XF06

Подпрограмма формирует матрицу поворота в пространстве относительно оси x и конкатенирует ее текущей матрицей преобразования (уравнение (3-6)).

rot(,) = матрица поворота

θ_{roll} — угол поворота в градусах
 $t(\cdot)$ — тангенс преобразования

$t(,)$ = текущая преобразующая матрица, с которой будет конкатенирована матрица поворота

subroutine xrot (theta; t(,))

option angle degrees

dimension rot(4,4)

Mat rot = Idn

$$\text{rot}(2,2) = \text{Cos}(\theta)$$

$$\text{rot}(2,3) = \text{Sin}(\theta)$$

$$\text{rot}(3,3) = \text{rot}(2,2)$$

$$\text{rot}(3,2) = -\text{rot}$$

Mat t =

установка матрицы поворота

конкатенировать матрицы

yrot

Подпрограмма формирует матрицу поворота в пространстве относительно оси y и конкатенирует ее с прикладной матрицей преобразования (уравнение (3-8)).

phi = угол поворота в градусах

rot(,) = матрица поворота

t(,) = текущая преобразующая матрица, с которой будет конкатенирована матрица поворота

subroutine yrot (phi; t(,))

option angle degrees

dimension rot(4,4)

Mat rot = Idn

rot(1,1) = Cos(phi)

установка матрицы поворота

rot(3,1) = Sin(phi)

rot(3,3) = rot(1,1)

rot(1,3) = -rot(3,1)

Mat t = t*rot

конкатенировать матрицы

return

zrot

Подпрограмма формирует матрицу поворота в пространстве относительно оси z и конкатенирует ее с текущей матрицей преобразования (уравнение (3-7)).

rot(,) = матрица поворота

theta = угол поворота в градусах

t(,) = текущая преобразующая матрица, с которой будет конкатенирована матрица поворота

subroutine zrot (theta; t(,))

option angle degrees

dimension rot(4,4)

Mat rot = Idn

rot(1,1) = Cos(theta)

установка матрицы поворота

rot(1,2) = Sin(theta)

rot(2,2) = rot(1,1)

rot(2,1) = -rot(1,2)

Mat t = t*rot

конкатенировать матрицы

return

ГЛАВА 4

Здесь даны алгоритмы, которые генерируют точки на периметре конического сектора. В алфавитном порядке это **circle**, **ellipse**, **hyper 1**, **hyper 2**, **parabola**. Каждый из этих алгоритмов легко модифицируется для генерации точек на отдельной части конического сектора.

circle

Подпрограмма генерирует точки на внешней окружности круга (уравнение (4-5)).

<i>cdtheta</i>	= $\cos(dtheta)$
<i>circ(,)</i>	= массив, содержащий точки плоскости
<i>circ(,1)</i>	= координата <i>x</i>
<i>circ(,2)</i>	= координата <i>y</i>
<i>dtheta</i>	= приращение <i>theta</i>
<i>h</i>	= координата <i>x</i> центра окружности
<i>k</i>	= координата <i>y</i> центра окружности
<i>n</i>	= число точек на окружности
<i>r</i>	= радиус
<i>sdtheta</i>	= $\sin(dtheta)$
<i>tcirc(,)</i>	= массив, содержащий точки плоскости
	<i>tcirc(,1)</i> = координата <i>x</i>
	<i>tcirc(,2)</i> = координата <i>y</i>
	<i>tcirc(,3)</i> = 1, однородная координата
<i>t(,)</i>	= матрица переноса в плоскости 3×3

subroutine circle (h,k,r,n; circ(),)

option angle degrees

dimension t(3,3),tcirc(256,3)

изменение размера, заполнение преобразования и массива tcirc

Mat rot = Idn

Mat tcirc = Zer(n,3)

Mat tcirc = Con(n,3)

определение констант

dtheta = 360/(n - 1)

cdtheta = Cos(dtheta)

sdtheta = Sin(dtheta)

определение начальной точки

tcirc(1,1) = r

tcirc(1,2) = 0

вычисление точек на исходной центрированной окружности

for i = 2 to n

tcirc(i,1) = cdtheta*tcirc(i - 1,1) - sdtheta*tcirc(i - 1,2)

tcirc(i,2) = sdtheta*tcirc(i - 1,1) + cdtheta*tcirc(i - 1,2)

next i

перенос результатов в h, k

call tran2d(h,k; t)

Mat tcirc = tcirc*t

приписывание перенесенных значений в массив 2D

for i = 1 to n

circ(i,1) = tcirc(i,1)

circ(i,2) = tcirc(i,2)

next i

return

ellipse

Подпрограмма генерирует точки на периметре эллипса (уравнение (4-8)).

a = длина большой полуоси
 b = длина малой полуоси
 $c1$ = константа $-(a/b)*cdtheta$
 $c2$ = константа $-(b/a)*sdtheta$
 $cdtheta$ = константа $-\cos(dtheta)$
 $dtheta$ = приращение параметра
 $ellip(,)$ = массив, содержащий точки эллипса
 $ellip(,1)$ = компонента x
 $ellip(,2)$ = компонента y
 h = x координата центра эллипса
 $inclin$ = угол наклона большой оси в градусах
 k = y координата центра эллипса
 n = число точек на эллипсе
 $sdtheta$ = константа $-\sin(dtheta)$
 $tellip(,)$ = 1, массив, содержащий однородные точки пространства
 $tellip(,1)$ = x координата
 $tellip(,2)$ = y координата
 $tellip(,3)$ = 1, однородная координата
 $t(,)$ = матрица преобразования в плоскости 3×3

subroutine ellipse (h,k,a,b,inclin,n; ellip(,))

option angle degrees

dimension t(3,3),tellip(256,3)

изменение размера, заполнение преобразования и массива tellip

Mat t = Idn

Mat tellip = Zer(n,3)

Mat tellip = Con(n,3)

определение констант

$dtheta = 360/(n - 1)$

$cdtheta = \text{Cos}(dtheta)$

$sdtheta = \text{Sin}(dtheta)$

$c1 = (a/b)*sdtheta$

$c2 = (b/a)*sdtheta$

определение начальной точки

$tellip(1,1) = a$

$tellip(1,2) = 0$

вычисление точек на исходном центрированном эллипсе

for i = 2 to n

$tellip(i,1) = tellip(i - 1,1)*cdtheta - c1*tellip(i - 1,2)$

$tellip(i,2) = c2*tellip(i - 1,1) + tellip(i - 1,2)*cdtheta$

next i

поворот на i и перенос результатов в h, k

call rot2d(inclin; t)

call tran2d(h,k; t)

Mat tellip = tellip*t

приписывание перенесенных значений в массив 2D

for i = 1 to n

$ellip(i,1) = tellip(i,1)$

```

ellip(i,2) = tellip(i,2)
next i
return

```

hyper 1

Подпрограмма генерирует точки на гиперболе (уравнение (4-15)). Генерирует только гипербола в первом квадранте, открытая вправо, с осью x в качестве оси симметрии. Полная гипербола может быть получена зеркальным отражением сначала относительно оси x , затем относительно оси y . Произвольные ориентация и центральные позиции могут быть получены соответствующими поворотами и переносами.

<i>a</i>	= расстояние от центра гиперболы до вертикали
<i>b</i>	= определяет наклон асимптот = $\pm b/a$
<i>c1</i>	= константа — $(b)*tdtheta$
<i>c2</i>	= константа — $(a)*cdtheta$
<i>cdtheta</i>	= константа — $\cos(dtheta)$
<i>dtheta</i>	= приращение параметра
<i>hyper(,)</i>	= массив, содержащий точки параболы
	<i>hyper(1,)</i> = x компонента
	<i>hyper(2,)</i> = y компонента
<i>n</i>	= число точек на гиперболе
<i>tdtheta</i>	= константа — $\tan(dtheta)$
<i>sdtheta(,)</i>	= 1, константа — $\sin(dtheta)$

subroutine hyper1 (a,b,n; hyper(,))

изменение размера и заполнение массива *hyper* нулями

Mat *hyper* = **Zer**(*n*,2)

определение констант

dtheta = (**Pi**/2)(*n* - 1)

cdtheta = **Cos**(*dtheta*)

sdtheta = **Sin**(*dtheta*)

tdtheta = **Tan**(*dtheta*)

c1 = (*b*)**tdtheta*

c2 = (*b*)**cdtheta*

определение начальной точки

hyper(1,1) = *a*

hyper(1,2) = 0

вычисление точек на исходной центрированной гиперболе

for *i* = 2 **to** *n*

hyper(*i*,1) = *b***hyper*(*i* - 1,1)/(*c2* - *sdtheta***hyper*(*i* - 1,2))

hyper(*i*,2) = *b**(*hyper*(*i* - 1,2) + *c1*)/(*b* - *hyper*(*i* - 1,2)**tdtheta*)

next *i*

return

hyper 2

Подпрограмма генерирует точки на гиперболе с использованием гиперболических функций (уравнение (4-17)). Генерируется только гипербола в первом квадранте, открытая вправо, с осью x в качестве оси симметрии. Полная гипербола

может быть получена зеркальным отражением сначала относительно оси x , потом относительно оси y . Произвольные ориентации и центральные позиции могут быть получены соответствующими поворотами и переносами.

<i>a</i>	= расстояние от центра гиперболы до вертикали
<i>b</i>	= определяет наклон асимптот = $\pm b/a$
<i>c1</i>	= константа $-(a/b)*sdtheta$
<i>c2</i>	= константа $-(b/a)*sdtheta$
<i>cdtheta</i>	= константа $-cosh(dtheta)$
<i>dtheta</i>	= приращение параметра
<i>hyper(,)</i>	= массив, содержащий точки параболы $hyper(1) = x$ компонента $hyper(2) = y$ компонента
<i>n</i>	= число точек на гиперболе
<i>sdtheta</i>	= константа $-sinh(dtheta)$
<i>thetamax(,)</i>	= максимальное значение параметра

subroutine hyper2 (a,b,thetamax,n; hyper(),)

изменение размера и заполнение массива *hyper* нулями

Mat hyper = Zer(n,2)

определение констант

dtheta = *thetamax*(n - 1)

cdtheta = **Cosh**(*dtheta*)

sdtheta = **Sinh**(*dtheta*)

c1 = (*a/b*)**sdtheta*

c2 = (*b/a*)**sdtheta*

определение начальной точки

hyper(1,1) = *a*

hyper(1,2) = 0

вычисление точек на исходной центрированной гиперболе

for i = 2 to n

hyper(i,1) = *hyper*(i - 1,1)**cdtheta* + *c1***hyper*(i - 1,2)

hyper(i,2) = *c2***hyper*(i - 1,1) + *hyper*(i - 1,2)**cdtheta*)

next i

return

parabola

Подпрограмма генерирует в первом квадранте точки на параболе, симметричной относительной оси x , открытой вправо (уравнение (4-12)). Для генерации полной параболы в пространстве, расположенной в произвольной позиции и произвольной ориентации, применяют необходимые преобразования зеркального отражения, поворота и параллельного переноса.

<i>a</i>	= расстояние от фокуса до вертикали параболы
----------	--

<i>a1</i>	= константа — $a*dtheta*dtheta$
-----------	---------------------------------

<i>b1</i>	= константа $2 * a * dtheta$
-----------	------------------------------

<i>dtheta</i>	= приращение параметра
---------------	------------------------

<i>n</i>	= число точек на параболе
----------	---------------------------

<i>para(,)</i>	= массив, содержащий точки параболы
----------------	-------------------------------------

para,(1) = координата x

para,(2) = координата y

<i>thetamax</i> = максимальное значение параметра

```

subroutine parabola (a,thetamax,n; para(),)
изменение размера и заполнение массива para нулями
Mat para = Zer(n,2)
определение констант
dtheta = thetamax/(n - 1)
a1 = a*dtheta*dtheta
b1 = 2*a*dtheta
определение начальной точки
para(1,1) = 0
para(1,2) = 0
вычисление точек на исходной параболе
for i = 2 to n
    para(i,1) = para(i - 1,1) + dtheta*para(i - 1,2) + a1
    para(i,2) = para(i - 1,2) + b1
next i
return

```

ГЛАВА 5

Здесь приведены алгоритмы реализации методов построения оболочек. Алгоритмы даны для кубического сплайна, параболически смешанных кривых и кривых В-сплайна. Алгоритмы каркасного представления кривых даны для кривых Безье, для рациональных и нерациональных кривых В-сплайна, однородных и неоднородных узловых векторов и для базисных функций В-сплайна: **basis**, **bezier**, **bsplfit**, **bspline**, **bsplineu**, **cblend**, **cspline**, **dbasis**, **dbasisu**, **dbezier**, **dbspline**, **dbsplineu**, **knot**, **knotc**, **knotu**, **matpbspl**, **nmatrix**, **parabld**, **param**, **rbasis**, **rbspline**, **rbspliu**.

basis

Подпрограмма генерирует базисные функции В-сплайна для открытых неоднородных узловых векторов (уравнение (5-84)).

<i>c</i>	= порядок базисной функции В-сплайна
<i>d</i>	= первый терм рекурсивного отношения базисной функции
<i>e</i>	= второй терм рекурсивного отношения базисной функции
<i>npts</i>	= число вершин определяющего многоугольника
<i>n()</i>	= массив, содержащий базисные функции <i>n(1,1)</i> содержит базисную функцию, соответствующую B_1 и т. д.
<i>nplusc</i>	= константа — <i>npts</i> + <i>c</i> — максимальное значение номера узла
<i>t</i>	= значение параметра
<i>temp()</i>	= временный массив
<i>x()</i>	= узловой вектор

```

subroutine basis (c,t,npts,x(); n(),)
dimension temp(20)                               допускает 20 вершин многоугольника
nplusc = npts + c
вычисление основных функций  $N_{i,1}$  (см. уравнения (5-84a))
for i = 1 to nplusc - 1

```

```

if t >= x(i) and t < x(i + 1) then
    temp(i) = 1
else
    temp(i) = 0
end if
next i
вычисление основных функций высшего порядка (см. уравнения (5-84б))
for k = 2 to c
    for i = 1 to nplusc - k
        if temp(i) ≠ 0 then   если основная функция равна нулю, пропустите вычисление
            d=((t-x(i))*temp(i))/(x(i+k-1)-x(i))
        else
            d=0
        end if
        if temp(i+1)≠0 then  если основная функция равна нулю, пропустите вычисление
            e = ((x(i + k) - t)*temp(i + 1))/(x(i + k) - x(i + 1))
        else
            e = 0
        end if
        temp(i)=d+e
    next i
next k
if t = x(nplusc) then temp(npts) = 1           начните с последней точки
вставить в массив
for i = 1 to npts
    n(1,i) = temp(i)
next i
if t = x(nplusc) then n(1,npts) = 1           начните с последней точки
return

```

bezier

Подпрограмма вычисляет кривую Безье (уравнение (5-62)).

b(,) = массив, содержащий вершины определяющего многоугольника
b(,1) содержит компоненту *x* вершины
b(,2) содержит компоненту *y* вершины
b(,3) содержит компоненту *z* вершины

Basis = функция для вычисления базисного значения Бернштейна (уравнение (5-63))

cpts = число точек, которые будут вычислены на кривой

Factrl = функции для вычисления факториала числа

j(,) = базисная функция Бернштейна

Ni = функции факториала для базисной функции Бернштейна

npts = число вершин определяющего многоугольника

p(,) = массив, содержащий точки кривой

p(,1) содержит компоненту *x* точки

p(,2) содержит компоненту *y* точки

p(,3) содержит компоненту *z* точки

t = значение параметра $0 \leq t \leq 1$

subroutine bezier (npts,b(,),cpts; p(,))

defNi(n,i)=Factrl(n)/Factrl(i)*Factrl(n-i))

функция факториал

```

def Basis(n,i,t) = Ni(n,i)*(t&i)*((1 -t) ∧(n - i))      базисная функция Бернштейна
dimension j(1,20)                                         допускает 20 вершин многоугольника
dimension temp(1,3)
icount = 0
Mat j = Zer(1,npts)
for t = 0 to 1 step 1/(cpts - 1)
    icount = icount + 1
    определение основной функции Бернштейна (см. уравнение (5-63))
    for i = 1 to npts
        j(1,i) = Basis(npts - 1,i - 1,t)
    next i
    определение точки на кривой
    Mat temp = j*b
    вставитъ в массив
    for i = 1 to 3
        p(icount,i) = temp(1,i)
    next i
mext t
return

```

bsplfit

Подпрограмма выравнивает кривую В-сплайна, используя открытый однородный узловой вектор (уравнение (5-117)).

b(,) = массив, содержащий вершины определяющего многоугольника
b(,1) содержит компоненту x вершины
b(,2) содержит компоненту y вершины
b(,3) содержит компоненту z вершины
dpts = число точек данных
d(,) = массив, содержащий точки данных
d(,1) содержит x компоненту точки данных
d(,2) содержит y компоненту точки данных
d(,3) содержит z компоненту точки данных
k = порядок базисной функции В-сплайна
n = матрица базисной функции
nbasis = массив, содержащий базисные функции для единственного значения t
ninv = обращение $trn(n) \times n$
nplusc = число узловых значений
npts = число вершин определяющего многоугольника
ntemp = временная матрица для хранения $trn(n) \times n$
ntmp = временная матрица для хранения обращения $trn(n) \times n \times d$
ntrn = транспонирование матрицы n
t = параметрическое значение $0 \leq t \leq 1$
tpar() = массив, содержащий аппроксимацию хордами для параметрических значений
x() = массив, содержащий узловой вектор

```

subroutine bsplfit (dpts,d(,),npts,k; b(,))
dimension nbasis(1,20),x(30),n(20,20),temp(1,3)      допускает данные о 20 точках
dimension tpar(20),ninv(20,20),ntrn(20,20),ntemp(20,20),ntmp(20,3)
изменение размера и заполнение матриц нулями

```

```

Mat nbasis = Zer(1,npts)
Mat x = Zer(npts + k)
Mat n = Zer(dpts,npts)
Mat ntrn = Zer(npts,dpts)
Mat ntemp = Zer(npts,npts)
Mat nttmp = Zer(npts,2)
Mat ninv = Zer(npts,npts)
Mat tpar = Zer(dpts)
call knot(npts,k; x())
call param(dpts,d; tpar),
nplusc = npts + k
генерация матрицы основных функций
for i = 1 to dpts
    t = tpar(i)*x(nplusc)           вычисление значения параметра для одной строки
    call basis(k,t,npts,x; nbasis)   вычисление основной функции для одной строки
    for j = 1 to npts
        n(i,j) = nbasis(1,j)         построение матрицы строка за строкой
    next j
next i
генерация вершин определяющего многоугольника методом наименьших квадратов
Mat ntrn = trn(n)                 нахождение транспонированной матрицы основной функции
Mat ntemp = ntrn*n                trn(n) × n
Mat ninv = Inv(ntemp)             обращение матрицы trn(n) × n
Mat nttmp = ntrn*d                обращение матрицы trn(n) × n × d
Mat b = ninv*nttmp               вычисление вершин определяющего многоугольника
return

```

bspline

Подпрограмма генерирует кривую В-сплайна, используя открытый однородный узловой вектор (уравнение (5-83)).

b() = массив, содержащий вершины определяющего многоугольника
b(,1) содержит компоненту *x* вершины
b(,2) содержит компоненту *y* вершины
b(,3) содержит компоненту *z* вершины
k = порядок базисной функции В-сплайна
nbasis = массив, содержащий базисные функции для единственного значения
plusc = число узловых значений
npts = число вершин определяющего многоугольника
p() = массив, содержащий точки кривой
p(,1) содержит компоненту *x* точки
p(,2) содержит компоненту *y* точки
p(,3) содержит компоненту *z* точки
p1 = число вычисляемых точек кривой
t = параметрическое значение $0 \leq t \leq 1$
x() = массив, содержащий вектор

subroutine bspline (npts,k,p1,b(,); p(),)

dimension nbasis(1,20),x(30),temp(1,3)

nplusc = *npts* + *k*

допускает 20 вершин многоугольников с базисной функцией порядка 5

обнулить и изменить размерность узлового вектора и базисного массива

Mat nbasis = Zer(1,npts)

Mat x = Zer(nplusc)

генерация однородного открытого узлового вектора

call knot(npts,k; x)

icount = 0

вычисление точек на кривой B-сплайна

for t = 0 to x(npts + k) step x(npts + k)/(p1 - 1)

icount = icount + 1

call basis(k,t,npts,x; nbasis)

генерация базисной функции для этого значения t

Mat temp = nbasis*b

генерация точки кривой

p(icount,1) = temp(1,1)

присваивание текущего значения точки кривой

p(icount,2) = temp(1,2)

массиву этой кривой

p(icount,3) = temp(1,3)

next t

return

bsplineu

Подпрограмма генерирует кривую B-сплайна, используя периодический однородный узловой вектор (уравнение (5-83)).

b(,) = массив, содержащий вершины определяющего многоугольника

b(,1) содержит компоненту x вершины

b(,2) содержит компоненту y вершины

b(,3) содержит компоненту z вершины

k = порядок базисной функции B-сплайна

nbasis = массив, содержащий базисные функции для единственного значения t

nplusc = число узловых значений

npts = число вершин определяющего многоугольника

p(,) = массив, содержащий точки кривой

p(,1) содержит компоненту x точки

p(,2) содержит компоненту y точки

p(,3) содержит компоненту z точки

p1 = число вычисляемых точек кривой

t = параметрическое значение $0 \leq t \leq 1$

x() = массив, содержащий узловой вектор

subroutine bspline (npts,k,p1,b(,); p(,))

dimension nbasis(1,20),x(30),temp(1,3)

*допускает 20 вершин многоугольника
с базисной функцией порядка 5*

обнулить и изменить размерность узлового вектора и базисного массива

Mat nbasis = Zer(1,npts)

Mat x = Zer(npts + k)

nplusc = npts + k

генерация однородного периодического узлового вектора

call knotu(npts,k; x)

icount = 0

вычисление точек на кривой B-сплайна

for t = (k - 1) to (npts - 1 + 1) step (npts - 1 + 1) - (k - 1)/(p1 - 1)

```

icount = icount + 1
call basis(k,t,npnts,x; nbasis)
Mat temp = nbasis*b
p(icount,1) = temp(1,1)
p(icount,2) = temp(1,2)
next t
return

```

генерация базисной функции для этого значения t
 генерация точки на кривой
 присваивание текущего значения точки на кривой
 массиву этой кривой

cblend

Подпрограмма вычисляет нормализованные кубические смешанные функции (уравнение (5-21)).

$f(,)$ = массив 1×4 , содержащий смешанные функции
 $f(1,1)$ содержит F_1
 $f(1,2)$ содержит F_2
 $f(1,3)$ содержит F_3
 $f(1,4)$ содержит F_4

τ = значение нормализованного параметра $0 \leq \tau \leq 1$

```

subroutine cblend (tau; f(,))
f(1,1) = 2* (tau)^ 3 - 3*(tau)^ 2 + 1
f(1,2) = -2* (tau)^ 3 + 3*(tau)^ 2
f(1,3) = tau* ((tau)^ 2 - 2*tau + 1)
f(1,4) = tau* ((tau)^ 2 - tau)
return

```

cspline

Подпрограмма вычисляет кривую кубического сплайна (уравнение (5-1)).

$bcflag$ = флаг граничного условия
 $bcflag$ = 1 – условия закрепленных концов
 $bcflag$ = 2 – условия свободных концов
 $d(,)$ = массив, содержащий подходящие точки данных
 $d(,1)$ содержит компоненту x точки данных
 $d(,2)$ содержит компоненту y точки данных
 $d(,3)$ содержит компоненту z точки данных
 $f(,)$ = массив 1×4 , содержащий смешанные функции
 $f(1,1)$ содержит F_1
 $f(1,2)$ содержит F_2
 $f(1,3)$ содержит F_3
 $f(1,4)$ содержит F_4
 g = геометрическая матрица
 $icount$ = счетчик числа точек на кривой
 $m(,)$ = матрица коэффициентов $[M]$ (уравнение (5-15))
 $minv(,)$ = обращение матрицы коэффициентов $[M]$
 $npnts$ = число подходящих точек
 $p(,)$ = массив, содержащий точки кривой
 $p(,1)$ содержит компоненту x точки
 $p(,2)$ содержит компоненту y точки
 $p(,3)$ содержит компоненту z точки

`point(,)` = единственная точка кривой
`point(1,1)` содержит компоненту x точки
`point(1,2)` содержит компоненту y точки
`point(1,3)` содержит компоненту z точки
`p1` = число точек, вычисляемых для каждого сегмента кривой
`r(,)` = матрица [R] для умножения справа (уравнение (5-15))
`t(,)` = матрица параметра, аппроксимированного хордами
`tangent(,)` = матрица касательного вектора (уравнение (5-15))
`tau` = значение нормализованного параметра $0 \leq \tau \leq 1$

```

subroutine cspline (npts,p1,d(,),bcflag,icount,tangent(,); p(,))

dimension t(20),m(20,20),minv(20,20),r(20,3)           допускает данные для 20 точек
dimension g(4,3),point(1,3),f(1,4)

Mat m = Zer(npts,npts)
Mat minv = Zer(npts,npts)
Mat r = Zer(npts,3)
Mat t = Zer(npts)
определение значений параметра с использованием хордовой аппроксимации
(уравнение (5-2))
for i = 2 to npts
    t(i) = Sqrt((d(i,1)-d(i - 1,1)) $\wedge$ 2+(d(i,2)- d(i - 1,2)) $\wedge$  2)
next i
вставить коэффициенты матрицы [M] (уравнение (5-15))
for i = 2 to npts - 1
    m(i,i - 1) = t(i + 1)
    m(i,i) = 2*(t(i) + t(i + 1))
    m(i,i + 1) = t(i)
next i
вставить первую строку матрицы [R] (уравнение (5-15))
for i = 2 to npts - 1
    coeff = (3/(t(i)*t(i + 1)))
    for j = 1 to 3
        r(i,j) = coeff*((t(i) $\wedge$ 2)*(d(i + 1,j) - d(i,j)) + (t(i + 1) $\wedge$ 2)*(d(i,j) - d(i - 1,j)))
    next j
    next i
добавить граничные условия в матрицы (табл. 5-3)
bcflag = 1 — зафиксировать концевые условия
bcflag = 2 — погасить концевые условия
циклические и антициклические граничные условия рассмотреть как упражнение

if bcflag = 1 then
    m(1,1) = 1
    m(npts,npts) = 1
    for j = 1 to 3
        r(1,j) = tangent(1,j)
        r(npts,j) = tangent(npts,j)
    next j
end if
if bcflag = 2 then
    m(1,1) = 1
    m(1,2) = 1/2
    m(npts,npts - 1) = 2

```

```

m(npts,npts) = 4
for j = 1 to 3
    r(i,j) = (3/2*t(2)))*(d(2,j) - d(1,j))
    r(npts,j) = (6/t(npts))*(d(npts,j) - d(npts - 1,j))
next j
end if
вставить коэффициенты матриц [M] (уравнение (5-16))
Mat minv = Inv(m)
вычислить внутренние касательные векторы для [M]-1[R]
Mat tangent = minv*r
генерировать точки на кубическом сплайне (уравнение (5-22))
ограничить интервал для предупреждения вычислений точки на конце интервала,
дважды убрать эту последнюю точку позже
icount = 0
for i = 2 to npts
    for tau = 0 to 1 - (1/(p1 - 1)) step 1/(p1 - 1)      p1>1 или ошибка в результате
        icount = icount + 1
        установить матрицу смешанной функции (уравнение (5-21))
        f(1,1) = -2*(tau^3) + 3*(tau^2)+1
        f(1,2) = 2*(tau^3) - 3*(tau^2)
        f(1,3) = tau*(tau^2 - 2*tau + 1)*t(i)
        f(1,4) = tau*(tau^2 - tau)*t(i)
        установить геометрическую матрицу (уравнение (5-20))
        for j = 1 to 3
            g(1,j) = d(i - 1,j)
            g(2,j) = d(i,j)
            g(3,j) = tangent(i - 1,j)
            g(4,j) = tangent(i,j)
        next j
        вычислить единственную точку на кривой (уравнение (5-22))
        Mat point = f*g
        приписать векторную матрицу
        for j = 1 to 3
            p(icount,j) = point(1,j)
        next j
    next tau
next i
сохранить специально последнюю точку
icount = icount + 1
for j = 1 to 3
    p(icount,j) = d(npts,j)
next j
return

```

dbasis

Подпрограмма генерирует базисные функции В-сплайна и их производные для однородных открытых узловых векторов (уравнения (5-84) и (5-97)–(5-100)).

$b1$ = первый терм базисной функции
 $b2$ = второй терм базисной функции
 c = порядок базисной функции В-сплайна
 $d1()$ = массив, содержащий производную базисной функции
 $d1(1,1)$ содержит производную базисной функции для B_1 и т. д.
 $d2()$ = массив, содержащий производную базисной функции
 $d2(1,1)$ содержит производную базисной функции для B_1 и т. д.
 $f1$ = первый терм первой производной базисной функции
 $f2$ = второй терм первой производной базисной функции
 $f3$ = третий терм первой производной базисной функции
 $f4$ = четвертый терм первой производной базисной функции
 $npts$ = число вершин определяющего многоугольника
 $n()$ = массив, содержащий базисные функции
 $n(1,1)$ содержит базисную функцию для B_1 и т. д.
 $nplusc$ = содержит максимальное узловое значение
 $s1$ = первый терм второй производной базисной функции
 $s2$ = второй терм второй производной базисной функции
 $s3$ = третий терм второй производной базисной функции
 $s4$ = четвертый терм второй производной базисной функции
 t = параметрическое значение
 $temp()$ = временный массив
 $x()$ = узловой вектор

```

subroutine dbasis (c,t,npts,x(); n(),d1(),d2(),)
dimension temp(20),temp1(20),temp2(20)           допускает 20 вершин многоугольника
nplusc = npts + c
заполнение нулями временных массивов
Mat temp = Zer(npts + c)
Mat temp1 = Zer(npts + c)
Mat temp2 = Zer(npts + c)
вычислить базисные функции первого порядка  $n(i,j)$  (уравнения (5-84a))
for i = 1 to nplusc - 1
  if t  $\geq$  x(i) and t < x(i + 1) then
    temp(i) = 1
  else
    temp(i) = 0
  end if
next i
if t = x(nplusc) then temp(npts) = 1           обработать конец специально
вычислить базисные функции высшего порядка и их производные
(уравнения (5-84b) и (5-97)–(5-100))
for k = 2 to c
  for i = 1 to nplusc - k
    вычислить базисную функцию
    if temp(i)  $\neq$  0 then           если базисная функция равна нулю, прервать вычисление
      b1 = ((t - x(i))*temp(i))/(x(i + k - 1) - x(i))
    else
  
```

```

b1 = 0
end if
if temp(i + 1) ≠ 0 then если базисная функция равна нулю, прервать вычисление
  b2 = ((x(i + k) - t)*temp(i + 1))/(x(i + k) - x(i + 1))
else
  b2 = 0
end if
вычислить первую производную
if temp(i) ≠ 0 then если базисная функция равна нулю, прервать вычисление
  f1 = temp(i)/(x(i + k - 1) - x(i))
else
  f1 = 0
end if
if temp(i + 1) ≠ 0 then если базисная функция равна нулю, прервать вычисление
  f2 = -temp(i + 1)/(x(i + k) - x(i))
else
  f2 = 0
end if
if temp1(i) ≠ 0 then если базисная функция равна нулю, прервать вычисление
  f3 = (t - x(i))*temp1(i)/(x(i + k - 1) - x(i + 1))
else
  f3 = 0
end if
if temp1(i + 1) ≠ 0 then если базисная функция равна нулю, прервать вычисление
  f4 = (x(i + k) - t)*temp1(i + 1)/(x(i + k) - x(i + 1))
else
  f4 = 0
end if
вычислить вторую производную
if temp1(i) ≠ 0 then если базисная функция равна нулю, прервать вычисление
  s1 = 2*temp1(i)/(x(i + k - 1) - x(i))
else
  s1 = 0
end if
if temp1(i + 1) ≠ 0 then если базисная функция равна нулю, прервать вычисление
  s2 = -2*temp1(i + 1)/(x(i + k) - x(i + 1))
else
  s2 = 0
end if
if temp2(i) ≠ 0 then если базисная функция равна нулю, прервать вычисление
  s3 = (t - x(i))*temp2(i)/(x(i + k - 1) - x(i))
else
  s3 = 0
end if
if temp2(i + 1) ≠ 0 then если базисная функция равна нулю, прервать вычисление
  s4 = (x(i + k) - t)*temp2(i + 1)/(x(i + k) - x(i + 1))
else
  s4 = 0
end if
temp(i) = b1 + b2
temp1(i) = f1 + f2 + f3 + f4
temp2(i) = s1 + s2 + s3 + s4
next i

```

```

next k
вставьте в массив
for i = 1 to npts
     $n(1,i) = \text{temp}(i)$ 
     $d1(1,i) = \text{temp1}(i)$ 
     $d2(1,i) = \text{temp2}(i)$ 
next i
return

```

dbasisu

Подпрограмма генерирует базисные функции B-сплайна и их производные для однородных периодических узловых векторов (уравнения (5-84) и (5-97)–(5-100)).

b_1	= первый терм базисной функции
b_2	= второй терм базисной функции
c	= порядок базисной функции B-сплайна
$d1(,)$	= массив, содержащий производную базисных функций $d1(1,1)$ содержит производную базисной функции для B_1 и т. д.
$d2(,)$	= массив, содержащий производную базисных функций $d2(1,1)$ содержит производную базисной функции для B_1 и т. д.
f_1	= первый терм первой производной базисной функции
f_2	= второй терм первой производной базисной функции
f_3	= третий терм первой производной базисной функции
f_4	= четвертый терм первой производной базисной функции
$npts$	= число вершин определяющего многоугольника
$n(,)$	= массив, содержащий базисные функции $n(1,1)$ содержит базисную функцию для B_1 и т. д.
$nplusc$	= содержит максимальное узловое значение
s_1	= первый терм второй производной базисной функции
s_2	= второй терм второй производной базисной функции
s_3	= третий терм второй производной базисной функции
s_4	= четвертый терм второй производной базисной функции
t	= параметрическое значение
$\text{temp}()$	= временный массив
$x()$	= узловой вектор

```

subroutine dbasisu (c,t,npts,x(); n(),d1(),d2(),)
dimension temp(20),temp1(20),temp2(20) до 20 вершин определяющего многоугольника
nplusc = npts + c
заполняем нулями временный массив
Mat temp = Zer(nplusc)
Mat temp1 = Zer(nplusc)
Mat temp2 = Zer(nplusc)
вычисляем базисные функции первого порядка  $N_{i,1}$  (уравнения (5-84a))
for i = 1 to nplusc - 1
    if  $t \geq x(i)$  and  $t < x(i + 1)$  then
         $\text{temp}(i) = 1$ 
    else
         $\text{temp}(i) = 0$ 
    end if
next i

```

```

if t = x(npts + 1) then                               отдельно определяем конец путем перестановки
    temp(npts) = 1                                         базисных функций первого порядка
    temp(npts + 1) 0
end if
вычисляем базисные функции высших порядков и их производные
уравнения (5-84б) и (5-97)-(5-100)
for k = 2 to c
    for i = 1 to nplusc - k
        вычисляем базисную функцию
        if temp(i) ≠ 0 then                                если базисная функция равна нулю,
            b1 = ((t - x(i))*temp(i))/(x(i + k - 1) - x(i)) пропускаем вычисления
        else
            b1 = 0
        end if
        if temp(i + 1) ≠ 0 then                            если базисная функция равна нулю,
            b2 = ((x(i + k) - t)*temp(i + 1))/(x(i + k) - x(i + 1)) пропускаем вычисления
        else
            b2 = 0
        end if
        вычисляем первую производную
        if temp(i) ≠ 0 then                                если базисная функция равна нулю,
            f1 = temp(i)/(x(i + k - 1) - x(i)) пропускаем вычисления
        else
            f1 = 0
        end if
        if temp(i + 1) ≠ 0 then                            если базисная функция равна нулю,
            f2 = -temp(i + 1)/(x(i + k) - x(i)) пропускаем вычисления
        else
            f2 = 0
        end if
        if temp1(i) ≠ 0 then                                если базисная функция равна нулю,
            f3 = (t - x(i))*temp1(i)/(x(i + k - 1) - x(i + 1)) пропускаем вычисления
        else
            f3 = 0
        end if
        if temp1(i + 1) ≠ 0 then                            если базисная функция равна нулю,
            f4 = (x(i + k) - t)*temp1(i + 1)/(x(i + k) - x(i + 1)) пропускаем вычисления
        else
            f4 = 0
        end if
        вычисляем вторую производную
        if temp1(i) ≠ 0 then                                если базисная функция равна нулю,
            s1 = 2*temp1(i)/(x(i + k - 1) - x(i)) пропускаем вычисления
        else
            s1 = 0
        end if

```

```

if temp1(i + 1) ≠ 0 then           если базисная функция равна нулю,
                                    пропускаем вычисления
    s2 = -2*temp1(i + 1)/(x(i + k) - x(i + 1))
else
    s2 = 0
end if
if temp2(i) ≠ 0 then             если базисная функция равна нулю,
                                    пропускаем вычисления
    s3 = (t - x(i))*temp2(i)/(x(i + k - 1) - x(i))
else
    s3 = 0
end if
if temp2(i + 1) ≠ 0 then           если базисная функция равна нулю,
                                    пропускаем вычисления
    s4 = (x(i + k) - t)*temp2(i + 1)/(x(i + k) - x(i + 1))
else
    s4 = 0
end if
temp(i) = b1 + b2
temp1(i) = f1 + f2 + f3 + f4
temp2(i) = s1 + s2 + s3 + s4
next i
next k
помещаем в массив
for i = 1 to npts
    n(1,i) = temp(i)
    d1(1,i) = temp1(i)
    d2(1,i) = temp2(i)
next i
return

```

dbezier

Подпрограмма вычисляет кривую Безье и ее первую и вторую производные (уравнения (5-62), (5-72) и (5-73)).

b(,) = массив, содержащий вершины определяющего многоугольника

b(,1) содержит компоненту x вершины

b(,2) содержит компоненту y вершины

b(,3) содержит компоненту z вершины

Basis = функция для вычисления базисного значения Бернштейна

cpts = число точек, которые будут вычислены на кривой

d1(,) = массив, содержащий первую производную кривой

d1(,1) содержит компоненту x производной

d1(,2) содержит компоненту y производной

d1(,3) содержит компоненту z производной

d2(,) = массив, содержащий вторую производную кривой

d2(,1) содержит компоненту x производной

d2(,2) содержит компоненту y производной

d2(,3) содержит компоненту z производной

Factrl = функция для вычисления факториала числа

j(,) = базисная функция Бернштейна

j1(,) = первая производная базисной функции Бернштейна

j2(,) = вторая производная базисной функции Бернштейна

N_i = функция факториала для базиса Бернштейна
 $npts$ = число вершин определяющего многоугольника
 $p(,)$ = массив, содержащий точки кривой
 p(,1) содержит компоненту x точки
 p(,2) содержит компоненту y точки
 p(,3) содержит компоненту z точки
 t = значение параметра $0 \leq t \leq 1$

```

subroutine dbezier (npts,b(,),cpts; p(,),d1(,),d2(,))
def Ni(n,i) = Factrl(n)/(Factrl(i)*Factrl(n - i))           функция факториала
def Basis(n,i,t) = Ni(n,i)*(t^i)*((1 - t)^(n - i))      базисная функция Бернштейна
dimension j(1,20),j1(1,20),j2(1,20)                      допускает 20 вершин многоугольника
dimension temp(1,3),temp1(1,3),temp2(1,3)
определяем размерности и заполняем нулями матрицы
Mat j = Zer(1,npts)
Mat j1 = Zer(1,npts)
Mat j2 = Zer(1,npts)
for t = 0 to 1 step 1/(cpts - 1)
    icount = icount + 1
    if icount = cpts then t = 1           необходимо организовать вычисление приращения t
    определяем базисную функцию Бернштейна и ее первую и вторую производные
    (уравнения (5-63), (5-74), (5-75))
    for i = 1 to npts
        j(1,i) = Basis(npts - 1, i - 1,t)
        if t ≠ 0 and t ≠ 1 then           специально устанавливаем конечные точки
            j1(1,i) = (((i - 1) - (npts - 1)*t)/(t*(1 - t)))*j(1,i)
            j2(1,i) = ((i - 1) - (npts - 1)*t)^2 - (npts - 1)*t*t - (i - 1)*(1 - 2*t)
            j2(1,i) = j(1,i)*2*(1,i)/(t*t*(1 - t)*(-t))
        end if
    next i
определяем точку кривой
Mat temp = j*b
определяем производные кривой первого и второго порядков (уравнения (5-62), (5-72)
(5-73))
if t <> 0 and t ≠ 1 then           специально устанавливаем концевые точки
    Mat temp1 = j1*b
    Mat temp2 = j2*b
else
    if t = 0 then
        temp1(1,1) = (npts - 1)*(b(2,1) - b(1,1))
        temp1(1,2) = (npts - 1)*(b(2,2) - b(1,2))
    end if
    if t = 0 then
        temp2(1,1) = (npts - 1)*(npts - 2)*(b(1,1) - 2*b(2,1) + b(3,1))
        temp2(1,2) = (npts - 1)*(npts - 2)*(b(1,2) - 2*b(2,2) + b(3,2))
    end if
    if t = 1 then
        temp1(1,1) = (npts - 1)*(b(npts,1) - b(npts - 1,1))
        temp1(1,2) = (npts - 1)*(b(npts,2) - b(npts - 1,2))
    end if
    if t = 1 then
        temp2(1,1) = (b(npts,1) - 2*b(npts - 1,1) + b(npts - 2,1))
    end if

```

```

temp2(1,1) = (npts - 1)*(npts - 2)*temp2(1,1)
temp2(1,2) = (b(npts,2) - 2*b(npts - 1,2) + b(npts - 2,2))
temp2(1,2) = (npts - 1)*(npts - 2)*temp2(1,2)
end if
end if
помещаем в массив
for i = 1 to 3
    p(icount,i) = temp(1,i)
    d1(icount,i) = temp1(1,i)
    d2(icount,i) = temp(1,i)
next i
next t
return

```

dbspline

Подпрограмма генерирует кривую В-сплайна и ее производные, используя открытый однородный узловой вектор (уравнения (5-83), (5-95) и (5-96)).

<i>b(,)</i>	= массив, содержащий вершины определяющего многоугольника <i>b(,1)</i> содержит компоненту <i>x</i> вершины <i>b(,2)</i> содержит компоненту <i>y</i> вершины <i>b(,3)</i> содержит компоненту <i>z</i> вершины
<i>d1(,)</i>	= массив, содержащий первую производную кривой <i>d1(,1)</i> содержит компоненту <i>x</i> производной <i>d1(,2)</i> содержит компоненту <i>y</i> производной <i>d1(,3)</i> содержит компоненту <i>z</i> производной
<i>d2(,)</i>	= массив, содержащий вторую производную кривой <i>d2(,1)</i> содержит компоненту <i>x</i> производной <i>d2(,2)</i> содержит компоненту <i>y</i> производной <i>d2(,3)</i> содержит компоненту <i>z</i> производной
<i>d1nbasis(,)</i>	= первая производная базисной функции для единственного значения <i>t</i>
<i>d2nbasis(,)</i>	= вторая производная базисной функции для единственного значения <i>t</i>
<i>k</i>	= порядок базисной функции В-сплайна
<i>nbasis</i>	= массив, содержащий базисные функции для единственного значения <i>t</i>
<i>nplusc</i>	= число узловых значений
<i>npts</i>	= число вершин определяющего многоугольника
<i>p(,)</i>	= массив, содержащий точки кривой <i>p(,1)</i> содержит компоненту <i>x</i> точки <i>p(,2)</i> содержит компоненту <i>y</i> точки <i>p(,3)</i> содержит компоненту <i>z</i> точки
<i>p1</i>	= число вычисляемых точек кривой
<i>t</i>	= параметрическое значение $0 \leq t \leq 1$
<i>x()</i>	= массив, содержащий узловой вектор

subroutine dbspline (npts,k,p1,b(,); p(,),d1(,),d2(,))

допускает 20 вершин многоугольника с базисной функцией порядка 5

dimension x(30),nbasis(1,20),d1nbasis(1,20),d2nbasis(1,20)

dimension temp(1,3),temp1(1,3),temp2(1,3)

определяем размерности и заполняем нулями узловой вектор

и массивы базиса, кривой и производных

Mat p = Zer(p1,3)

Mat d1 = Zer(p1,3)

```

Mat d2 = Zer(p1,3)
Mat nbasis = Zer(1,npts)
Mat d1nbasis = Zer(1,npts)
Mat d2nbasis = Zer(1,npts)
Mat x = Zer(npts + k)
nplusc = npts + k
генерация однородного открытого узлового вектора
call knot(npts,k; x)
icount = 0
вычисляем точки на кривой B-сплайна и их первую и вторую производные
for t = 0 to x(nplusc) step x(nplusc)/(p1 - 1)
    icount = icount + 1
    if icount = p1 then t = x(nplusc)      компенсация для вычисления приращения  $t$ 
    генерация базисной функции и ее производных для этого значения  $t$ 
    (уравнения (5-84) и (5-97)–(5-100))
    call dbasis(k,t,npts,x; nbasis,d1nbasis,d2nbasis)
    Mat temp = nbasis*b
    Mat temp 1 = d1nbasis*b
    Mat temp 2 = d2nbasis*b
    p(icount,1) = temp(1,1)           генерация точки на кривой
    p(icount,2) = temp(1,2)           первая производная в этой точке на кривой
    d1(icount,1) = temp1(1,1)         вторая производная в этой кривой
    d1(icount,2) = temp1(1,2)         сохраняем текущее значение точки на кривой
    d2(icount,1) = temp2(1,1)         в массиве этой кривой
    d2(icount,2) = temp2(1,2)         сохраняя текущее значение производной в этой точке
                                        на кривой в массиве производной
                                        сохраняя текущее значение производной в этой точке
                                        на кривой в массиве производной
                                        на кривой в массиве производной
                                        на кривой в массиве производной
next t
return

```

dbsplineu

Подпрограмма генерирует кривую В-сплайна и ее производные, используя периодический однородный узловой вектор (уравнения (5-83), (5-95) и (5-96)).

b(,)	= массив, содержащий вершины определяющего многоугольника b(,1) содержит компоненту x вершины b(,2) содержит компоненту y вершины b(,3) содержит компоненту z вершины
d1(,)	= массив, содержащий первую производную кривой d1(,1) содержит компоненту x производной d1(,2) содержит компоненту y производной d1(,3) содержит компоненту z производной
d2(,)	= массив, содержащий вторую производную кривой d2(,1) содержит компоненту x производной d2(,2) содержит компоненту y производной d2(,3) содержит компоненту z производной
d1nbasis(,)	= первая производная базисной функции для единственного значения t
d2nbasis(,)	= вторая производная базисной функции для единственного значения t
k	= порядок базисной функции B-сплайна
nbasis	= массив, содержащий базисные функции для единственного значения t
nplusc	= число узловых значений
npts	= число вершин определяющего многоугольника
p(,)	= массив, содержащий точки кривой

$p(,1)$ содержит компоненту x точки
 $p(,2)$ содержит компоненту y точки
 $p(,3)$ содержит компоненту z точки
 $p1 =$ число вычисляемых точек кривой
 $t =$ параметрическое значение $0 \leq t \leq 1$
 $x()$ = массив, содержащий узловой вектор

subroutine dbsplineu (npts,k,p1,b(,); p(,),d1(,),d2(,))
допускает 20 вершин многоугольника с базисной функцией порядка 5
dimension x(30),nbasis(1,20),d1nbasis(1,20),d2nbasis(1,20)
dimension temp(1,3),temp1(1,3),temp2(1,3)
определляем размерности и заполняем нулями узлового вектор
и массивы базиса, кривой и производных массива

Mat p = **Zer**(p1,3)
Mat d1 = **Zer**(p1,3)
Mat d2 = **Zer**(p1,3)
Mat nbasis = **Zer**(1,npts)
Mat d1nbasis = **Zer**(1,npts)
Mat d2nbasis = **Zer**(1,npts)
Mat x = **Zer**(npts + k)

nplsc = npts + k

генерация периодического однородного открытого узлового вектора

call knotu(npts,k; x)

icount = 0

вычисляем точки на кривой В-сплайна и их первую и вторую производные
(уравнения (5-84), (5-93), (5-94))

for t = (k - 1) **to** npts **step** (npts - 1)/(p1 - 1)

icount = icount + 1

if icount = p1 **then** t = npts

компенсация для вычисления приращения t

генерация базисной функции и ее производных для этого значения t

call dbasisu(k,t,npts,x; nbasis,d1nbasis,d2nbasis)

Mat temp = nbasis*b

генерация точки на кривой

Mat temp 1 = d1nbasis*b

первая производная в этой точке на кривой

Mat temp 2 = d2nbasis*b

вторая производная в этой точке на кривой

p(icount,1) = temp(1,1)

сохраняем текущее значение точки на кривой

p(icount,2) = temp(1,2)

в массиве кривой

d1(icount,1) = temp1(1,1)

сохраняем текущее значение производной в этой точке

d1(icount,2) = temp1(1,2)

на кривой в массиве производной

d2(icount,1) = temp2(1,1)

сохраняем текущее значение производной в этой точке

d2(icount,2) = temp2(1,2)

на кривой в массиве производной

next t

return

knot

Подпрограмма генерирует открытый узловой вектор В-сплайна с коэффициентом на концах (с. 315).

c = порядок базисной функции

n = число вершин определяющего многоугольника

nplus2 = индекс массива x() для первого приближения максимального значения узлового вектора
nplusc = максимальное значение узлового вектора – n + c
x() = массив, содержащий узловой вектор

```
subroutine knot (n,c; x())
nplusc = n + c
nplus2 = n + 2
x(1) = 0
for i = 2 to nplusc
  if i > c and i < nplus2 then
    x(i) = x(i - 1) + 1
  else
    x(i) = x(i - 1)
  end if
next i
return
```

knotc

Подпрограмма генерирует неоднородный открытый узловой вектор, пропорциональный длинам хорд между вершинами определяющего многоугольника (уравнение (5-86)).

<i>b</i>	= массив, содержащий вершины определяющего многоугольника
<i>b(,1)</i>	содержит компоненту <i>x</i> вершины
<i>b(,2)</i>	содержит компоненту <i>y</i> вершины
<i>b(,3)</i>	содержит компоненту <i>z</i> вершины
<i>c</i>	= порядок базисной функции
<i>chord</i>	= хордовое расстояние между вершинами определяющего многоугольника
<i>csimt</i>	= накопитель суммы хордовых расстояний
<i>maxchord</i>	= сумма хордовых расстояний между вершинами определяющего многоугольника
<i>npts</i>	= число вершин определяющего многоугольника
<i>nplusc</i>	= максимальное значение узлового вектора – n + c
<i>nnumeratop</i>	= нумератор уравнения (5-86)
<i>x()</i>	= массив, содержащий узловой вектор
<i>xchord</i>	= компонента <i>x</i> расстояния между вершинами определяющего многоугольника
<i>ychord</i>	= компонента <i>y</i> расстояния между вершинами определяющего многоугольника

```
subroutine knotc (npts,c,b(,); x())
```

```
dimension chord(20)
```

```
nplusc = npts + c
```

```
n = npts - 1
```

определяем размерности и заполняем нулями узловой вектор и значения хорд

```
Mat chord = Zer(n)
```

определение хордового расстояния между вершинами определяющего многоугольника и их суммы

```
maxchord = 0
```

```
for i = 2 to npts
```

```

xchord = b(i,1) - b(i - 1,1)
ychord = b(i,2) - b(i - 1,2)
chord(i - 1) = Sqrt(xchord*xchord + ychord*ychord)
maxchord = maxchord + chord(i - 1)
next i
заводим к нулей в начале открытого узлового вектора
for i = 1 to c
  x(i) = 0
next i
генерация внутренних узловых значений
for i = 1 to n - c + 1
  csum = 0
  for j = 1 to i
    csum = csum + chord(j)
  next j
  numerator = (i/(n - c + 2))*chord(i + 1) + csum
  x(c + i) = (numerator/maxchord)*(n - c + 2)
next i
заводим к нулей в конце открытого узлового вектора
for i = n + 2 to nplusc
  x(i) = n - c + 2
next i
return

```

knotu

Подпрограмма генерирует периодический однородный открытый узловой вектор В-сплайна (с. 311).

c = порядок базисной функции
n = число вершин определяющего многоугольника
nplusc = максимальное значение узлового вектора — *n* + *c*
x() = массив, содержащий узловой вектор

```

subroutine knotu (n,c; x())
nplusc = n + c
for i = 1 to nplusc
  x(i) = i - 1
next i
return

```

matpbspl

Подпрограмма генерирует кривую В-сплайна с использованием матричных методов и периодических однородных узловых векторов (уравнение (5-93)).

b(,) = массив, содержащий вершины определяющего многоугольника в скользящем окне
b(,1) содержит компоненту *x* вершины в скользящем окне
b(,2) содержит компоненту *y* вершины в скользящем окне
b(,3) содержит компоненту *z* вершины в скользящем окне

$d(,)$ = массив, содержащий вершины определяющего многоугольника
 $d(,1)$ содержит компоненту x вершины
 $d(,2)$ содержит компоненту y вершины
 $d(,3)$ содержит компоненту z вершины
 $fcoeff$ = коэффициент для целочисленной матрицы
 k = порядок базисной функции B-сплайна
 $nbasis$ = массив, содержащий базисные функции для единственного значения t
 $nplusc$ = число узловых значений
 $npts$ = число вершин определяющего многоугольника
 $p(,)$ = массив, содержащий точки кривой
 $p(,1)$ содержит компоненту x точки
 $p(,2)$ содержит компоненту y точки
 $p(,3)$ содержит компоненту z точки
 $p1$ = число вычисляемых точек кривой
 $ptemp(,)$ = временная матрица, содержащая единственную точку кривой $-t * n * b$
 $t(,)$ = параметрическая матрица
 $temp(,)$ = временная матрица $-t * n$
 u = нормализованное параметрическое значение $0 \leq u \leq 1$

```

subroutine matpbspl (k,npts,p1,d(,); p(,))
dimension t(1,9),b(9,3),p(9,9),temp(1,9),ptemp(1,3)           допускает 9-й порядок
определяем размерность и заполняем нулями матрицы
Mat b = Zer(k,3)
Mat n = Zer(k,k)
Mat t = Zer(1,k)
Mat temp = Zer(1,k)
устанавливаем значения матрицы [N]
call nmatrix(k; fcoeff,n)
устанавливаем значения матрицы скользящих вершин многоугольника
и вычисляем точки
icount = 0
for j = 0 to npts                                         устанавливаем значения матрицы
for 1 = 0 to k - 1                                         скользящих вершин многоугольника
    b(1 + 1,1) = d(mod(j + 1,npts) + 1,1)
    b(1 + 1,2) = d(mod(j + 1,npts) + 1,2)
    b(1 + 1,3) = d(mod(j + 1,npts) + 1,3)
next 1
for u = 0 to 1 - (1/(p1 - 1)) step 1/(p1 - 1)
    icount = icount + 1
    for i = 1 to k                                         устанавливаем значения параметрической матрицы
        t(1,i) = u^(k - i)
    next i
    Mat t = fcoeff*t                                         вычисляем точки на этом сегменте
    Mat temp = t*m
    Mat ptemp = temp*b
    p(icount,1) = ptemp(1,1)                               присваиваем текущее значение точки
    p(icount,2) = ptemp(1,2)                               на кривой массиву кривой
    p(icount,3) = ptemp(1,3)
next u
next j
return

```

nmatrix

Подпрограмма вычисляет общую матрицу периодического базиса В-сплайна (уравнение (5-91)).

Factrl = функция, вычисляющая факториал числа
fcoeff = коэффициент целочисленной матрицы $-1/(k-1)!$
k = порядок периодической базисной функции
n() = целочисленная форма матрицы
Ni(n, i) = функция, вычисляющая $\binom{n}{i}$

```
subroutine nmatrix (k; fcoeff,n())
def Ni(n,i) = Factrl(n)/Factrl(i)*Factrl(n-i)
определяем размерность и заполняем нулями матрицу [N]
Mat n = Zer(k,k)
fcoeff = 1/Factrl(k - 1) вычисляем постоянный коэффициент
устанавливаем значения матрицы
for i = 0 to k - 1
    temp = Ni(k - 1,i)
    for j = 0 to k - 1
        sum = 0
        for l = j to k - 1
            sum1 = (k - (l + 1))li
            sum2 = (-1)(l - j)
            sum3 = Ni(k,l - j)
            sum = sum + sum1*sum2*sum3
        next l
        n(i + 1, j + 1) = temp*sum
    next j
next i
return
```

parabld

Подпрограмма выполняет параболическое смешивание alpha \neq beta, используя аппроксимацию хордами (уравнения (5-44) и (5-6)).

Предполагается, что эта программа в каждый момент времени обеспечивается четырьмя точками данных. Вычисляются точки, расположенные внутри интервала вдоль параболически смешанной кривой.

a() = матрица 4×4 параболически смешанной функции
alpha = значение параметра *r* при P_2
beta = значение параметра *s* при P_3
c() = массив, содержащий точки смешанной кривой
 c(,1) содержит компоненту *x* точки на кривой
 c(,2) содержит компоненту *y* точки на кривой
 c(,3) содержит компоненту *z* точки на кривой
c12 = хордовое расстояние между точками 1 и 2
c23 = хордовое расстояние между точками 2 и 3
c34 = хордовое расстояние между точками 3 и 4

`icount = счетчик числа точек на смешанной кривой`
`inter() = промежуточная матрица [A][G]`
`p() = массив, содержащий четыре точки данных`
`p(,1) содержит компоненту x точки данных`
`p(,2) содержит компоненту y точки данных`
`p(,3) содержит компоненту z точки данных`
`point() = единственная точка смешанной кривой`
`point(,1) содержит компоненту x точки`
`point(,2) содержит компоненту y точки`
`point(,3) содержит компоненту z точки`
`p1 = число точек, вычисляемых на этом интервале`
`t() = кубическая параметрическая матрица`
`t1 = параметрическое значение вдоль смешанной кривой`

subroutine parabld (p(), c(), p1, icount)
dimension a(4,4), t(1,4), inter(4,3), point(1,3)
вычисляем длины хорд
 $c12 = \text{Sqrt}((p(2,1) - p(1,1))^2 + (p(2,2) - p(1,2))^2)$
 $c23 = \text{Sqrt}((p(3,1) - p(2,1))^2 + (p(3,2) - p(2,2))^2)$
 $c34 = \text{Sqrt}((p(4,1) - p(3,1))^2 + (p(4,2) - p(3,2))^2)$
вычисляем alpha и beta
 $\alpha = c12 / (c23 + c12)$
 $\beta = c23 / (c23 + c34)$
вычисляем [A]
 $a(1,1) = -((1 - \alpha)^2) / \alpha$
 $a(2,1) = 2 * ((1 - \alpha)^2) / \alpha$
 $a(3,1) = a(1,1)$
 $a(4,1) = 0$
 $a(1,2) = ((1 - \alpha) + \alpha * \beta) / \alpha$
 $a(2,2) = -(2 * (1 - \alpha) + \alpha * \beta) / \alpha$
 $a(3,2) = (1 - 2 * \alpha) / \alpha$
 $a(4,2) = 1$
 $a(1,3) = (-(1 - \alpha) - \alpha * \beta) / (1 - \beta)$
 $a(2,3) = (2 * (1 - \alpha) - \beta * (1 - 2 * \alpha)) / (1 - \beta)$
 $a(3,3) = \alpha$
 $a(4,3) = 0$
 $a(1,4) = (\beta^2) / (1 - \beta)$
 $a(2,4) = -a(1,4)$
 $a(3,4) = 0$
 $a(4,4) = 0$
вычисляем матрицу [A]/[G]
`Mat inter = a*p`
определяем точки на сегменте кривой
`for t1 = 0 to 1 - (1/(p1 - 1)) step 1/(p1 - 1)`
устанавливаем значения матрицы [T]
 $t(1,1) = t1^3$
 $t(1,2) = t1^2$
 $t(1,3) = t1$
 $t(1,4) = 1$
вычисляем точки на смешанной кривой и запоминаем их в C(t)
`Mat point = t*inter`

```

icount = icount + 1
for i = 1 to 3
    c(icount,i) = point(1,i)
next i
next t1
return

```

param

Подпрограмма вычисляет хордовые значения, базирующиеся на хордовых расстояниях.

d() = массив, содержащий точки данных
dpts = число точек данных
isum = накопитель суммы хордовых расстояний
sum = сумма хордовых расстояний
tparm() = массив, содержащий параметрические значения

```

subroutine param (dpts,d(),; tparm())
sum = 0
isum = 0
tparm(1) = 0
вычисляем сумму хордовых расстояний для всех точек данных
for i = 2 to dpts
    sum = sum + Sqrt((d(i,1) - d(i - 1,1))^2 + (d(i,2) - d(i - 1,2))^2)
next i
вычисляем параметрические значения
for i = 2 to dpts
    isum = isum + Sqrt((d(i,1) - d(i - 1,1))^2 + (d(i,2) - d(i - 1,2))^2)
    tparm(i) = isum/sum
next i
return

```

rbasis

Подпрограмма генерирует базисные функции рационального В-сплайна для открытого узлового вектора (уравнение (5-123)).

c = порядок базисной функции В-сплайна
d = первый терм рекурсивного отношения базисной функции
e = второй терм рекурсивного отношения базисной функции
h() = массив, содержащий однородные координатные весовые множители
npts = число вершин определяющего многоугольника
plusc = константа – *npts* + *c* – максимальное число узловых значений
r() = массив, содержащий производную базисных функций
 r(1,1) содержит производную базисной функции для B_1 и т. д.
sum = сумма произведений нерациональных базисных функций
 и однородных координатных весовых множителей
t = параметрическое значение
temp() = временный массив
x() = узловой вектор

```

subroutine rbasis (c,t,npts,x(),h(); r(),)
dimension temp(20)                               допускает 20 вершин многоугольника
nplusc = npts + c
вычисляем нерациональные базисные функции первого порядка  $N_{i,1}$  (уравнение (5-84a))
for i = 1 to nplusc - 1
  if t  $\geq$  x(i) and t < x(i + 1) then
    temp(i) = 1
  else
    temp(i) = 0
  end if
next i
вычисляем нерациональные базисные функции высшего порядка (уравнение (5-84b))
for k = 2 to c
  for i = 1 to nplusc - k
    if temp(k)  $\neq$  0 then                      если нуль, пропускаем вычисление
      d = ((t - x(i))*temp(i))/(x(i + k - 1) - x(i))
    else
      d = 0
    end if
    if temp(i + 1)  $\neq$  0 then                  если нуль, пропускаем вычисление
      e = ((x(i + k) - t)*temp(i + 1))/(x(i + k) - x(i + 1))
    else
      e = 0
    end if
    temp(i) = d + e
  next i
next k
if t = x(nplusc) then temp(npts) = 1          добавляем последнюю точку
вычисляем сумму для знаменателя рациональных базисных функций (уравнение (5-123))
sum = 0
for i = 1 to npts
  sum = sum + temp(i)*h(i)
next i
образуем рациональные базисные функции и помещаем их в вектор r (уравнение (5-123))
for i = 1 to npts
  if sum  $\neq$  0 then
    r(1,i) = temp(i)*h(i)/sum
  else
    r(1,i) = 0
  end if
next i
return

```

rbspline

Подпрограмма генерирует кривую рационального В-сплайна, используя открытый однородный узловый вектор (уравнения (5-122) и (5-124)).

b(,) = массив, содержащий вершины определяющего многоугольника
b(,1) содержит компоненту x вершины
b(,2) содержит компоненту y вершины
b(,3) содержит компоненту z вершины

$h()$ = массив, содержащий однородные координатные весовые множители
 k = порядок базисной функции В-сплайна
 $nbasis$ = массив, содержащий базисные функции для единственного значения t
 $nplusc$ = число узловых значений
 $npts$ = число вершин определяющего многоугольника
 $p()$ = массив, содержащий точки кривой
 $p(,1)$ содержит компоненту x точки
 $p(,2)$ содержит компоненту y точки
 $p(,3)$ содержит компоненту z точки
 $p1$ = число вычисляемых точек кривой
 t = параметрическое значение $0 \leq t \leq 1$
 $x()$ = массив, содержащий узловой вектор

subroutine rbspline (npts,k,p1,b(),h(); p(),)
dimension nbasis(1,20),x(30),temp(1,3)

допускает 20 заданных точек
с базисной функцией порядка 5

$nplusc = npts + k$
 определяет размерность и заполняет нулями узловой вектор и базисный массив
 $Mat nbasis = Zer(1,npts)$
 $Mat x = Zer(nplusc)$
 генерирует открытый однородный узловой вектор
 $call knot(npts,k; x)$
 $icount = 0$
 вычисляем точки на кривой рационального В-сплайна
 $for t = 0 to x(npts + k) step x(npts + k)/(p1 - 1)$
 $icount = icount + 1$
 $call rbasis(k,t,npts,x,h; nbasis)$ рациональные базисные функции
для этого значения t
 $Mat temp = nbasis*b$
 $p(icount,1) = temp(1,1)$ генерируем текущее значение точки на кривой
 $p(icount,2) = temp(1,2)$
 $p(icount,3) = temp(1,3)$ массиву кривой
 $next t$
 $return$

rbsplinu

Подпрограмма генерирует кривую рационального В-сплайна, используя периодический однородный узловой вектор (уравнения (5-122) и (5-124)).

$b()$ = массив, содержащий вершины определяющего многоугольника
 $b(,1)$ содержит компоненту x вершины
 $b(,2)$ содержит компоненту y вершины
 $b(,3)$ содержит компоненту z вершины
 $h()$ = массив, содержащий однородные координатные весовые множители
 k = порядок базисной функции В-сплайна
 $nbasis$ = массив, содержащий базисные функции для единственного значения t
 $nplusc$ = число узловых значений
 $npts$ = число вершин определяющего многоугольника
 $p()$ = массив, содержащий точки кривой
 $p(,1)$ содержит компоненту x точки
 $p(,2)$ содержит компоненту y точки
 $p(,3)$ содержит компоненту z точки

p1 = число вычисляемых точек кривой
t = параметрическое значение
x() = массив, содержащий узловой вектор

```

subroutine rbsplinu (npts,k,p1,b(,),h(); p())
dimension nbasis(1,20),x(30),temp(1,3)
dimension nbasis(1,20),x(30),temp(1,3)           допускает 20 заданных точек
с базисной функцией порядка 5

nplusc = npts + k
определяем размерности и заполняем нулями узловой вектор и базисный массив
Mat nbasis = Zer(1,npts)
Mat x = Zer(nplusc)
генерируем открытый однородный вектор
call knotu(npts,k; x)
icount = 0
вычисляем точки на кривой рационального В-сплайна
for t = (k - 1) to (npts - 1 + 1) step ((npts - 1 + 1) - (k - 1))/(p1 - 1)
  icount = icount + 1
  call rbasis(k,t,npts,x,h; nbasis)           рациональные базисные функции
                                                для этого значения t
  Mat temp = nbasis*b
  p(icount,1) = temp(1,1)
  p(icount,2) = temp(1,2)
  p(icount,3) = temp(1,3)           генерируем точку на кривой
присваиваем текущее значение точки на кривой
массиву этой кривой

next t
return

```

ГЛАВА 6

Здесь представлены алгоритмы генерации трехмерных поверхностей. Включены алгоритмы для билинейных, правильных поверхностей, линейных и бикубических поверхностей Кунса, поверхностей Безье, рационального и нерационального В-сплайна. В алфавитном порядке представлены следующие алгоритмы:

bezsurf

Подпрограмма вычисляет поверхность Безье (уравнение (6-58)).

b(,) = массив, содержащий вершины определяющего многоугольника
b(,1) содержит компоненту *x* вершины
b(,2) содержит компоненту *y* вершины
b(,3) содержит компоненту *z* вершины
Замечание: массив $B_{i,j} = b(,)$ имеет размерность $n * m \times 3$, причем *j* изменяется быстрее
Basis = функция для вычисления базисного значения Вернштейна
Factrl = функция для вычисления факториала числа
jin = базисная функция Вернштейна в направлении *u* (уравнение (5-63))
kjm = базисная функция Вернштейна в направлении *w* (уравнение (5-63))
m = число, на единицу меньше, чем число вершин многоугольника в направлении *w*

n = число, на единицу меньшее, чем число вершин многоугольника в направлении и
 Ni = функции факториала для базисной функции Бернштейна
 $p1$ = число параметрических линий в направлении и
 $p2$ = число параметрических линий в направлении w
 $q()$ = координатные векторы для точек на поверхности
 q(,1) содержит компоненту x вектора
 q(,2) содержит компоненту y вектора
 q(,3) содержит компоненту z вектора
 для фиксированного значения и следующие т элеменитов содержит
 значения для кривой $q(u_i, w)$; q имеет размерность $p1*p2 \times 3$

```

subroutine bezsurf (b(),n,m,p1,p2; q())
def Ni(n,i) = Factrl(n)/(Factrl(i)*Factrl(n - i))
def Basis(n,i,t) = Ni(n,i)*(t^i)*((1 - t)^(n - i))
icount = 0
for u = 0 to 1 step 1/(p1 - 1)           для фиксированного и вычисляем различные w
  for w = 0 to 1 step 1/(p2 - 1)
    icount = icount + 1
    for i = 0 to n
      jin = Basis(n,i,u)           базисная функция Бернштейна в направлении и
      for j = 0 to m
        j1 = (m + 1)*i + (j + 1)
        kjm = Basis(m,j,w)         базисная функция Бернштейна в направлении w
        for k = 1 to 3
          q(icount,k) = q(icount,k) + b(j1,k)*jin*kjm   вычисляет точку поверхности
          next k
        next j
      next i
    next w
  next u
return

```

bicubic

Подпрограмма генерирует бикубическую поверхность Кунса (уравнение (6-51)).

$n()$ = матрица смешанной функции 4×4 (уравнение (6-31))
 $p()$ = матрица граничных условий 4×4 (уравнение (6-36))
 $p1$ = число точек вдоль направления и
 $p2$ = число точек вдоль направления w
 $q()$ = матрица, содержащая координатные векторы для бикубической поверхности,
 имеет размерность $p1*p2 \times 3$
 первый столбец – компонента x
 второй столбец – компонента y
 третий столбец – компонента z
 первые $p2$ элементов для $u = 0$
 вторые $p2$ элементов для $u = 1/(p1-1)$ и т. д.
 $u()$ = кубический смешанный вектор 1×4 (уравнение (6-37))
 $w()$ = кубический смешанный вектор 4×1 (уравнение (6-37))
 $x()$ = массив 4×4 , содержащий компоненты x матрицы
 граничных условий p (уравнение (6-36))

$y(,) = \text{массив } 4 \times 4, \text{ содержащий компоненты у матрицы}$
 $\text{граничных условий } p \text{ (уравнение (6-36))}$

$z(,) = \text{массив } 4 \times 4, \text{ содержащий компоненты } z \text{ матрицы}$
 $\text{граничных условий } p \text{ (уравнение (6-36))}$

subroutine bicubic (p1,p2,x(,),y(,),z(,); q(,))
dimension p(4,4),n(4,4),ntrn(4,4),u(1,4),w(4,1)
dimension temp1(4,4),temp2(4,4),temp3(1,4),temp4(1,1)
определяем размерности матриц и заполняем нулями эти матрицы

Mat p = Zer(4,4)
Mat n = Zer(4,4)
Mat u = Zer(1,4)
Mat w = Zer(4,1)
Mat q = Zer(p1*2,3)
Mat temp1 = Zer(4,1)
Mat temp2 = Zer(4,1)
Mat temp3 = Zer(4,1)
Mat temp4 = Zer(1,1)

устанавливаем значения n-матрицы

$n(1,1) = 2$
 $n(1,2) = -2$
 $n(1,3) = 1$
 $n(1,4) = 1$
 $n(2,1) = -3$
 $n(2,2) = 3$
 $n(2,3) = -2$
 $n(2,4) = -1$
 $n(3,3) = 1$
 $n(4,1) = 1$

транспонируем n-матрицу Mat ntrn = trn(n)

for k = 1 to 3

 icount = 0

if k = 1 then Mat p = x
if k = 2 then Mat p = y
if k = 3 then Mat p = z

Mat temp1 = p*ntrn

Mat temp2 = n*temp1

for u1 = 0 to 1 step 1/(p1 - 1)

(1,1) = u1*u1*u1

(1,2) = u1*u1

(1,3) = u1

(1,4) = 1

for w1 = 0 to 1 step 1/(p2 - 1)

 icount = icount + 1

 w(1,1) = w1*w1*w1

 w(2,1) = w1*w1

 w(3,1) = w1

 w(4,1) = 1

Mat temp3 = u*temp2

Mat temp4 = temp3*w

q(icount,k) = temp4(1,1)

*вычисляем p * ntrn-константу для всех u, w*

*вычисляем n * p * ntrn-константу для всех u, w*

устанавливаем значения n-матрицы

устанавливаем значения w-матриц

*вычисляем u * n * ntrn*

вычисляем q(u, w)

запоминаем q(u, w)

```

next w1
next u1
next k
return

```

bilinear

Подпрограмма вычисляет билинейную поверхность (уравнение (6-41)).

n = число приращений в *u*

m = число приращений в *w*

p(,) = содержит координаты угловых точек

p имеет размерность 4×3 . Первый столбец содержит

компоненты *x*, второй — компоненты *y*, а третий — компоненты *z* координатных векторов углов.

q(,) = содержит координаты интерполируемой поверхности *z*

q имеет размерность *m* * *n* * 3

первые *m* координат соответствуют константе *u* = 0

вторые *m* координат соответствуют константе *u* = $1/(n - 1)$

третьи *m* координат соответствуют константе *u* = $2/(n - 1)$ и т. д.

subroutine bilinear (n,m,p(,); q(,))

Mat *q* = Zer(*n***m*,3) определяем размерность массива *q* и заполняем его нулями

icount = 0

вычисляем поверхность

for *u* = 0 to 1 step $1/(n - 1)$

цикл установки константы *u*

for *w* = 0 to 1 step $1/(m - 1)$

цикл установки константы *w*

icount = *icount* + 1

увеличиваем счетчик точек

q(icount,1) = *p(1,1)**(1 - *u*)*(1 - *w*) + *p(2,1)**(1 - *u*)**w* + *p(3,1)***u**(1 - *w*)

q(icount,1) = *q(icount,1)* + *p(4,1)***u***w*

q(icount,2) = *p(1,2)**(1 - *u*)*(1 - *w*) + *p(2,2)**(1 - *u*)**w* + *p(3,2)***u**(1 - *w*)

q(icount,2) = *q(icount,2)* + *p(4,2)***u***w*

q(icount,3) = *p(1,3)**(1 - *u*)*(1 - *w*) + *p(2,3)**(1 - *u*)**w* + *p(3,3)***u**(1 - *w*)

q(icount,3) = *q(icount,3)* + *p(4,3)***u***w*

next *w*

next *u*

return

bsplsurf

Подпрограмма вычисляет декартово произведение В-сплайн поверхности, используя открытые однородные узловые векторы (уравнение (6-58)).

b() = массив, содержащий точки сетки многоугольника

b(,1) содержит компоненту *x* вершины

b(,2) содержит компоненту *y* вершины

b(,3) содержит компоненту *z* вершины

Замечание: $B_{i,j} = b(,)$ имеет размерность $n * m \times 3$,

причем *j* изменяется быстрее. Размер сетки многоугольника: *n* * *m*

k = порядок в направлении *u*

l = порядок в направлении *w*

mbasis(,) = нерациональные базисные функции для одного значения *w* (уравнение (5-84))

mpts = число вершин многоугольника в направлении *w*

n_{basis} = нерациональные базисные функции для одного значения u (уравнение (5-84))
 $npts$ = число вершин многоугольника в направлении u
 p_1 = число параметрических линий в направлении u
 p_2 = число параметрических линий в направлении w
 $q()$ = массив, содержащий результатирующую поверхность
 q(,1) содержит компоненту x вектора
 q(,2) содержит компоненту y вектора
 q(,3) содержит компоненту z вектора
 для фиксированного значения u следующие p_1 элементы массива q имеют размерность $p_2 \times 3$.
 значения для кривой $q(u_i, w)$ q имеет размерность $p_1 \times p_2$.
 Отображается поверхность размера $p_1 \times p_2$

subroutine bsplsurf (b(),k,l,npts,mpts,p1,p2; q(),)

допускает 20 заданных точек с базисной функцией порядка 5

dimension x(30),y(30),nbasis(1,20),mbasis(1,20)

определяем размерности и заполняем массивы нулями

Mat x = Zer(npts + k)

Mat y = Zer(mpts + 1)

Mat nbasis = Zer(1,npts)

Mat mbasis = Zer(1,mpts)

Mat q = Zer(p1*p2,3)

генерируем открытые однородные узловые векторы

call knot(npts,k; x())

вычисляем угловой вектор и

call knot(mpts,l; y())

вычисляем угловой вектор w

nplusc = npts + k

mplusc = mpts + 1

icount = 0

вычисляем точки на В-сплайн поверхности

for u = 0 to x(nplusc) step x(nplusc)/(p1 - 1)

базисная функция для этого значения u

call basis(k,u,npts,x; nbasis)

for w = 0 to y(mplusc) step y(mplusc)/(p2 - 1)

базисная функция для этого значения w

call basis(l,w,mpts,y; mbasis)

icount = icount + 1

for i = 1 to npts

for j = 1 to mpts

j1 = mpts*(i - 1) + j

for s = 1 to 3

q(icount,s) = q(icount,s) + b(j1,s)*nbasis(1,i)*mbasis(1,j)

next s

next j

next i

next w

next u

return

bspsurfu

Подпрограмма вычисляет декартово произведение для В-сплайн поверхности, используя периодические однородные узловые векторы (уравнения (6-58)).

b() = массив, содержащий точки сетки многоугольника


```
next u
return
```

coonslin

Подпрограмма вычисляет линейную поверхность Кунса (уравнение (6-49)).

$a(,)$ = массив размера $p(u, 0)$, содержащий точки на кривой $n \times 3$
 $b(,)$ = массив размера $p(u, 1)$, содержащий точки на кривой $n \times 3$
 $c(,)$ = массив размера $p(0, w)$, содержащий точки на кривой $m \times 3$
 $d(,)$ = массив размера $p(1, w)$, содержащий точки на кривой $m \times 3$
 n = число точек на граничных кривых $p(u, 0)$ и $p(u, 1)$
 m = число точек на граничных кривых $p(0, w)$ и $p(1, w)$
 $q(,)$ = массив размера $n * m \times 3$. Формат дан для фиксированного значения u , каждые следующие
 m строк матрицы q содержат значения для переменного w .

subroutine coonslin (n,m,a(,),b(,),c(,),d(,); q(,))

Mat q = Zer(n*m,3)

определяем размерность матрицы q
и заполняем нулями эту матрицу

вычисляем поверхность

for $u = 0$ to 1 step $1/(m - 1)$
for $w = 0$ to 1 step $1/(n - 1)$

icount = icount + 1

for $l = 1$ to 3

$q(\text{icount}, l) = a(k, l)*(1 - w) + b(k, l)*w + c(j, l)*(1 - u) + d(j, l)*u$
 $q(\text{icount}, l) = q(\text{icount}, l) - a(1, l)*(1 - u)*(1 - w) - b(1, l)*(1 - u)*w$
 $q(\text{icount}, l) = q(\text{icount}, l) - a(n, l)*u*(1 - w) - b(n, l)*u*w$

next l

next w

next u

return

dbicubic

Подпрограмма вычисляет бикубическую поверхность Кунса и ее производные (уравнения (6-51) и (6-53)–(6-57)).

$du(,)$ = смешанный вектор производной размера 1×4 (уравнение (6-53))
 $dw(,)$ = смешанный вектор производной размера 4×1 (уравнение (6-54))
 $n(,)$ = матрица смешанной функции размера 4×4 (уравнение (6-26))
 $p(,)$ = матрица граничных условий размера 4×4 (уравнение (6-51))
 $p1$ = число точек вдоль направления u
 $p2$ = число точек вдоль направления w
 $q(,)$ = матрица размера $p1*p2 \times 3$, содержащая
координатные векторы для бикубической поверхности
первый столбец – компонента x
второй столбец – компонента y
третий столбец – компонента z
первые $p2$ элементов для $u = 0$
вторые $p2$ элементов для $u = 1/(p1 - 1)$
все производные размещаются подобным образом

$du(,)$ = матрица размера $p1 * p2 \times 3$,
 содержащая векторы производной и для бикубической поверхности
 $dw(,)$ = матрица размера $p1 * p2 \times 3$,
 содержащая векторы производной w для бикубической поверхности
 $duu(,)$ = матрица размера $p1 * p2 \times 3$,
 содержащая векторы производной ui для бикубической поверхности
 $duw(,)$ = матрица размера $p1 * p2 \times 3$,
 содержащая векторы производной uw для бикубической поверхности
 $dww(,)$ = матрица размера $p1 * p2 \times 3$,
 содержащая векторы производной ww для бикубической поверхности
 $u(,)$ = кубический смешанный вектор размера 1×4 (уравнение (5-26))
 $w(,)$ = кубический смешанный вектор размера 4×1 (уравнение (5-26))
 $x(,)$ = массив 4×4 , содержащий x компоненты матрицы
 граничных условий p (уравнение (6-36))
 $y(,)$ = массив 4×4 , содержащий y компоненты матрицы
 граничных условий p (уравнение (6-36))
 $z(,)$ = массив 4×4 , содержащий z компоненты матрицы
 граничных условий p (уравнение (6-36))

subroutine dbicubic (p1,p2,x(,),y(,),z(,); q(,),qu(,),qw(,),quu(,),quw(,),qww(,))
dimension p(4,4),n(4,4),ntrn(4,4),u(1,4),w(4,1)
dimension du(1,4),dw(4,1),duu(1,4),dww(1,4)
dimension temp1(4,4),temp2(4,4),temp3(1,4),temp4(4,1)
dimension temp5(4,1),temp6(4,1),temp7(4,1)
dimension dtemp1(1,1),dtemp2(1,1),dtemp3(1,1),dtemp4(1,1),dtemp5(1,1)

определяем размерность матриц и заполняем их нулями

$\text{Mat } p = \text{Zer}(4,4)$
 $\text{Mat } n = \text{Zer}(4,4)$
 $\text{Mat } u = \text{Zer}(1,4)$
 $\text{Mat } w = \text{Zer}(4,1)$
 $\text{Mat } q = \text{Zer}(p1*p2,3)$
 $\text{Mat } qu = \text{Zer}(p1*p2,3)$
 $\text{Mat } qw = \text{Zer}(p1*p2,3)$
 $\text{Mat } quu = \text{Zer}(p1*p2,3)$
 $\text{Mat } quw = \text{Zer}(p1*p2,3)$
 $\text{Mat } qww = \text{Zer}(p1*p2,3)$
 $\text{Mat } temp1 = \text{Zer}(4,4)$
 $\text{Mat } temp2 = \text{Zer}(4,4)$
 $\text{Mat } temp3 = \text{Zer}(1,4)$
 $\text{Mat } temp4 = \text{Zer}(4,1)$
 $\text{Mat } temp = \text{Zer}(1,1)$

$\text{Mat } dtemp1 = \text{Zer}(1,1)$
 $\text{Mat } dtemp2 = \text{Zer}(1,1)$
устанавливаем значения n-матрицы

$n(1,1) = 2$
 $n(1,2) = -2$
 $n(1,3) = 1$
 $n(1,4) = 1$
 $n(2,1) = -3$
 $n(2,2) = 3$
 $n(2,3) = -2$
 $n(2,4) = -1$
 $n(3,3) = 1$

```

n(4,1) = 1
транспонируем n-матрицу
Mat ntrn = Trn(n)
for k = 1 to 3
  icount = 0
  if k = 1 then Mat p = x
  if k = 2 then Mat p = y
  if k = 3 then Mat p = z
  Mat temp1 = p*ntrn
  Mat temp2 = n*temp1
  for u1 = 0 to 1 step 1/(p1 - 1)
    u(1,1) = u1*u1*u1
    u(1,2) = u1*u1
    u(1,3) = u1
    u(1,4) = 1
    du(1,1) = 3*u1*u1
    du(1,2) = 2*u1
    du(1,3) = 1
    du(1,4) = 0
    duu(1,1) = 6*u1
    duu(1,2) = 2
    duu(1,3) = 0
    duu(1,4) = 0
  for w1 = 0 to 1 step 1/(p2 - 1)
    icount = icount + 1
    w(1,1) = s1*s1*s1
    w(2,1) = s1*s1
    w(3,1) = w1
    w(4,1) = 1
    dw(1,1) = 3*w1*w1
    dw(2,1) = 2*w1
    dw(3,1) = 1
    dw(4,1) = 0
    dw(1,1) = 6*w1
    dw(2,1) = 2
    dw(3,1) = 0
    dw(4,1) = 0
    Mat temp3 = u*temp2
    Mat temp4 = temp2*w
    Mat temp5 = du*temp2
    Mat temp6 = temp2*dw
    Mat temp7 = temp3*w
    Mat dtemp1 = du*temp4
    Mat dtemp2 = temp3*dw
    Mat dtemp3 = temp5*dw
    Mat dtemp4 = duu*temp4
    Mat dtemp5 = temp3*dww
    q(icount,k) = temp7(1,1)
    qu(icount,k) = dtemp1(1,1)
    qw(icount,k) = dtemp2(1,1)
    quw(icount,k) = dtemp3(1,1)
    quu(icount,k) = dtemp4(1,1)
  
```

вычисляем $p * ntrn$ -константу для всех u , w

вычисляем $n * p * ntrn$ -константу для всех u , w

устанавливаем значения u -матрицы

устанавливаем значения du -матрицы

устанавливаем значения duu -матрицы

устанавливаем значения w -матрицы

устанавливаем значения dw -матрицы

устанавливаем значения dww -матрицы

вычисляем $u * n * p * ntrn$

вычисляем $n * p * ntrn * w$

вычисляем $du * n * p * ntrn$

вычисляем $n * p * ntrn * dw$

вычисляем $q(u, w)$

вычисляем $qu(u, w)$

вычисляем $qw(u, w)$

вычисляем $quw(u, w)$

вычисляем $quu(u, w)$

вычисляем $qww(u, w)$

запоминаем точку поверхности q

запоминаем производную поверхности qu

запоминаем производную поверхности qw

запоминаем производную поверхности quw

запоминаем производную поверхности quu

```
qww(icount,k) = dtemp5(1,1)
```

запоминаем производную поверхности q_{ww}

```
next w1
```

```
next u1
```

```
next k
```

```
return
```

dbsurf

Подпрограмма вычисляет декартово произведение для поверхности В-сплайна и его производные, используя открытые однородные узловые векторы (уравнения (6-58) и (6-62)–(6-66)).

$b()$	= массив, содержащий точки сетки многоугольника
	$b(,1)$ содержит компоненту x вершины
	$b(,2)$ содержит компоненту y вершины
	$b(,3)$ содержит компоненту z вершины
	Замечание. $B_i,j = b(,)$ имеет размерность $n * t \times 3$, причем j изменяется быстрее. Размер сетки многоугольника $n \times t$
$d1mbasis(,)$	= первая производная базисных функций для w (уравнение (5-74))
$d1nbasis(,)$	= первая производная базисных функций для u (уравнение (5-74))
$d2mbasis(,)$	= вторая производная базисных функций для w (уравнение (5-75))
$d2nbasis(,)$	= вторая производная базисных функций для u (уравнение (5-75))
k	= порядок в направлении u
l	= порядок в направлении w
$mbasis(,)$	= базисные функции для w (уравнение (5-63))
$mpts$	= число вершин многоугольника в направлении w
$nbasis(,)$	= базисные функции u (уравнение (5-84))
$nptw$	= число вершин многоугольника в направлении w
$p1$	= число параметрических линий в направлении u
$p2$	= число параметрических линий в направлении w
$q()$	= массив, содержащий результирующую поверхность
	$q(,1)$ содержит компоненту x вектора
	$q(,2)$ содержит компоненту y вектора
	$q(,3)$ содержит компоненту z вектора
	для фиксированного значения u следующие t элементов содержат
	значения для кривой $q(u(zibi), w)$; q имеет размеры
	$p1 * p2 \times 3$. Отображается поверхность размера $p1 \times p2$
$qu(,)$	= производная векторов u для В-сплайн поверхности;
	имеет размеры $p1 * p2 \times 3$ (уравнение (6-62))
$qw(,)$	= производная векторов w для В-сплайн поверхности;
	имеет размеры $p1 * p2 \times 3$ (уравнение (6-63))
$quu(,)$	= производная векторов uu для В-сплайн поверхности;
	имеет размеры $p1 * p2 \times 3$ (уравнение (6-64))
$quiw(,)$	= производная векторов uiw для В-сплайн поверхности;
	имеет размеры $p1 * p2 \times 3$ (уравнение (6-65))
$quw(,)$	= производная векторов uw для В-сплайн поверхности;
	имеет размеры $p1 * p2 \times 3$ (уравнение (6-66))

```
subroutine dbsurf (b(),k,l,npts,mpts,p1,p2; q(),qu(),qw(),quw(),quu(),qww(),)
```

допускает 20 заданных точек с базисной функцией порядка 5

```
dimension x(30),y(30)
```

```
dimension nbasis(1,20),mbasis(1,20)
```

```
dimension d1nbasis(1,20),d1mbasis(1,20)
```

```
dimension d2nbasis(1,20),d2mbasis(1,20)
```

определяем размерности и заполняем массивы нулями

```
Mat x = Zer(npts + k)
```

```
Mat y = Zer(mpts + 1)
```

```
Mat nbasis = Zer(1,npts)
```

```
Mat mbasis = Zer(1,mpts)
```

```
Mat d1nbasis = Zer(1,npts)
```

```
Mat d1mbasis = Zer(1,mpts)
```

```
Mat d2nbasis = Zer(1,npts)
```

```
Mat d2mbasis = Zer(1,mpts)
```

```
Mat q = Zer(p1*2,3)
```

```
Mat qu = Zer(p1*2,3)
```

```
Mat qw = Zer(p1*2,3)
```

```
Mat quw = Zer(p1*2,3)
```

```
Mat quu = Zer(p1*2,3)
```

```
Mat qww = Zer(p1*2,3)
```

генерируем открытые однородные узловые векторы

```
call knot(npts,k; x())
```

```
call knot(mpts,l; y())
```

```
nplusc = npts + k
```

```
mplusc = mpts + l
```

```
icount = 0
```

вычисляем точки на B-сплайн поверхности

```
for u = 0 to x(nplusc) step x(nplusc)/(p1 - 1)
```

```
call dbasis(k,u,npts,x; nbasis,d1nbasis)
```

вычисляем узловой вектор и

вычисляем узловой вектор w

```
for w = 0 to y(mplusc) step y(mplusc)/(p2 - 1)
```

базовые функции и поверхности

```
call dbasis(l,w,mpts,y; mbasis,d1mbasis,d2mbasis)
```

```
icount = icount + 1
```

```
for i = 1 to npts
```

```
for j = 1 to mpts
```

```
for s = 1 to 3
```

```
j1 = mpts*(i - 1) + j
```

```
q(icount,s) = q(icount,s) + b(j1,s)*nbasis(1,i)*mbasis(1,j)
```

```
qu(icount,s) = qu(icount,s) + b(j1,s)*d1nbasis(1,i)*mbasis(1,j)
```

```
qw(icount,s) = qw(icount,s) + b(j1,s)*nbasis(1,i)*d1mbasis(1,j)
```

```
quw(icount,s) = quw(icount,s) + b(j1,s)*d1nbasis(1,i)*d1mbasis(1,j)
```

```
quu(icount,s) = quu(icount,s) + b(j1,s)*d2nbasis(1,i)*mbasis(1,j)
```

```
qww(icount,s) = qww(icount,s) + b(j1,s)*nbasis(1,i)*d2mbasis(1,j)
```

```
next s
```

```
next j
```

```
next i
```

```
next w
```

```
next u
```

```
return
```

loftlinu

Подпрограмма вычисляет правильную поверхность в направлении *u* (уравнение (6-43)).

$a(,)$ = массив размера $n \times 3$, содержащий точки на кривой $p(u, 0)$
 $b(,)$ = массив размера $n \times 3$, содержащий точки на кривой $p(u, 1)$
 $inum$ = число точек на каждой правильной линии
 n = число точек на граничных кривых
 $q(,)$ = массив размера $n * inum \times 3$, содержащий точки на промежуточных кривых

```

subroutine loftlinu (n,a(,),b(,),inum; q(,))
  icoount = 0
  вычисляем поверхность
  for w = 0 to 1 step 1/(inum - 1)
    for l = 1 to n
      icoount = icoount + 1
      for k = 1 to 3
        q(icoount,k) = a(l,k)*(1 - w) + b(l,k)*w вычисляем поверхность (уравнение (6-43))
      next k
    next l
  next w
  return

```

loftlinw

Подпрограмма вычисляет правильную поверхность в направлении w (уравнение (6-44)).

$a(,)$ = массив размера $n \times 3$, содержащий точки на кривой $p(0, w)$
 $b(,)$ = массив размера $n \times 3$, содержащий точки на кривой $p(1, w)$
 $inum$ = число точек на правильной линии
 n = число точек на граничных кривых
 $p(,)$ = массив размера $n * inum \times 3$, содержащий точки на правильных линиях

```

subroutine loftlinw (n,a(,),b(,),inum; q(,))
  вычисляем поверхность
  for u = 0 to 1 step 1/(inum - 1)
    for l = 1 to n
      icoount = icoount + 1
      for k = 1 to 3
        p(icoount,k) = a(l,k)*(1 - u) + b(l,k)*u вычисляем поверхность (уравнение (6-44))
      next k
    next l
  next u
  return

```

mbezsurf

Подпрограмма вычисляет поверхность Безье, используя матричные методы (уравнение (6-59)).

$b(,)$ = массив, содержащий вершины многоугольника

$b(,1)$ содержит компоненту x вершины

$b(,2)$ содержит компоненту y вершины

$b(,3)$ содержит компоненту z вершины

Замечание: $B_{i,j} = b(,)$ имеет размерность $n * m \times 3$, причем j изменяется быстрее

jin = базисная функция Бернштейна в направлении u (уравнение (5-63))

kjm = базисная функция Бернштейна в направлении w (уравнение (5-63))

$m(,)$ = матрица [M] (уравнение (6-59))

$m1$ = число, на единицу меньшее числа вершин многоугольника в направлении w

$n1$ = число, на единицу меньшее числа вершин многоугольника в направлении u

$Factrl$ = функция для вычисления факториала числа

$n(,)$ = матрица [N] (уравнение (6-59))

$Ni(n,i)$ = функция, вычисляющая $\binom{n}{i}$ (уравнение (5-64))

$p1$ = число параметрических линий в направлении u

$p2$ = число параметрических линий в направлении w

$q(,)$ = координатные векторы для точек на поверхности

$q(,1)$ содержит компоненту x вектора

$q(,2)$ содержит компоненту y вектора

$q(,3)$ содержит компоненту z вектора

для фиксированного значения i следующие m элементов содержат
значения для кривой $q(u_i, w)$; q имеет размерность $p1 * p2 \times 3$

$u(,)$ = [U] параметрическая матрица (уравнение (6-59))

$w(,)$ = [W] параметрическая матрица (уравнение (6-59))

subroutine mbezsurf (b(,),n1,m1,p1,p2; q(,))

допустимая сетка Безье 10×10

dimension x(10,10),y(10,10),z(10,10),g(10,10)

dimension n(10,10),m(10,10),u(1,10),w(10,1)

dimension temp1(10,10),temp2(10,10),temp3(1,10),temp4(1,1)

def Ni(n,i) = Factrl(n)/(Factrl(i)*Factrl(n - i))

определяем размерности матриц и заполняем нулями эти матрицы

Mat n = Zer(n1 + 1,n1 + 1)

Mat m = Zer(m1 + 1,m1 + 1)

Mat x = Zer(n1 + 1,m1 + 1)

Mat y = Zer(n1 + 1,m1 + 1)

Mat z = Zer(n1 + 1,m1 + 1)

Mat g = Zer(n1 + 1,m1 + 1)

Mat u = Zer(1,n1 + 1)

Mat w = Zer(m1 + 1,1)

Mat temp1 = Zer(n1 + 1,m1 + 1)

Mat temp2 = Zer(n1 + 1,m1 + 1)

Mat temp3 = Zer(1,m1 + 1)

Mat temp4 = Zer(1,1)

устанавливаем значения для матриц x , y , z , т. е. компоненты матрицы [B]

for i = 1 to n1 + 1

 for j = 1 to m1 + 1

```

ij = (m1 + 1)*(i - 1) + j
x(i,j) = b(ij,1)
y(i,j) = b(ij,2)
z(i,j) = b(ij,3)
next j
next i
устанавливаем значения для матриц [N] и [M]
for i = 0 to n1
  for j = 0 to n1
    if i + j <= n1 then
      n(i + 1,j + 1) = Ni(n1,j)*Ni(n1 - j,n1 - i - j)*(-1)^(n1 - i - j)
    else
      n(i + 1,j + 1) = 0
    end if
  next j
next i
for i = 0 to m1
  for j = 0 to m1
    if i + j <= m1 then
      m(i + 1,j + 1) = Ni(m1,j)*Ni(m1 - j,m1 - i - j)*(-1)^(m1 - i - j)
    else
      m(i + 1,j + 1) = 0
    end if
  next j
next i
Mat m = Trn(m)      транспонируем матрицу [M] для последующего использования
for k = 1 to 3
  icount = 0           переустанавливаем счетчик точек поверхности
  if k = 1 then Mat g = x      присваиваем значение каждой компоненты
  if k = 2 then Mat g = y
  if k = 3 then Mat g = z
  устанавливаем значения для постоянной матрицы [N][B][M]
  (напомним, что [M]=[M]T)
Mat temp1 = g*m          [B][M]T
Mat temp2 = n*temp1      [N][B][M]T
for t = 0 to 1 step 1/(p1 - 1)      устанавливаем значения матрицы и
  for i = 0 to n1
    u(1,i + 1) = t^(n1 - i)
  next i
  Mat temp3 = u*temp2      [U][N][B][M]T — константа для каждого значения и
  for s = 0 to 1 step 1/(p2 - 1)      устанавливаем значения матрицы w
    icount = icount + 1
    for i = 0 to m1
      w(i + 1,1) = s^(m1 - i)
    next i
    Mat temp4 = temp3*w      вычисляем точку поверхности
    q(icount,k) = temp4(1,1)  запоминаем точку в массиве q
  next s
next t

```

```
next k
return
```

rbspsurf

Подпрограмма вычисляет декартово произведение поверхности рационального В-сплайна, используя открытые однородные узловые вектора (уравнение (6-85)).

b() = массив, содержащий точки сетки многоугольника
b(,1) содержит компоненту *x* вершины

b(,2) содержит компоненту *y* вершины

b(,3) содержит компоненту *z* вершины

b(,4) содержит однородную координату нормирующего множества

Замечание: $B_{i,j} = b(.,)$ имеет размерность $n * m \times 4$, причем *j* изменяется быстрее. Размер сетки многоугольника $n \times m$

k = порядок в направлении *u*

l = порядок в направлении *w*

mbasis = массив, содержащий нерациональные базисные функции для *w* (уравнение (5-84))

mpts = число вершин многоугольника в направлении *w*

nbasis = массив, содержащий нерациональные базисные функции для *w* (уравнение (5-84))

npts = число вершин многоугольника в направлении *u*

p1 = число параметрических линий в направлении *u*

p2 = число параметрических линий в направлении *w*

q() = массив, содержащий результирующую поверхность

q(,1) содержит компоненту *x* вектора

q(,2) содержит компоненту *y* вектора

q(,3) содержит компоненту *z* вектора

для фиксированного значения *i* и следующие *m* элементов содержат

значения для кривой $q(u_i, w); q$ имеет размерность $p1 * p2 \times 3$.

Отображается поверхность размера $p1 \times p2$

sum = сумма рациональных базисных функций поверхности

subroutine rbspsurf (b(),k,l,npts,mpts,p1,p2; q(),)

допускает 20 заданных точек с базисной функцией порядка 5

dimension x(30),y(30),nbasis(1,20),mbasis(1,20)

определяем размерности и заполняем массивы нулями

Mat *x* = Zer(*npts* + *k*)

Mat *y* = Zer(*mpts* + *l*)

Mat *nbasis* = Zer(1,*npts*)

Mat *mbasis* = Zer(1,*mpts*)

Mat *q* = Zer(*p1*p2*,3)

генерируем открытые однородные узловые векторы

call knot(*npts*,*k*; *x()*)

call knot(*mpts*,*l*; *y()*)

nplusc = *npts* + *k*

mplusc = *mpts* + *l*

icount = 0

вычисляем точки на В-сплайн поверхности

for *u* = 0 to *x(nplusc)* step *x(nplusc)/(p1 - 1)*

вычисляем узловой вектор и
вычисляем узловой вектор *w*

```

call basis(k,u,npts,x; nbasis)           нерациональная базисная функция для u
for w = 0 to y(mplusc) step y(mplusc)/(p2 - 1)
    call basis(l,w,mpts,y; mbasis)           нерациональная базисная функция для w
    call sumrbas(b,nbasis,mbasis,npts,mpts; sum)   суммируем базисные функции
    icount = icount + 1
for i = 1 to npts
    for j = 1 to mpts
        for s = 1 to 3
            j1 = mpts*(i - 1) + j
            qtemp = b(j1,4)*b(j1,s)*nbasis(1,i)*mbasis(1,j)/sum      точка поверхности
            q(icount,s) = q(icount,s) + qtemp      сохраняем в массиве поверхности
        next s
    next j
    next i
next w
next u
return

```

sumrbas

Подпрограмма вычисляет сумму нерациональных базисных функций (уравнение (6-87)).

b() = массив, содержащий точки сетки многоугольника
b(,1) содержит компоненту *x* вершины
b(,2) содержит компоненту *y* вершины
b(,3) содержит компоненту *z* вершины
b(,4) содержит однородную координату нормирующего множителя
Замечание: $B_{i,j} = b(.,)$ имеет размерность $n * t \times 4$, причем *j* изменяется быстрее. Размер сетки многоугольника $n \times t$

mbasis = массив, содержащий нерациональные базисные функции для *w* (ур. 5-84)
mpts = число вершин многоугольника в направлении *w*
nbasis = массив, содержащий нерациональные базисные функции для *u* (ур. 5-84)
npts = число вершин многоугольника в направлении *u*
sum = сумма базисных функций

```

subroutine sumrbas (b(,),nbasis(,),mbasis(,),npts,mpts; sum)
sum = 0
for i 1 to npts
    for j 1 to mpts
        j1 = mpts*(i - 1) + j
        sum = sum + b(j1,4)*nbasis(1,i)*mbasis(i,j)
    next j                                         вычисляем сумму
next i
return

```

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

- Адресация по строкам и столбцам (addressing row-column)** 50
- Адресуемость (addressability)** 72
- Активная графика (active graphics)** 72, 73
- Алгоритм, рисования (drawing algorithm)** 17, 18, 76, 491
- соединения (connectivity) 491
- Анаглиф поляризованный (polarized and glyptic)** 199
- цветовой (chromatic) 199
- Андерсон (Anderson D.)** 287
- Аппроксимация кусочно-полиномиальная (piecewise polynomial approximation)** 219
- по наименьшим квадратам (least squares) 219
 - хордовая (chord length) 276
- Арифметические выражения (arithmetic expressions)** 506
- Аффинная геометрия (affine geometry)** 144
- Аффинное преобразование (transformation)** 144, 296, 311
- пространство (space) 381
- База данных (data base)** 18–20
- Базис глобальный (basis global)** 309
- неглобальный (nonglobal) 310
- Базисная функция (basis function)** 297
- В-сплайна (B-spline) 310
- Базисные функции периодические (periodic basis functions)** 318–353
- Барский (Barsky B.)** 343, 458
- Безье (Bézier P.)** 296, 381
- Безье многоугольник (polygon)** 297
- поверхность (surface) 382, 436–445
- Бем (Böhm W.)** 355
- Бернштейна базисная функция (Bernstein basis function)** 297–300
- Брезенхэма алгоритм (Bresenham's algorithm)** 70
- Брюэр (Brewer J.)** 287
- де Бур (de Boor C.)** 310
- Буфер кадра (frame buffer)** 38, 39, 44–48
- Бэрнхилл (Barnhill R.)** 436
- Вектор касательный (tangent vector)** 221, 434
- кручения (twist) 411, 434–436
 - узловой (knot) 310–318
- Векторное произведение (cross production)** 100, 141
- Векторы координатные (position vectors)** 76, 94, 221
- Весприлл (Versprille K.)** 359
- Вид вспомогательный (auxillary view)** 150
- сверху (top) 150
 - сзади (rear) 150
 - слева (left) 150
 - снизу (bottom) 150
 - спереди (front) 150
 - справа (right) 150
- Видеокамеры цветные (color film cameras)** 72
- Видеомонитор (video monitor)** 44–47

- Восприятие глубины (depth perception) 199
- Восстановление объектов (reconstruction of images) 210–216
- трехмерной формы (three-dimensional) 212
- Вращение (rotation) 87–91
- комбинированное (combined) 124, 125
 - относительно начала координат (about the origin) 89
 - осей параллельных (parallel axis) 130–134
 - произвольных (arbitrary) 134–141
 - трехмерное (three-dimensional) 120
- Генератор символов аппаратный (hardware character generation) 21, 22
- программный (software) 21, 22
- Геометрические граничные условия (geometric boundary conditions) 220
- Геометрическое моделирование (geometric modeling) 482
- Геометрия аффинная (affine geometry) 144
- перспективная (начертательная) (perspective) 144
- Гипербола параметрическая (parametric hyperbola) 236–239
- Гиперболоид (hyperboloid) 387
- двуполостный (of two sheets) 405, 408
 - однополостный (of one sheet) 382, 405, 408
- Гиперболы сегмент (hyperbolic segment) 251
- Гордон (Gordon W.) 296, 310
- Горизонтальный перевод луча (horizontal retrace) 47, 48
- Графические примитивы (graphics primitives) 483, 484
- , ввод (input) 483
- , составные операции (compound operations) 484
 - , управление устройством (device state control) 483
 - устройства активные (active devices) 72
 - пассивные (passive) 72
- Графический метафайл компьютера (computer graphic metafile) 481, 489
- пакет растрового дисплея (raster display graphic package) 493–499
- Графических данных формат передачи (initial graphics exchange standard) 14, 73
- Графическое ядро системы (graphical kernel system) 481, 482
- Графопостроители (plotters) 57, 62, 67–71
- Грегори (Gregory J.) 436
- Гринберг (Greenberg D.) 458
- Дил (Dill J.) 422
- Диоды, регенерирующие свет (light-emitting diodes) 49
- Диск плоский (planar disc) 382
- Дисплей (display) 31–37, 49–56
- плазменный (plasma) 49–51
 - растровый (raster) 38–43
 - цветной (color) 56
- Доводка (lofting) 260
- Документация программ (program documentation) 524
- Евклида* геометрия (Euclidean geometric space) 110
- Запоминающая трубка (storage tube) 31, 32

- Иерархическая интерактивная графическая система программиста (programmer's hierarchical interactive graphical standard)** 58, 481, 482, 484
- Изображение (picture)** 17, 18, 484
- перевернутое (inverted image) 184, 197
- , сегментация (segmentation picture) 35, 36
- Изопараметрические линии (isoparametric lines)** 414, 419
- Интерактивная машинная графика (interactive computer graphics)** 14
- Интерактивные устройства (interactive devices)** 22–31
-- , алфавитно-цифровая клавиатура (alphanumeric keyboard) 22
-- , валюатор (valuator) 25, 26
-- , логическая кнопка (logical button) 29
-- , локатор (locator) 22
-- , мышь (mouse) 27
-- , планшет (tablet) 22, 29
-- , рычаг (joystick) 22, 25, 26
-- , световое перо (light pen) 22, 31
-- , сенсорная панель (touch panel) 25
-- , шар (track ball) 25, 26
- Интерполяция (interpolation)** 219, 283–304, 458–461
- Интерфейс виртуального устройства (virtual device interface)** 58, 481, 482
- прикладной программы (application programmer's) 481
- Истинная форма (true shape)** 154, 163
- Истинный размер (size)** 163
- Клейн (Cline A.)** 283
- Кокс (Cox W.)** 311
- Кокса – де Бура формула (Cox – de Boor formula)** 310, 316
- Комментарий (comment statements)** 505
- Константы (constants)** 505
- Контуры корабля (ship's lines)** 355
- Конус мнимый (imaginary cone)** 406
- прямой круговой усеченный (truncated right circular) 382, 383
- эллиптический (elliptic)** 408
- Координаты системы (coordinate systems)** 486–488
- Координаты абсолютные (absolute coordinates)** 18, 22
- глобальные (global) 133
- локальные (local) 133
- мировые (world) 20
- нормализованные (normalized) 22
- однородные (homogeneous) 19, 102, 109, 115, 144
- относительные (relative) 18
- пользователя (user) 486
- преобразованные (transformed) 77
- устройства (device) 486
- Коэн (Cohen E.)** 301, 338, 339, 355
- Коэффициент искажения (foreshortening factor)** 154
- Кривая аналитическая (analytical curve)** 218
- Безье (Bézier) 294–310
-- , матричная форма (matrix form) 301
-- , вычерчивание (fitting) 219, 256, 261
-- замкнутая (closed) 221
-- непараметрическая (nonparametric) 219–221
-- осезависимая (axis-dependent) 221
-- осенезависимая (axix-independent) 222
-- параметрическая (parametric) 221–225
-- плоская (two-dimensional) 218
- Кривизна (curvature)** 302
- гауссова (Gaussian)** 463, 466

- главная (principal) 423, 463
- , радиус (radius of) 260
- средняя (average) 465
- эйлерова (Eulerian) 463
- Кривые граничные (boundary curves) 413, 416
- непараметрические (nonparametric) 219
- параметрические (parametric) 221
- плоские (plane) 218–245
- пространственные (space) 256–359
- Куб деформированный (distorted cube) 183
- Кунс (Coons S.) 359
- Кунса поверхность (surface) 425–430
- Курсор (cursor) 18, 23

- Лестничный эффект (aliasing) 38, 58, 70
- Линза увеличителя (enlarger lens) 198
- Линия проекции (line of projection) 154
- Логические операции (logical operators) 506
- Локатор (locator) 22, 29, 30

- Массивы (array variables) 506
- Масштабирование (scaling) 79, 93–95, 98, 116–120
 - локальное (local) 116–120
 - неоднородное (nonuniform) 94
 - области (area) 99
 - однородное (uniform) 94, 95
 - пропорциональное (overall) 108
 - трехмерное (three-dimensional) 116
- Матрица, вычитание (matrix subtraction) 502
 - геометрическая (geometry) 418
 - диагональная (diagonal) 267, 501
 - единичная (identity) 501
 - квадратная (square) 501
 - конкатенированная (concatenated) 9
- нулевая (null) 502
- обратная (inversion) 84
- , обращение (inverse) 504
- , определитель (determinant) 503
- ортогональная (orthogonal) 90, 101
- отражения (reflection) 113
- переноса (translation) 141
- поворота (rotation) 122
- симметричная (symmetrical) 502
- , сложение (addition) 502
- транспонированная (transpose) 90, 504
- тридиагональная (tridiagonal) 267
- , умножение (multiplication) 503
- Мезофаза нематическая (mesophase nematic) 54
- смектическая (smetic) 54
- холестерическая (cholesteric) 54
- Меню командное (command menu) 524, 525
- помощи (help) 524, 525
- Меридиан долготы (meridian of longitude) 411
- Мерцание изображения (flicker) 46

- Направление взгляда (sight vector) 207
- Негатив фотографический (photographic negative) 196, 197
- Нильсон (Nielson G.) 283

- Обратный ход вертикальный (vertical retrace) 47
- горизонтальный (horizontal) 47
- Оверхаузер (Overhauser A.) 284, 287, 296
- Окно (window) 20, 486
 - отсечения (clipping) 20
 - указания (hit) 31
- Окружность единичная (unit circle) 224, 230
- касательная (tangent) 244

- непараметрическая (nonparametric) 223
 - параметрическая (parametric) 223, 226
 - проходящая через три точки (through three points) 241, 242
- Операторы** (statements) 506–508
- Осло алгоритм** (Oslo algorithm) 355
- Отражение** (reflection) 79, 90–92, 101, 104, 116, 126, 127
- от произвольной плоскости (through arbitrary plane) 141–145
 - - - прямой (line) 103
 - трехмерное (three-dimensional) 126
- Отсечение** (clipping) 20, 22
- Оцифровыватель** (digitizer) 29
-
- Парабола** (parabola) 220, 233–235, 245, 247
- Параболоид гиперболический** (paraboloid hyperbolic) 40, 405
- эллиптический (elliptic) 408
- Параллакс движения** (movement parallax) 198
- , панорамограммы (panoramograms) 199
 - , панорамные стереограммы (panoramic stereograms) 199
 - , стереограммы (stereograms) 199
- Параллель широты** (parallel of latitude) 411
- Переменные** (variables) 505
- Перенос пространственный** (three-dimensional translation) 128
- Пересечение параллельных прямых** (intersection of parallel lines) 111
- плоскости и сферы (plane and sphere) 411
- Петерс** (Peters G.) 381
- Пигл** (Piegl L.) 359

- Пирамида видимости** (frustum of vision) 8
- Планшет** (tablet) 22, 29
- Пиксель** (pixel) 37
- Плоскость битовая** (bit plane) 38
- проекции (of projection) 146
 - физическая (physical) 107
- Площадь вписанная наибольшая** (maximum inscribed area) 231, 234, 239
- Поверхности Безье** (Bezier surface) 436–446
- вращения (of revolution) 382–396
 - кусочные (piecewise) 411
 - , представление параметрическое (parametric representation) 384
 - скульптурные (sculptured) 428
- Поверхность билинейная** (surface bilinear) 417–421
- бипараметрическая (biparametric) 32, 384
 - заметающая (sweep) 396–402
 - интерполяционная (interpolation) 417
 - квадратичная (quadric) 402–405
 - линейчатая (ruled) 419
 - развертывающаяся (developed) 420–425
 - седловая (saddle) 407
 - цилиндрическая (cylindrical) 382
- Поворот** (rotation) 87, 103, 130, 134
- Подсказки бинокулярные** (binocular cues) 198
- монокулярные (monocular) 198
- Подпрограммные модули** (subroutine modules) 508
- Поле зрения** (viewport) 20, 21
- Поляризатор** (polarizer) 55
- Поляризация света** (polarization of light) 54
- Порог переключения** (switching threshold) 50

- Правило правой руки (right-hand sense) 121, 122
- Представление изображений (representing pictures) 16
- кривых (of curves) 226–236
 - Преобразование аффинное (transformation affine) 122, 296, 311, 361
 - билинейное (bilinear) 116
 - единичного квадрата (of unit square) 97–99
 - жестких конструкций (solid body) 100–101
 - заметающее (sweep) 397
 - инвариантное (invariant) 362
 - комбинированное (combined) 95–98
 - композиции (multiple) 129–131
 - конкатенации (concatenated) 96
 - линейное (linear) 116
 - линий параллельных (parallel lines) 82
 - - пересекающихся (intersecting) 83, 86
 - масштабирования (scaling) 93–95, 116–120
 - отражения (reflection) 101
 - перспективное (perspective) 116, 170–181
 - поверхности (surface) 412–415
 - прямых линий (of straight lines) 79, 80
 - стереотипное (stereo) 198–202
 - точек (points) 77, 78, 181, 183, 189
- Принтер лазерный (laser printer) 60
- струйный (ink jet) 64–66
 - цветной (color) 65
- Программное обеспечение машинной графики (computer graphic software) 74
- Программы спецификации (programming projects) 523–536
- Проекторы (projectors) 145
- Проекция аксонометрическая (projection axonometric) 154–162
- диметрическая (dimetric) 159
 - изометрическая (isometric) 154
- косоугольная (oblique) 163–169
 - - кабине (cabinet) 163–168
 - - кавалье (ca cavalier) 163–168
 - ортографическая (orthographic) 146, 147
 - параллельная (parallel) 145
 - перспективная (perspective) 145
 - плоская (plane) 145, 147
 - стереографическая (stereographic) 198–206
 - триметрическая (trimetric) 154–157
- Проектирование (projection) 106
- Производная кривой Безье (derivative Bézier curve) 304
- параметрической кривой (parametric curve) 221
 - поверхности Безье (Bézier surface) 442
 - В-сплайн базисной функции (B-spline basic function) 341
- Пространство объектное (object space) 415, 416
- физическое (physical) 360
- Прямой доступ в память (direct memory access) 37
- Псевдокод (pseudocode) 505
-
- Разрешение (resolution) 72, 488
- Расстояние между зрачками (inter-pupil distance) 203
- Растровая развертка (scan conversion) 44, 46
- Расширение (expansion) 94, 108, 119
- Реальное время (real time) 15–20, 40–42
- Ризенфельд (Riesenfeld P.) 301, 310
- Роджерс (Rogers D.) 20, 243, 382, 458, 460
-
- Сабин (Sabin M.) 381
- Сазерленд (Sutherland I.) 213
- Свет (highlight) 198
- Сдвиг (shearing) 79, 80, 98, 120

- трехмерный (three-dimensional) 120
- Сегмент параболический (parabolic segment) 236
- Секущая плоскость (cutting plane) 153
- Сечение (sectional view) 153
- Сжатие (compression) 94, 108, 119
- Символов генератор (character generation) 18, 22
- Символы (characters) 21
- Система глаз — мозг (eye — brain system) 198, 199
- Скаляр (scalar) 502
- Скалярное произведение (scalar product) 100
- Сканирующая строка (scan line) 47
- Складка (knuckle) 451
- Скользящая сетка (sliding net) 455
- Смит (Smith L.) 234, 236
- Спецификации программ (programming projects) 523–536
- Спираль (helix) 259, 396
- Сплайн математический (mathematical spline) 260–262
 - напряженный (under tension) 283
 - физический (physical) 260–262
- В-сплайн (B-spline) 309–343
 - интерполяция (interpolation) 458
 - поверхности (surfaces) 446–457
 - рациональный (rational) 360
- ν -сплайн (ν -spline) 283
- Сплайны кубические (cubic splines) 260–287
- Стереопара (stereopair) 202–205
- Стереоскоп (stereoscope) 194, 202–204
 - зеркальный (mirrored) 204
 - фокальный плоский (focal plane) 203
- Теневая маска (shadow mask) 45, 46
- Тени (shadows) 198
- Тиллер (Tiller W.) 360
- Точек следы (trace points) 191–195
 - совокупность (collection of) 218
- Точка, алгоритм (point, algorithm) 17, 18
- поверхности вращения (on surface of revolution) 384
- основные операции (fundamental operation) 18
- плечевая (shoulder) 372
- представления кривой (representation of curve) 220
- схода (vanishing) 173, 191–195
- Точки бесконечности (points at infinity) 109–111
- концевые (end points) 221
- Углы сближения (convergence angles) 198
- Угол наклона (slope) 83
- Уравнение второго порядка (second-degree equation) 245
 - окружности (circle) 385
 - прямой (of a straight line) 219
 - сферы (sphere) 385
 - тора (torus) 386
- Уравнения конических сечений (conic equations) 245–254
- Устройство указания (pick device) 28
- Фасад боковой (side elevation) 151
- главный (front) 151
- Фергюсона поверхность (Ferguson putch) 436
- Фокус (focal point) 196, 197
- Фокусное расстояние (focal length) 197
- Функции гиперболические (hyperbolic functions) 238
 - графические (graphics) 484
 - матричные (matrix) 509
 - полиномиальные (polynomial) 219, 290, 306
 - специальные (special) 510
 - степенные (power) 219

- тригонометрические (trigonometric) 509
- экспоненциальные (exponential) 219
- Функция бипараметрическая (biparametric function) 384
- селектора (pick) 23
- Фог (Fog N. G.) 460
- Формат файла (file format) 511
- Форрест (Forrest A.) 296

- Цвет (color)** 39
- Цветной буфер кадра (color frame buffer) 41
- Цветов таблица (lookup table) 40–43
- Центральные квадрики (central quadrics) 402
 - конические сечения (conic) 239, 245, 248
 - - - вырожденные (degenerative forms) 245
- Центр проекции (center of projection) 145, 179, 181
- Цилиндр круговой (circular cylinder) 382, 383
 - мнимый (imaginary) 406
 - параболический (parabolic) 408

- эллиптический (elliptic) 408
- Цифро-аналоговый преобразователь (digital-analog converter) 39

- Чаррот (Charrot P.) 436
- Чересстрочная развертка (interlace technique) 47

- Швикерт (Schwickert D.)** 283
- Шенберг (Schoenberg I.) 310
- Ширина полосы (bandwidth) 36

- Эйлера уравнение (Euler's equation)** 261
- Электронно-лучевая трубка (cathode ray tube) 44–47
- Эллипс (ellipse) 220, 246, 248, 249
 - параметрический (parametric) 228–231
 - предельный случай (limiting case) 249
 - центрированный в начале (origin-centered) 232
- Эллипсоид (ellipsoid) 386, 403, 407
 - мнимый (imaginary) 406
- Эрмита функция (Hermite function)** 274

ОГЛАВЛЕНИЕ

От редакторов перевода	5
Предисловие к русскому изданию	6
Введение к первому изданию	7
Предисловие	8
Предисловие к первому изданию	11
Глава 1. Введение в машинную графику	14
1-1 Обзор машинной графики	16
1-2 Представление изображений	16
1-3 Подготовка изображений для вывода	18
1-4 Представление предварительно подготовленного изображения	19
1-5 Организация взаимодействия с изображением	22
1-6 Типы графических устройств	31
1-7 Графические дисплеи на запоминающей трубке	31
1-8 Графические дисплеи с регенерацией изображения	33
1-9 Растровые графические дисплеи с регенерацией изображения	37
1-10 Устройство электронно-лучевой трубы	44
1-11 Устройство цветной растровой ЭЛТ	45
1-12 Системы с телевизионным растром	46
1-13 Дисплеи с плоским экраном	48
1-14 Электростатические графопостроители	56
1-15 Лазерные печатающие устройства	60
1-16 Матричные графопостроители	63
1-17 Устройство струйной печати	64
1-18 Устройство термопечати	66
1-19 Перьевые графопостроители	67
1-20 Цветные видеокамеры	71
1-21 Активные и пассивные графические устройства	72
1-22 Программное обеспечение машинной графики	73
1-23 Литература	74
Глава 2. Двумерные преобразования	76
2-1 Введение	76
2-2 Изображение точек	76
2-3 Преобразования и матрицы	76
2-4 Преобразование точек	77
2-5 Преобразование прямых линий	79

2-6	Преобразование средней точки	81
2-7	Преобразование параллельных линий	82
2-8	Преобразование пересекающихся прямых	83
2-9	Поворот	87
2-10	Отражение	90
2-11	Масштабирование	93
2-12	Комбинированные преобразования	95
2-13	Преобразование единичного квадрата	97
2-14	Преобразования жестких конструкций	100
2-15	Перемещения и однородные координаты	101
2-16	Поворот вокруг произвольной точки	102
2-17	Отражение относительно произвольной прямой	103
2-18	Проектирование — геометрическая интерпретация однородных координат	106
2-19	Пропорциональное масштабирование	108
2-20	Точки бесконечности	109
2-21	Правила выполнения преобразований	112
2-22	Литература	114
Глава 3.	Пространственные преобразования и проекции	115
3-1	Введение	115
3-2	Трехмерное масштабирование	116
3-3	Трехмерные сдвиги	120
3-4	Трехмерное вращение	120
3-5	Трехмерное отражение	126
3-6	Пространственный перенос	128
3-7	Композиции преобразований	129
3-8	Повороты вокруг оси, параллельной координатной оси	131
3-9	Поворот вокруг произвольной оси в пространстве	134
3-10	Отражение относительно произвольной плоскости	141
3-11	Аффинная и перспективная (начертательная) геометрия	144
3-12	Ортографические проекции	147
3-13	Аксонометрические проекции	154
3-14	Косоугольные проекции	163
3-15	Перспективные преобразования	170
3-16	Методы создания перспективных видов	182
3-17	Точки схода	191
3-18	Фотография и перспективное преобразование	196
3-19	Стереографическая проекция	198
3-20	Сравнение двух методов построения проекций — с фиксированным объектом и с фиксированным центром проекции	207
3-21	Восстановление трехмерных объектов по проекциям	210
3-22	Литература	217
Глава 4.	Плоские кривые	218
4-1	Введение	218
4-2	Представление кривых	218

4-3	Непараметрические кривые	219
4-4	Параметрические кривые	221
4-5	Параметрическое представление окружности	226
4-6	Параметрическое представление эллипса	229
4-7	Параметрическое представление параболы	232
4-8	Параметрическое представление гиперболы	236
4-9	Процедура использования конических сечений	240
4-10	Общие уравнения конических сечений	244
4-11	Литература	255
Глава 5. Пространственные кривые		256
5-1	Введение	256
5-2	Представление пространственных кривых	257
5-3	Кубические сплайны	260
5-4	Нормализованные кубические сплайны	273
5-5	Другие граничные условия	278
5-6	Параболическая интерполяция	286
5-7	Обобщенная параболическая интерполяция	292
5-8	Кривые Безье	296
5-9	В-сплайны	310
5-10	Конечные условия для периодических В-сплайнов	343
5-11	Подгонка В-сплайнов	350
5-12	Разбиение В-сплайнов	355
5-13	Рациональные В-сплайны	360
5-14	Литература	378
Глава 6. Поверхности		381
6-1	Введение	381
6-2	Поверхности вращения	382
6-3	Заметающие поверхности	396
6-4	Квадратичные поверхности	402
6-5	Кусочное представление поверхностей	410
6-6	Отображение параметрических поверхностей	413
6-7	Билинейная поверхность	417
6-8	Линейчатые и развертывающиеся поверхности	419
6-9	Линейная поверхность Кунса	425
6-10	Бикубическая поверхность Кунса	428
6-11	Поверхности Безье	436
6-12	В-сплайн поверхности	446
6-13	В-сплайн интерполяция	458
6-14	Разбиение В-сплайн поверхностей	461
6-15	Гауссова кривизна и качество поверхности	463
6-16	Рациональные В-сплайн поверхности	467
6-17	Литература	479
Приложение А. Графическое программное обеспечение		481
A-1	Введение	481

A-2 Графические примитивы машинной графики	483
A-3 Элементарные графические функции	484
A-4 Системы координат	486
A-5 Простой графический метафайл	489
A-6 Простые форматы графического ввода	489
A-7 Простые алгоритмы соединения	491
A-8 Простой графический пакет	493
A-9 Литература	500
Приложение В. Матричные операции	501
B-1 Терминология	501
B-2 Сложение и вычитание	502
B-3 Умножение	502
B-4 Определитель квадратной матрицы	503
B-5 Транспонирование матрицы	504
B-6 Обращение квадратной матрицы	504
Приложение С. Псевдокод	505
C-1 Комментарии	505
C-2 Константы	505
C-3 Переменные	505
C-4 Массивы	506
C-5 Оператор присваивания	506
C-6 Арифметические выражения	506
C-7 Логические операции и операции отношения	506
C-8 Оператор finish	506
C-9 Операторы while и end while	506
C-10 Оператор if-then	507
C-11 Операторы if-then-else и end if	507
C-12 Оператор for-next	508
C-13 Оператор go to	508
C-14 Подпрограммные модули	508
C-15 Тригонометрические функции	509
C-16 Матричные функции	509
C-17 Специальные функции	510
Приложение D. Формат файла В-сплайн поверхности	511
Приложение Е. Задачи	513
Глава 2	513
Глава 3	514
Глава 4	515
Глава 5	517
Глава 6	520
Приложение F. Спецификации программ	523
Глава 2	525
Глава 3	530
Глава 4	533

Глава 5.....	534
Глава 6.....	535
Приложение G. Алгоритмы	537
Глава 2.....	538
Глава 3.....	539
Глава 4.....	546
Глава 5.....	551
Глава 6.....	576
Алфавитный указатель	592

Учебное издание

Дэвид Ф. Роджерс, Дж. Алан Адамс
Математические основы машинной графики

Зав. редакцией академик В. И. Арнольд

Зам. зав. редакцией А. С. Попов

Ведущий редактор М. В. Хатунцева

Художник Ю. Л. Максимов

Художественный редактор Н. В. Зотова

Технический редактор О. Г. Лапко

Корректор Е. Н. Клитина

Оригинал-макет подготовлен В. Н. Цлаф
в пакете *LATEX 2\epsilon* с использованием
кириллических шрифтов семейства LH

Лицензия ЛР № 010174 от 20.05.97 г.

Подписано к печати 14.12.2000 г. Формат 70 × 100/16.

Гарнитура Computer Modern. Печать офсетная.

Объем 19,00 бум. л. Усл.-печ. л. 49,40. Уч.-изд. л. 41,57.

Изд. № 1/7803. Тираж 3 000 экз. Заказ 1009

Издательство «Мир»
Министерства РФ по делам печати,
телерадиовещания и средств массовых коммуникаций
107996, ГСП-6, Москва, 1-й Рижский пер., 2.

Диапозитивы изготовлены
в издательстве «Мир»

Отпечатано в соответствии с качеством предоставленных диапозитивов
в ППП «Типография «Наука» 121099, Москва, Шубинский пер., 6

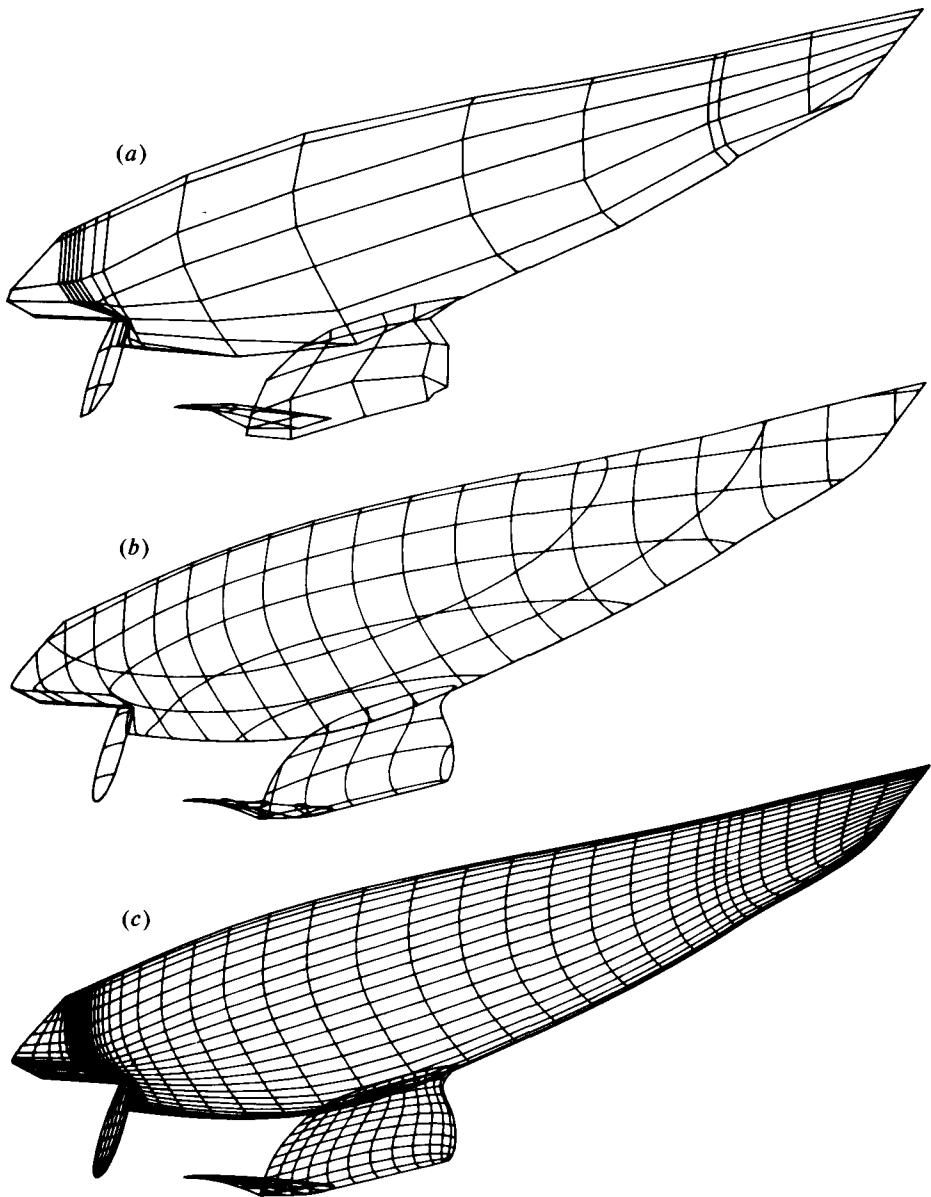


Рис. 1. Американская яхта класса Stars & Strips в форме В-сплайн поверхности, определяемой (а) сеткой многоугольников; (б) телом корпуса, хвостовой частью и ватерлинией; (с) параметрической сеткой. (С разрешения Джорджа Хейзена.)

Математические основы машинной графики