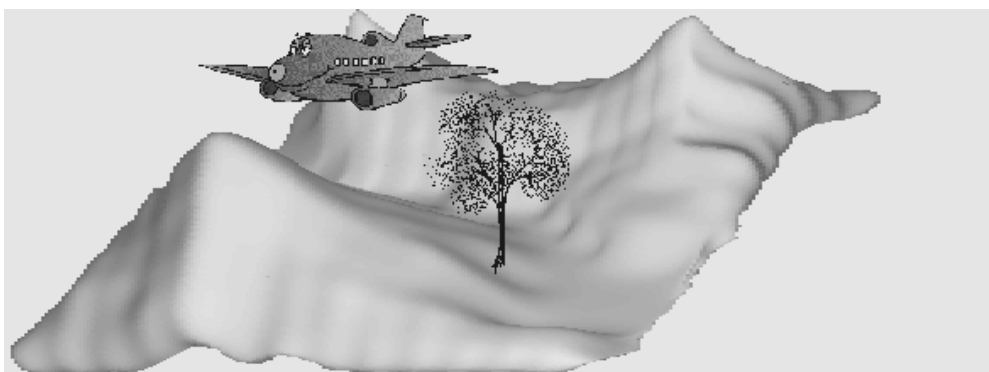

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Ю.Н. Косников



***Поверхностные модели
в системах трехмерной
компьютерной графики***

Учебное пособие

П е н з а 2007

УДК 681.3

Косников Ю.Н.

Поверхностные модели в системах трехмерной компьютерной графики. Учебное пособие. – Пенза: Пензенский государственный университет, 2007. – 60 с.

Излагаются методы построения математических моделей пространственных геометрических объектов. Описаны особенности применения плоских полигонов, поверхностей второго порядка и бикубических сплайнов в качестве геометрических примитивов графических систем. Показана связь геометрического моделирования объектов с их отображением.

Пособие рассчитано на студентов бакалавриата и магистратуры технических специальностей.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 ОБЪЕКТЫ И МОДЕЛИ В КОМПЬЮТЕРНОЙ ГРАФИКЕ	6
1.1 Терминология в области графических систем	6
1.2 Классификация математических моделей трехмерных объектов	8
2 ПРОЦЕСС ОТОБРАЖЕНИЯ ТРЕХМЕРНЫХ ОБЪЕКТОВ	12
2.1 Этапы процесса отображения	12
2.2 Пространственное отсечение примитивов	13
2.3 Преобразования пространственной динамики	15
2.4 Определение видимости примитивов	18
2.5 Растровые преобразования	20
2.6 Графический конвейер	25
3 ГЕОМЕТРИЧЕСКИЕ ПРИМИТИВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ	29
3.1 Полигоны	29
3.2 Поверхности второго порядка	32
3.3 Бикубические сплайны	38
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	53
ПРИЛОЖЕНИЯ	54
Приложение А. Моделирование сплайна Безье по его параметрическому описанию	54
Приложение Б. Моделирование <i>B</i> -сплайна по его параметрическому описанию	56
Приложение В. Формирование отсека <i>B</i> -сплайновой поверхности на основе опорных точек различной кратности ...	58

ВВЕДЕНИЕ

«Чайники» считают, что компьютерная графика – это программы, рисующие красивые картинки: Photoshop, Corel Draw, компьютерные игры. Это не совсем так. Заглянув в толковый словарь по информатике (авторы Першиков В.И., Савинков В.М.), мы узнаем, что компьютерная графика – это «система методов, алгоритмов и программ для ввода, обработки и отображения информации на графических устройствах ЭВМ». Таким образом, компьютерную графику образует целый набор различных средств – математических, алгоритмических, программных, технических. Другими словами, это комплексная дисциплина, имеющая солидную научную, в том числе математическую, базу. Дело в том, что изображение, прежде чем появиться на экране монитора, проходит целый ряд преобразований, и на каждом их этапе применяются свои методы, алгоритмы, приемы. Ряд этапов работает с геометрическими характеристиками объектов. Эти характеристики учитываются не только при описании формы и движения объекта, но и в процессе моделирования его освещенности. Для работы с геометрией объектов служит аппарат геометрического моделирования.

Геометрическое моделирование – это математическое моделирование геометрических объектов, то есть объектов, заданных в пространстве своими формой, составом и размерами. При отображении геометрических объектов, кроме формы, нужно учитывать их пространственное поведение: перемещение и повороты относительно трех координатных осей (шесть степеней свободы), а также изменение геометрических характеристик во времени – метаморфозы. Перемещения и повороты нужны не только для отображения динамики, с их помощью осуществляется «сборка» сложных объектов из простых составных частей – примитивов. Для получения образов пространственных объектов на плоскости экрана (**3D**→**2D**) применяют еще одно геометрическое преобразование – проецирование.

Графические объекты характеризуются не только геометрическими, но и визуальными свойствами – цветом, текстурой (узором), освещенностью. В настоящем пособии эти свойства подробно не рассматриваются. Внимание уделяется математическому описанию графических объектов, которые представлены в виде поверхностных моделей на основе плоских полигонов, поверхностей второго порядка и бикубических сплайнов. Показаны геометрические преобразования, необходимые для отображения составных поверхностей и моделирования их динамики. Материал пособия выстроен в порядке нарастания сложности и предназначен как для студентов бакалавриата, так и для магистрантов. Более глубокое изложение математических аспектов моделирования пространственных объектов можно найти в литературе [1,...,5].

Автор будет благодарен за замечания и предложения, которые можно направлять по электронной почте на адрес: jkos@diamond.stup.ac.ru

1 ОБЪЕКТЫ И МОДЕЛИ В КОМПЬЮТЕРНОЙ ГРАФИКЕ

1.1 Терминология в области графических систем

Прежде, чем приступить к синтезу изображения объекта, нужно занести в графическую систему информацию о его структуре (топологии), геометрии, текстуре, визуальных свойствах (атрибутах) и отношениях между объектом и его окружением (расположении в пространстве). Эти данные образуют информационную модель объекта. Строго говоря, процесс построения модели проходит ряд этапов, а сама модель имеет иерархическую структуру и на разных уровнях иерархии (на разных этапах процесса отображения) представляется конструкциями различных моделирующих языков.

Прежде всего, отображаемые сущности нужно формализовать. Путем абстрагирования от их внутренней структуры и принципа действия переходят от самих сущностей к представлению о их внешнем виде и поведении. Такое представление можно назвать визуально-поведенческой информационной моделью. Она формируется в мозгу специалиста-разработчика. Далее с помощью операций аппроксимации и масштабирования отображаемые объекты избавляют от несущественных для конкретной задачи деталей и приводят к размерам объема индикации системы. Возникает информационная модель, которую можно назвать объемным изображением. Она может иметь форму рисунка, чертежа. Закономерности построения объемного изображения описываются на языке математики, в результате возникает математическая модель объектов. Она состоит из нескольких постоянных составляющих: это описания структуры объектов, их примитивов, атрибутов, текстур. Их содержимое вводится в систему средствами входного языка, образуя графическую базу данных. В процессе отображения форма объектов и их внешний вид, как правило, не изменяются, а значит, не изменяются и соответствующие составляющие математической модели. Зато отношения между объектом и его окружением могут претерпевать

значительные изменения: местоположение, ориентация, интенсивность освещения и некоторые другие параметры объекта в сцене способны принимать различные значения. Данные о подобных параметрах также относятся к математической модели объекта, образуя еще одну ее составляющую – сценарную.

Перечисленные составляющие математической модели относятся к отображаемым сущностям разного уровня иерархии: сцене, объекту, примитивам. Область пространства, в которой определены функции применяемого в графической системе языка моделирования, может быть названа **модельным миром**. Будем исходить из того, что в графических системах отображаются не сущности реального мира – физические и технические объекты, а сущности модельного мира, то есть модели реальных сущностей. Другими словами, мир – это совокупность сущностей (объектов), которые представлены своими моделями и изображения которых могут появиться в поле вывода графической системы параллельно или последовательно во времени.

Сцена – часть модельного мира. Это совокупность сущностей (объектов), представленных своими моделями и отвечающих решаемой в данный момент задаче отображения. Можно назвать сценой набор отображаемых сущностей, описания которых одновременно извлекаются из графической базы данных.

Сцена состоит из **графических объектов**. Объектом будем называть совокупность точек модельного пространства, объединенных функциональной общностью с точки зрения решаемой целевой задачи. Можно подойти к определению объекта со стороны примитивов: объект – это совокупность примитивов, характеризующаяся единством визуальных свойств и идентифицируемая одним именем.

Понятие примитива имеет два основных аспекта рассмотрения. Во-первых, примитив служит "строительным кирпичом" при описании сложных объектов, а во-вторых, – элементарным изображением, формируемым аппаратурой или процедурой графической системы. Поэтому будем различать геометрический и графический примитивы. **Геометрический примитив** – это используемая для построения объектов простая геометрическая фигура (точка, вектор, поверхность

или тело), имеющая формализованное описание. **Графический примитив** – это элементарное изображение, для формирования которого в графической системе имеется специальный аппаратный блок. В различных системах в качестве графического примитива выступают точка, вектор, трапеция и изображения других форм.

1.2 Классификация математических моделей трехмерных объектов

Специфика моделей трехмерного моделирования обусловлена необходимостью передавать ощущение глубины пространства и пространственной формы объектов. Существенное влияние на построение модели оказывает требуемый уровень подобия синтезированного изображения реальной картине. Различают три таких уровня [2]: физическое, физиологическое и психологическое подобие.

При построении модели на уровне физического подобия геометрические, характеристики синтезированного изображения должны полностью соответствовать аналогичным характеристикам реальной картины. Например, для описания объекта в виде чайника нужно использовать криволинейные поверхности сложной формы, какие имеют место в реальности. При физиологическом подобии соответствие модели и реальной картины устанавливается на уровне зрительных ощущений. Модель лишь приближенно передает характеристики реальной картины, однако, благодаря ограниченным возможностям зрительного аппарата, наблюдатель не ощущает возникающих различий. Например, модель чайника состоит из набора плоских участков небольшого размера, не различаемых глазом с расстояния наблюдения. Наконец, психологическое подобие предполагает, что модель, существенно отличаясь по своим характеристикам от реальной картины, создает у наблюдателя адекватное ей зрительное ощущение ("обман глаза"). Так, чайник может быть представлен плоской хорошо "раскрашенной" моделью, не позволяющей показать объект с противоположной стороны. В интерактивной компьютерной графике, как

правило, используется физиологическое подобие синтезированного объекта реальной картине (второй уровень подобия). Оно позволяет строить модели, верно отражающие закономерности трехмерного мира, и в то же время реализовать их с разумными затратами на основе допустимого упрощения этих закономерностей.

Чтобы проанализировать способы, применяемые для моделирования пространственных объектов, проведем краткий обзор математических моделей компьютерной графики. Обзор проведем на основе ограниченного числа логически непротиворечивых признаков представления объектов. Классификация моделей показана на рисунке 1.1.

Во-первых, модели объектов следует различать по их топологическим свойствам, от которых зависит полнота описания конфигурации объектов. По этому признаку можно выделить твердотельные, поверхностные, каркасные и точечные модели. Твердотельные модели представляют объекты как сплошные тела, то есть в виде сочетания всех точек объема, занимаемого объектом в пространстве. Поверхностные модели несут информацию обо всех точках пространства, принадлежащих поверхности объекта, а внутренние точки в них не учитываются. Каркасные модели тоже дают представление только о поверхности объекта, но описывают поверхность как сочетание принадлежащих ей дискретных элементов каркаса – точек или линий. Информация о точках поверхности, лежащих между элементами каркаса, отсутствует. Для описания точечных объектов введены модели, специально предназначенные для этого – точечные модели. Они несут геометрическую информацию лишь о местоположении объектов.

На структуру и сложность моделей решающим образом влияет выбор примитивов. Внешний вид, а следовательно, и изобразительные возможности примитивов, зависят от степени описывающих их функций (многочленов). Этот признак можно использовать для классификации моделей по форме составляющих их примитивов. Выделим модели, у которых описания примитивов имеют нулевую, первую и более высокую степени.

Классификация моделей пространственных объектов

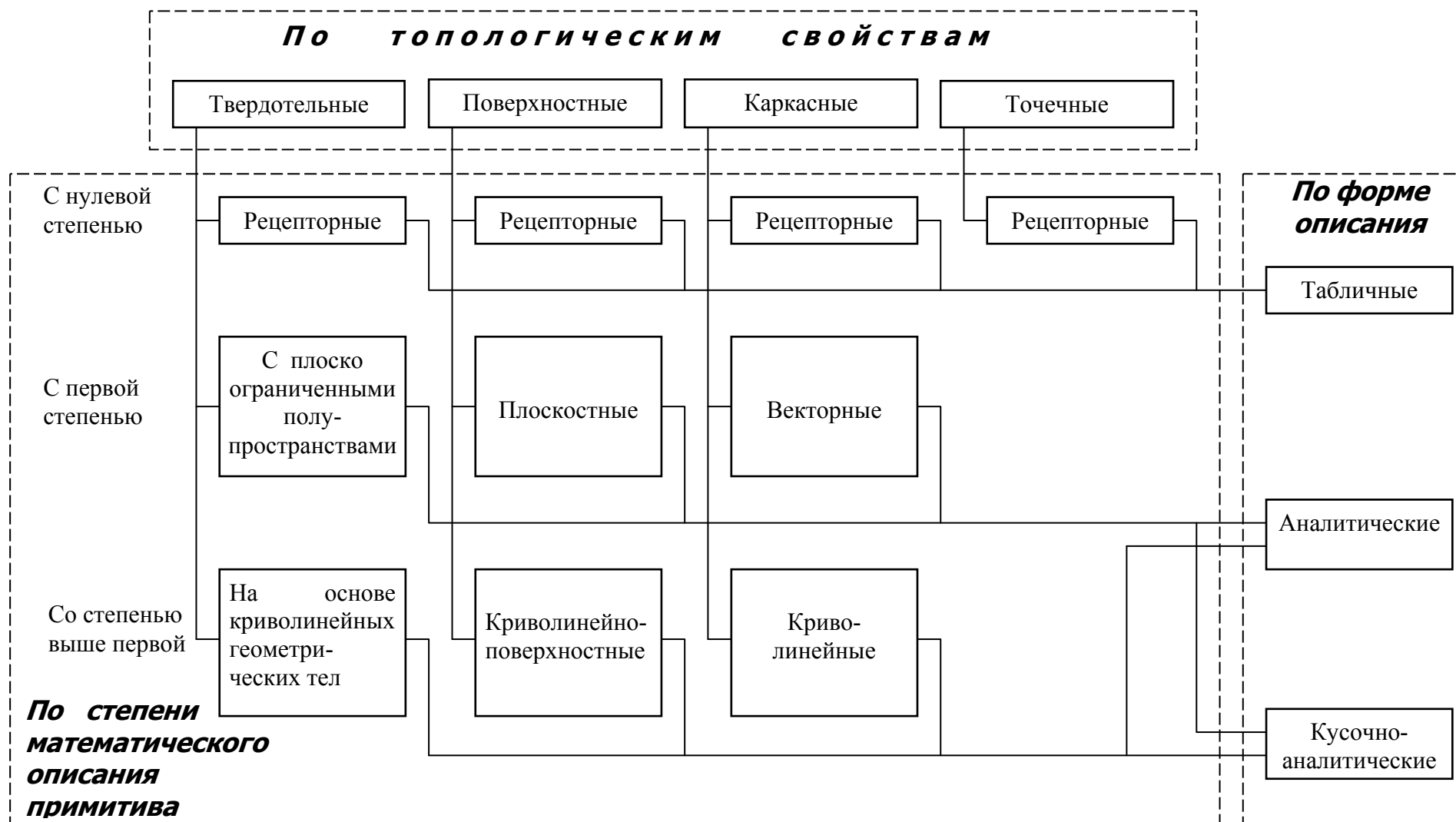


Рисунок 1.1 – Классификация и взаимосвязь моделей пространственных объектов

Описание нулевой степени характерно для точек. Набором точек, заданных с достаточной плотностью, можно представить поверхность любой сложности и формы. Набором пространственных элементов (вокселей) представляются и твердотельные объекты. Описанием таких наборов могут служить трехмерные матрицы, элементы которых указывают на принадлежность объекту элементов пространства, поставленных в соответствие элементам матрицы. Такие модели называют рецепторными. Многочлен первой степени в зависимости от числа аргументов описывает в пространстве прямую линию или плоскость. Отрезки прямых линий используются для представления поверхностей объектов в каркасных моделях, участки плоскости – в поверхностных. В твердотельных моделях примитивами могут служить полупространства, ограниченные плоскостями. Использование для описания моделей многочленов более высокой, чем первая, степени дает криволинейные примитивы. В каркасных моделях это кривые линии, в поверхностных – криволинейные поверхности, в твердотельных – участки пространства, ограниченные криволинейными поверхностями.

Отображаемые объекты редко состоят из одного примитива, как правило, они являются составными. С другой стороны, объекты могут быть весьма простыми, например, точечные. Ясно, что объекты различной сложности приходится описывать по-разному. Форма описания объектов, принятая в моделях, является еще одним важным признаком их классификации. По этому признаку выделяются табличные, аналитические и кусочно-аналитические модели.

В табличных моделях для описания объектов используют наборы пространственных координат элементов, характерных для этих объектов. Табличное описание может применяться в моделях со всеми перечисленными разновидностями топологических свойств. Аналитические модели описывают объекты с помощью аналитических выражений (уравнений). Применяются они для объектов простой формы, например, ограниченных поверхностями второго порядка. Кусочно-аналитические модели сочетают аналитическую и логическую формы описания действий. Они включают аналитическое описание примитивов и

сведения об их связях в объекте. Для описания "стыковки" примитивов в таких моделях применяют, например, операции над множествами, булевы функции, R-функции [2].

Далее рассматриваются только поверхностные модели. Именно они нашли преимущественное применение в интерактивной компьютерной графике. При этом не рассматриваются рецепторные модели, которые практически не применяются из-за больших потребностей в вычислительных ресурсах. Нужно отметить, что в компьютерной графике существует целый ряд моделей, не нашедших отражения в приведенной классификации. Это кинематические поверхности, поверхности свертки, фрактальные поверхности и ряд других [1,2,4]. Из-за большой длительности отрисовки они практически не применяются в интерактивных системах и по этой причине не рассматриваются.

2 ПРОЦЕСС ОТОБРАЖЕНИЯ ТРЕХМЕРНЫХ ОБЪЕКТОВ

2.1 Этапы процесса отображения

Если понимать процесс отображения в широком смысле, его можно разделить на два этапа: подготовительный этап и этап рендеринга. На подготовительном этапе составляются, кодируются и вводятся в графическую систему описания объектов сцены, то есть их математические модели. Выбор моделей осуществляется на основе подходов, освещенных в 1.2. На этапе рендеринга описания объектов по определенным алгоритмам преобразуются в изображения. На первом этапе отображения в графическую систему закладываются внешний вид и расположение объектов сцены, а на втором задается их поведение. Это поведение описывается с помощью целого ряда графических преобразований.

Выделяют модельные, видовые и растровые преобразования. Модельные (иначе – сценарные) преобразования оперируют объектами в целом, описывая изменение их положения в сцене. Видовые преобразования связаны с переводом

описаний объектов и примитивов в систему координат наблюдателя СКН (иначе – видовую систему координат), а также с формированием плоских проекций пространственных объектов на картинную (экранную) плоскость. С помощью растровых преобразований на экране дисплея получают реалистический образ объектов, привязанный к растровой решетке приборной системы координат.

Таким образом, отображаемые объекты на различных шагах своих преобразований представляются в различных координатных системах. Мировая система координат привязана к географическим координатам, видовая – к положению наблюдателя. В них описывается динамика объектов сцены в целом. Структуру объектов удобно описывать в своей – объектной – координатной системе СКО, в которой объекты остаются неизменными. Если различные объекты конструируются из одинаковых примитивов, то сами примитивы имеет смысл описать в своей системе координат – в системе координат примитива СКП. К ней будет привязана (или будет совпадать с ней) система координат текстуры СКТ, в которой представляется распределение цветояркости по площади примитива. Перечисленные координатные системы не зависят от особенностей технических средств отображения. Эти особенности учитывает приборная система координат.

Рассмотрим графические операции, выполняемые в рамках перечисленных преобразований.

2.2 Пространственное отсечение примитивов

В общем случае объекты сцены обладают шестью степенями свободы относительно наблюдателя. Аналогично наблюдатель, например пилот летательного аппарата, может обладать шестью степенями свободы относительно объектов. Используя принцип относительности, каждый из этих случаев можно свести к противоположному и рассматривать объекты сцены в той системе координат, в которой это будет более удобным.

В силу ограниченности области вывода изображения (экрана дисплея) не все объекты сцены попадают в поле зрения наблюдателя. Невидимые

наблюдателю объекты и их части выявляют с помощью операции отсечения и исключают из дальнейшего рассмотрения. В системах реалистического отображения сцена представляется в перспективе, поэтому видимыми считаются объекты и примитивы сцены, попадающие внутрь пространственного окна в виде усеченной пирамиды. Боковые грани пирамиды видимости проходят через точку наблюдения (это вершина пирамиды) и стороны экрана. Меньшее основание лежит в плоскости экрана, а большее основание параллельно плоскости экрана и отстоит от него на расстояние, зависящее от геометрических размеров объектов. В интерактивной графике в качестве примитивов обычно используются плоские полигоны, задаваемые своими ребрами. Поэтому задачей отсечения является, во-первых, выявить и отбросить объекты, целиком не попавшие в пирамиду видимости, а во-вторых, – установить видимые части ребер у тех примитивов, которые входят в частично видимые наблюдателем объекты.

Алгоритмы пространственного отсечения плоских примитивов достаточно хорошо проработаны в компьютерной графике [6,7]. Они относятся к так называемой базовой графике. Алгоритмы базовой графики закладываются в стандартные графические процедуры, поддерживаются техническими средствами графической системы (графическим процессором). Поэтому пользователь использует их в готовом виде.

В системах интерактивной графики операцию отсечения необходимо выполнять для каждой фазы динамики сцены (в пределе – для каждого кадра), то есть в режиме реального времени. Для упрощения и убыстрения алгоритмов отсечения объекты заключают в оболочки, представляющие собой простые геометрические тела: многогранники, например, параллелепипеды, или эллипсоиды [2,7]. Возможна иерархия оболочек. Это означает, что в сцене выделяют компактные группы объектов, которые охватывают общими оболочками. Если возможно, объединяют группы и охватывают эти объединения своими оболочками и т.д. Проверку попадания оболочек в пирамиду видимости начинают с верхнего уровня иерархии (с групп объектов), доходя до ее нижнего уровня (отдельных объектов). Применение оболочек может привести к

погрешностям в определении видимых наблюдателем объектов. Из-за неплотного прилегания оболочек к поверхностям объектов некоторые объекты, определенные при проверке как частично видимые, после дальнейшей обработки не попадут в поле зрения наблюдателя.

В современных графических системах отсечение выполняется в пространстве наблюдателя (в СКН). В этом пространстве грани пирамиды видимости имеют различное положение, что затрудняет определение видимости оболочек. Значительное упрощение этой операции достигается с помощью перспективной трансформации пространства наблюдателя [7]. Его суть заключается в том, что все оболочки искажаются в СКН по закону перспективы. В этом случае пирамида видимости преобразуется в параллелепипед видимости и перспективное проецирование оболочек на плоскость экрана может быть заменено параллельным проецированием. Число оболочек в сцене сравнительно невелико, и сама оболочка описывается небольшим числом геометрических параметров. Например, оболочка-параллелепипед описывается координатами восьми вершин. Поэтому пересчет оболочек по закону перспективной проекции не требует больших вычислительных затрат.

2.3 Преобразования пространственной динамики

Для выполнения отсечения и дальнейших графических операций в пространстве наблюдателя должны быть представлены все примитивы объектов, хотя бы частично попавших в объем видимости. Исходно примитивы описаны в СКО или СКП. Их перевод в СКН в графической системе выполняется на основе суперпозиции частных аффинных преобразований, учитывающих эволюции объекта в пространстве. Эти преобразования удобно записывать и выполнять в матричной форме в однородных координатах

$$R^* = R \cdot M, \quad (2.1)$$

где R – вектор-строка исходных координат: $R = \begin{vmatrix} x & y & z & 1 \end{vmatrix}$;

R^* – вектор-строка пересчитанных координат (h – скалярный множитель):

$$R^* = \begin{vmatrix} x^*h & y^*h & z^*h & h \end{vmatrix};$$

M – матрица преобразований размера 4×4 .

К основным аффинным преобразованиям относятся масштабирование, сдвиг и поворот в пространстве (есть еще отражение, или симметрия). Кроме них, до или после операции отсечения выполняется центральное проецирование примитивов на картинную плоскость. Оно не относится к аффинным преобразованиям, но описывается сходным образом и потому рассматривается вместе с ними. Матрицы преобразований масштабирования (DL), сдвига (TR), поворота (RT) и проецирования (PR) имеют следующий вид:

$$\begin{aligned} DL &= \begin{vmatrix} M_x & 0 & 0 & 0 \\ 0 & M_y & 0 & 0 \\ 0 & 0 & M_z & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, & TR &= \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_0^* & y_0^* & z_0^* & 1 \end{vmatrix}, \\ RT &= \begin{vmatrix} t_{11} & t_{21} & t_{31} & 0 \\ t_{12} & t_{22} & t_{32} & 0 \\ t_{13} & t_{23} & t_{33} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, & PR &= \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/z_v^* \\ 0 & 0 & 0 & 1 \end{vmatrix}, \end{aligned} \quad (2.2)$$

где M_x, M_y, M_z – коэффициенты масштабирования по координатным осям;

x_0^*, y_0^*, z_0^* – координаты вектора, соединяющего начала координат наблюдателя и объекта;

t_{11}, \dots, t_{33} – направляющие косинусы осей СКО в СКН;

z_v^* – расстояние наблюдения экранной плоскости.

Направляющие косинусы являются функциями углов поворота СКО вокруг своих координатных осей: x (угол φ), y (ψ) и z (θ). В зависимости от отсчета углов и последовательности поворотов выражения для расчета направляющих косинусов имеют различный вид. В компьютерной графике сложные повороты представляют в виде сочетания частных случаев поворота: поворотов вокруг координатных осей. Вид описывающих эти преобразования матриц зависит от расположения начала координат и точки наблюдения в пространстве наблюдателя.

Различают правую и левую системы координат, в которых матрицы преобразований выглядят по-разному. В правой системе координат поворот оси x к оси y , оси y к оси z , оси z к оси x осуществляются против часовой стрелки, если смотреть из первого октанта. В левой системе координат эти повороты осуществляются по часовой стрелке. Если начало СКН лежит в плоскости экрана, ось глубины (ось z^v) проходит через его центр, а точка наблюдения расположена на отрицательной полуоси глубины, то имеет место левая система координат. В ней матрицы преобразований поворота имеют вид:

$$RT_x = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & -\sin \varphi & 0 \\ 0 & \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad RT_y = \begin{vmatrix} \cos \psi & 0 & \sin \psi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \psi & 0 & \cos \psi & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad RT_z = \begin{vmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

В правой системе координат они выглядят следующим образом:

$$RT_x = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi & 0 \\ 0 & -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad RT_y = \begin{vmatrix} \cos \psi & 0 & -\sin \psi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \psi & 0 & \cos \psi & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad RT_z = \begin{vmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

Результирующая матрица преобразований M , в общем случае, определяется произведением (суперпозицией) приведенных матриц. Матрицы включаются в суперпозицию слева направо в соответствии с последовательностью описываемых преобразований. Например, необходимо найти местоположение на экране пространственной точки P с координатами (x_p, y_p, z_p) после выполнения следующих преобразований: 1) поворот на угол β вокруг оси, параллельной оси глубины и проходящей через точку экрана с координатами (x_a, y_a) , 2) перспективное проецирование на плоскость экрана. Рассмотренная матрица RT_z предназначена для поворота вокруг оси z , следовательно, для применения RT_z нужно совместить ось поворота точки P с осью глубины. Для этого следует сдвинуть ее по горизонтали и вертикали на $(-x_a), (-y_a)$, соответственно. Такой же сдвиг получит сама точка P . Для этого осуществляется умножение ее координат на матрицу $TR1$. Далее нужно выполнить поворот на угол β , а затем устранить последствия первого сдвига, вернув точку P на $(+x_a), (+y_a)$ – умножение на $TR2$. Последним

действием должно быть проецирование, выполняемое умножением на матрицу PR . В результате динамика точки P описывается выражением (2.1), где матрица преобразований M имеет вид: $M=TR1 \cdot RT \cdot TR2 \cdot PR$. Входящие в суперпозицию матрицы сдвига и поворота выглядят так:

$$TR1 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_a & -y_a & 0 & 1 \end{vmatrix}, \quad TR2 = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x_a & y_a & 0 & 1 \end{vmatrix}, \quad RT_z = \begin{vmatrix} \cos \beta & -\sin \beta & 0 & 0 \\ \sin \beta & \cos \beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}.$$

2.4 Определение видимости примитивов

Проецированию на картинную плоскость подвергаются только лицевые примитивы, то есть примитивы, обращенные к наблюдателю лицевой стороной. Лицевой считается сторона, к которой проведена внешняя (по отношению к объекту) нормаль. Нелицевые примитивы выявляются и отбрасываются, чтобы далее не тратить время на их обработку. Алгоритм выявления нелицевых примитивов предполагает анализ угла β между лучом зрения \bar{L} , падающим в некоторую точку примитива, и внешней нормалью \bar{N} , проведенной через эту точку. Острый угол β говорит о видимости грани, прямой и тупой – о ее невидимости. О величине угла судят по знаку его косинуса, который получают с помощью скалярного произведения векторов \bar{L} и \bar{N} :

$$\cos \beta = \frac{\bar{L} \cdot \bar{N}}{|\bar{L}| \cdot |\bar{N}|}.$$

При $\text{sgn}(\bar{L} \cdot \bar{N}) > 0$ примитив лицевой, в противном случае – нелицевой.

Векторная алгебра дает следующее выражение для нахождения скалярного произведения векторов \bar{L} и \bar{N} :

$$\bar{L} \cdot \bar{N} = l_x n_x + l_y n_y + l_z n_z,$$

где l_x, l_y, l_z и n_x, n_y, n_z – соответственно координаты векторов \bar{L} и \bar{N} , то есть их проекции на координатные оси.

Выявление нелицевых примитивов можно значительно упростить, проведя перспективную трансформацию пространства. В этом случае перспективное проецирование заменяется параллельным, и все лучи зрения становятся параллельными оси глубины. Тогда $l_x=l_y=0$, а $l_z=\overline{L}$. Координаты нормали к некоторой поверхности определяются частными производными функции $F(x,y,z)$, которая описывает эту поверхность. Если примитив плоский, то описывающая его функция имеет вид:

$$F(x,y,z)=Ax+By+Cz+D,$$

где A,B,C,D – числовые коэффициенты, задающие положение плоскости.

В этом случае координата n_z определяется выражением

$$n_z = \frac{\partial F}{\partial z} = C$$

и признак видимости vp получает весьма простой вид: $vp=\text{sgn } C$.

Если лицевые примитивы в пространстве наблюдателя загорожены впереди стоящими объектами, их изображения полностью или частично не должны появиться на экране. Для этого предназначена большая группа алгоритмов удаления невидимых участков [6,7]. В их основе лежит анализ глубины расположения участков различных примитивов. Видимыми считаются участки, наиболее близкие к точке наблюдения. В современных графических системах преимущественное применение нашли алгоритм z -буфера (буфера глубины) и его модификации, работающие в процессе проецирования элементов (точек) примитивов на плоскость экрана. По вычисленным в процессе проецирования координатам очередного элемента в буферной памяти выбирается многоразрядная ячейка, в которую записываются код цветояркости элемента и код его удаления от наблюдателя (код глубины). Если со временем элемент другого примитива получит тот же адрес в памяти буфера, то его код удаления будет сравниваться с кодом удаления, занесенным в буфер по этому адресу ранее. По результатам сравнения в буфер будет занесена информация об элементе с меньшим кодом глубины, то есть об элементе, более близком к точке наблюдения. В итоге, когда содержимое буфера будет выводиться на экран, на засветку каждого пикселя

экрана поступит элемент примитива, наиболее близкий к наблюдателю из всех элементов, лежащих на одном проекторе. Удаление загороженных участков примитивов с помощью алгоритма *z*-буфера сопровождается растровые преобразования. В ходе этих преобразований все элементы поверхности объектов отображения находят свое место на растровом экране дисплея с реалистичными значениями атрибутов.

2.5 Растровые преобразования

Растрирование не относится к **геометрическому** моделированию, но методы моделирования всегда увязываются с растровыми преобразованиями, так как они вместе образуют единый процесс отображения объектов. В связи с этим растровые преобразования рассматриваются обзорно.

Растровые преобразования (растрирование, растеризация) – это процесс конвертации уже преобразованных в экранные координаты примитивов в атрибуты пикселей, принадлежащих области вывода. Растрирование включает операции фрагментации примитива, его текстурирования, удаления его заслоненных участков (что уже описано), наложения на него теней и спецэффектов.

С помощью фрагментации проекция примитива на экран разбивается на отдельные пиксели, "привязанные" к растровой решетке. Фрагментация задает пиксельные позиции в буфере кадра. Для каждой пиксельной позиции в буфер заносится фрагмент – набор кодов, определяющих основные характеристики пикселя (цветояркость, глубину, текстурные координаты и др.). В современных графических системах объект передается на растеризацию в полигональной форме. В разных системах полигоны разбиваются на фрагменты по-разному. Чаще всего применяется выделение хорд и квадов. Хорда – участок строки раstra, находящийся внутри проекции полигона на плоскость экрана. Квад – пиксельный участок размером 2×2 , принадлежащий проекции полигона. Например, фрагментация хорд организуется следующим образом. В процессе нахождения

хорд сначала вычисляют координаты пикселей, образующих ребра полигонов-проекций на экране. Для этого может быть использована одна из модификаций алгоритма Брезенхема. Сортировка полученных координат по строкам, а внутри их – по позициям пикселей дает список, каждая пара элементов которого определяет хорду. В процессе продвижения по хорде определяют глубину, цвет и другие характеристики пикселей.

Для повышения реалистичности изображения объекта на его поверхность накладывают характерный рисунок – текстуру. Текстура в самом простом случае это двумерный массив, элементами которого являются точки узора – тексели. Тексельный массив (текстурная карта) привязывается к своей системе координат – СКТ. В памяти графической системы текселям сопоставляются коды, описывающие их визуальные атрибуты. Детальная текстура формируется с помощью библиотеки текстурных карт. Суть текстурирования – в отыскании кода цветояркости для каждого элемента изображения. Для этого сначала находится функция отображения, устанавливающая однозначное соответствие между системами координат текстуры и примитива. Затем в процессе отображения пиксель экрана, в который проецируется элемент примитива, получает цветояркость соответствующего текселя. На рисунке 2.1 показано наложение текстуры, заданной в своей координатной системе s, t , на треугольный примитив, представленный в системе координат u, v .

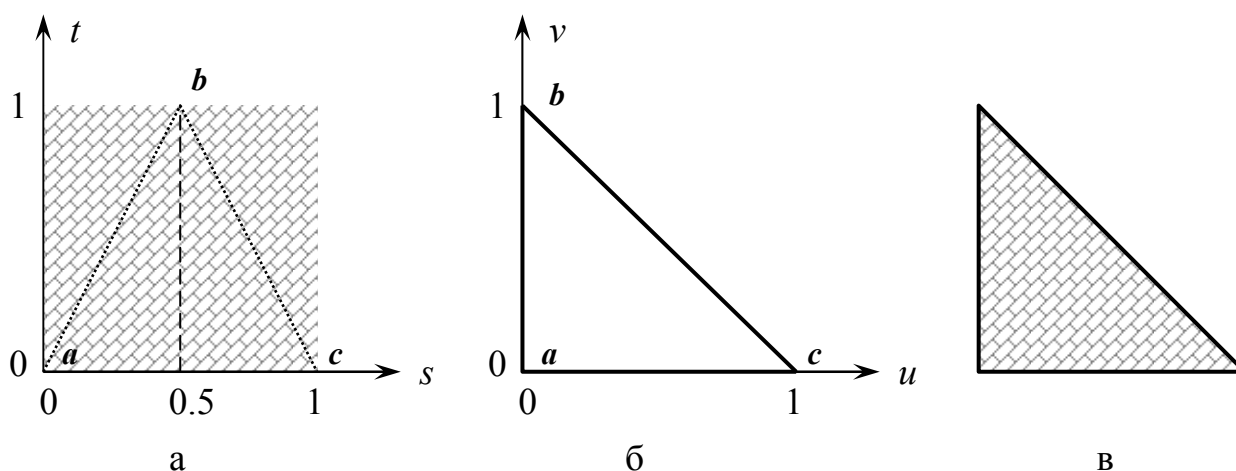


Рисунок 2.1 – Иллюстрация к определению функции отображения для наложения текстуры: представление текстуры (а) и полигона (б) в своих системах координат и результат текстурирования (в)

Функция отображения определяется пользователем и может быть как линейной, так и нелинейной. Например, для случая, показанного на рисунке 2.1, соответствие между системами координат примитива и текстуры может быть задано парой линейных уравнений

$$s = Au + Bv + C, \quad (2.3)$$

$$t = Du + Ev + F, \quad (2.4)$$

где A, \dots, F – шесть числовых коэффициентов функции отображения. Для их отыскания необходимо сформулировать ограничения на численные значения координат и представить их в виде шести уравнений с этими коэффициентами. В качестве ограничений для примера на рисунке 2.1 установим соответствие элементов примитива и текстуры, обозначенных одинаковыми буквами – a, b и c : для точки a выражение (2.3) – $0=A \cdot 0+B \cdot 0+C$, выражение (2.4) – $0=D \cdot 0+E \cdot 0+F$, для точки b выражение (2.3) – $0.5=A \cdot 0+B \cdot 1+C$, выражение (2.4) – $1=D \cdot 0+E \cdot 1+F$, для точки c выражение (2.3) – $1=A \cdot 1+B \cdot 0+C$, выражение (2.4) – $0=D \cdot 1+E \cdot 0+F$. Решение уравнений дает значения коэффициентов: $A=E=1$, $B=0.5$, $C=D=F=0$. В итоге функция отображения принимает вид

$$s=u+0.5v,$$

$$t=v,$$

позволяющий вычислить текстурные координаты для любой точки примитива.

Линейная функция отображения реализуется наиболее просто и не требует от вычислительной системы больших ресурсных затрат. Но она дает и самое низкое качество текстурирования. Более качественное, но более медленное текстурирование дает интерполяция текстурных координат квадратичной параболой и сплайн-функцией.

Для криволинейных примитивов, например, поверхностей второго порядка, функция отображения принципиально является нелинейной. Ее вычисление нужно выполнить для каждого элемента поверхности-примитива, что требует довольно много вычислительного времени. Кроме того, с целью повышения реалистичности отображения в современных графических системах применяют мультитекстурирование, то есть смешивание на одном примитиве нескольких

текстур. Этот прием резко увеличивает вычислительные затраты на текстурирование, поэтому криволинейные примитивы после геометрических преобразований обычно аппроксимируются плоскими полигонами. Полигоны проецируются на плоскость экрана тоже в виде полигонов. Их закрашка на экране идет по хордам или квадам, принадлежащим проекции примитива. Для ускорения процесса текстурирования точные значения текстурных координат вычисляют только для граничных фрагментов, а для промежуточных фрагментов текстурные координаты находят приближенно с помощью интерполяции.

При удалении объекта в СКН его видимые размеры на экране уменьшаются. Это означает, что несколько элементов текстуры проецируются в один пиксель. Очевидно, в этом случае каждый пиксель должен быть засвечен с некоторой усредненной цветояркостью, иначе возникнут цветовые искажения изображения. Чтобы не проводить усреднение цветояркостей в режиме реального времени, его выполняют заранее. На предварительном этапе отображения создается иерархия (пирамида) текстур различной детальности. Нулевой уровень иерархии составляет исходная текстура, а текстура каждого следующего уровня получается путем сжатия в 4 раза (в два раза по каждой координате) текстуры предыдущего уровня. В процессе сжатия проводится усреднение цветояркости текселей. На самом верхнем уровне иерархии находится текстура, соответствующая максимальному удалению текстурируемого объекта. На этапе реального времени после определения удаления примитива от наблюдателя из пирамиды текстур выбирается текстура, соответствующая этому удалению. Текстурирование с использованием пирамиды текстур имеет название *mip-mapping* [7]. Иерархия текстур увеличивает потребные для их хранения затраты памяти на одну треть. В современных графических системах наложение текстуры поддерживается аппаратно.

Чтобы повысить реалистичность отображения трехмерных объектов, нужно смоделировать их освещенность от одного или нескольких источников света. Точный расчет освещенности, основанный на учете физических законов поглощения, преломления и отражения света, весьма сложен, поэтому на

практике пользуются упрощенными моделями освещенности. Одна из наиболее простых моделей учитывает освещенность источником рассеянного света, а также диффузную и бликовую компоненты освещенности точечным источником света [7]. В расчете интенсивности I элемента поверхности примитива учитываются интенсивности I_p , I_T этих источников света, расстояние d от наблюдателя до примитива, взаимное расположение источника света, наблюдателя и примитива, а также свойства поверхности примитива:

$$I = I_p k_{\text{др}} + \frac{I_T}{d + K} (k_{\text{дт}} \cos \theta + k_3 \cos^n \alpha),$$

где $k_{\text{др}}, k_{\text{дт}}$ – коэффициенты диффузного отражения рассеянного света и света от точечного источника;

k_3 – коэффициент зеркального отражения;

K – экспериментально подбираемая константа;

θ – угол между падающим лучом света и нормалью к поверхности;

α – угол между отраженным от поверхности лучом и лучом зрения наблюдателя;

n – показатель степени, задающий пространственное распределение отраженного света (зависит от гладкости поверхности).

Для формирования цветного изображения интенсивность отдельно рассчитывается для каждого цветового компонента. Если точечных источников света несколько, то создаваемые ими освещенности складываются. Более точный учет закономерностей распространения света при расчете освещенности еще более увеличивает объем вычислений. Выполнять их для каждого пикселя в режиме реального времени слишком сложно, поэтому в современных графических системах применяют расчет освещенности с интерполяцией. Наиболее часто используется модель освещенности Гуро. Ее сущность заключается в том, что для каждой фазы динамики примитива точные значения освещенности вычисляются только в его вершинах. На ребрах, соединяющих вершины, освещенность вычисляется с помощью линейной интерполяции. Отрезок строки раstra внутри полигона (хорда) пересекает ребра проекции

полигона в двух точках, по координатам которых находятся интерполяционные коэффициенты. Они используются для расчета освещенности в конечных точках хорды, а в ее промежуточных точках освещенность также определяется путем линейной интерполяции вдоль хорды.

Закраска Гуро обладает рядом недостатков, но благодаря своей простоте применяется в большинстве графических систем. Делаются попытки реализовать в режиме реального времени более качественную, но и более сложную модель освещенности – модель Фонга. Она предполагает билинейную интерполяцию положения нормали по площади примитива между положениями нормали в его углах.

2.6 Графический конвейер

Перечисленные преобразования могут выполняться в различной последовательности и по различным алгоритмам, что зависит от выбранного метода отображения. Различают два основных метода: метод обратной трассировки лучей и метод прямого проецирования на экран. Они иллюстрируются рисунком 2.2, где индексы v , o , p указывают на принадлежность к СКН, СКО и СКП. Суть первого метода заключается в отыскании цветояркости для засветки каждого пикселя экрана. Упрощенный алгоритм метода таков:

- после выполнения сценарных преобразований организуется последовательный перебор пикселей;
- для текущего пикселя определяется проектор. Это луч, исходящий из центра проецирования (из точки наблюдения V) и проходящий через текущий пиксель ЭО (элемент отображения);
- находятся точки пересечения проектора с геометрическими примитивами (на рисунке – $P1, P2$), и из них выбирается точка, наиболее близкая к центру проецирования, в данном случае это элемент примитива ЭП;
- найденной точке примитива ставится в соответствие элемент текстуры ЭТ. Его код цветояркости, например, считывается из памяти (из СКТ);

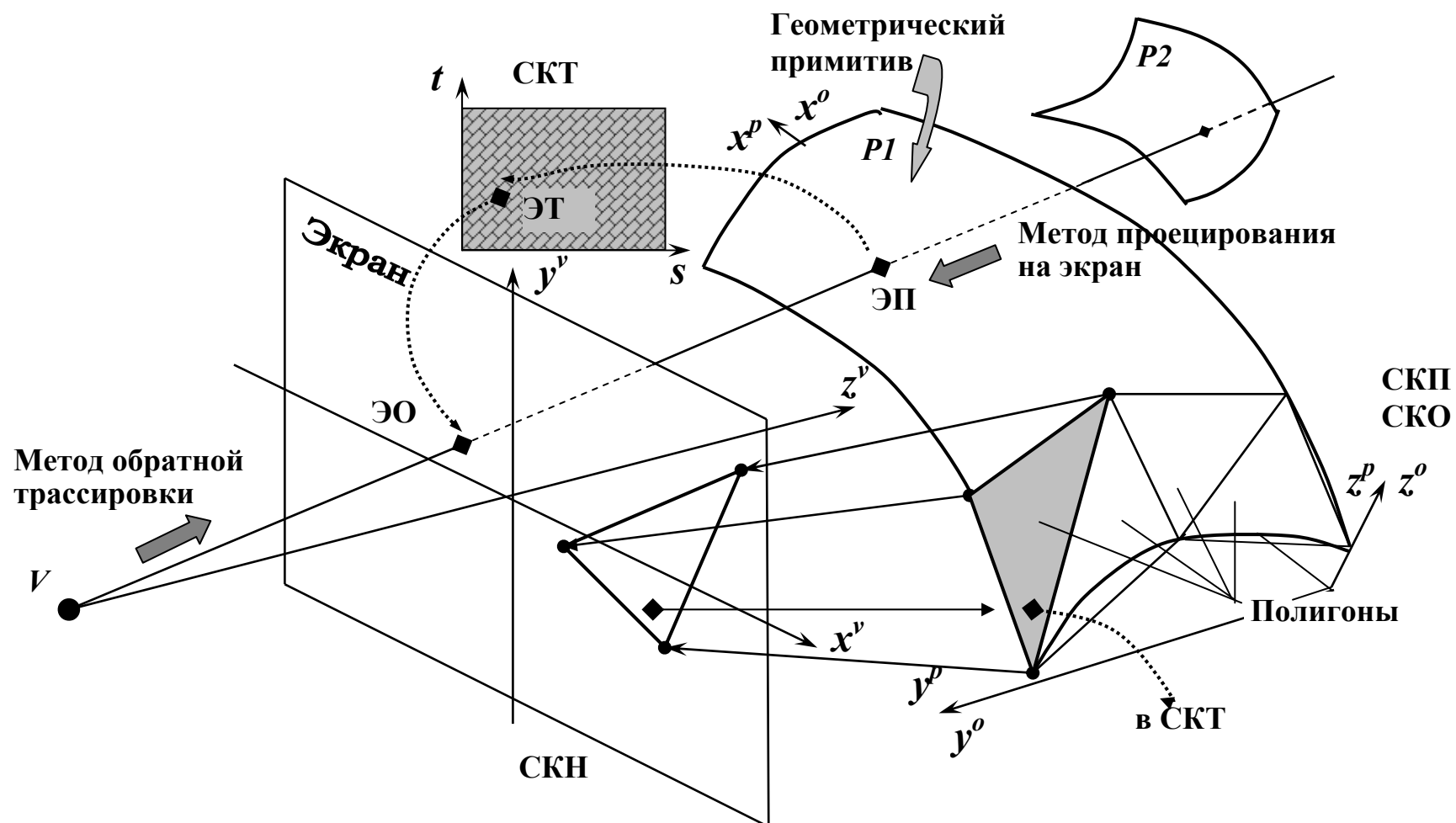


Рисунок 2.2 – Два метода отображения пространственных объектов

- код цветояркости заносится в буфер кадра по адресу текущего пикселя.

В результате каждый пиксель информационного поля находит свое "наполнение". Метод обратной трассировки позволяет получать изображения фотографического качества, но характеризуется большими вычислительными затратами. Они обусловлены, во-первых, необходимостью проводить все расчеты для множества фоновых ("пустых") точек растра, а во-вторых, – длительным поиском ближайшей точки пересечения проектора с многочисленными примитивами сцены.

Суть метода прямого проецирования заключается в отыскании на экране места (пикселя) для отображения каждой точки объекта. Упрощенный алгоритм имеет следующие шаги:

- после выполнения сценарных преобразований организуется перебор геометрических примитивов, попавших в зону видимости наблюдателя. У каждого примитива рассматриваются все или только характерные точки (это зависит от информационной модели примитива);
- для текущей точки примитива ЭП в СКТ определяется код цветояркости и проводится проектор, соединяющий текущую точку с центром проецирования V ;
- находится точка пересечения проектора с плоскостью экрана – это пиксель ЭО, на который отображается текущая точка примитива;
- код цветояркости текущей точки заносится в буфер кадра по адресу найденного пикселя;
- в процессе проецирования возможно совпадение проекторов для нескольких точек, принадлежащих разным примитивам. В этом случае сравнивается удаление этих точек от центра проецирования, и в буфере кадра остается код цветояркости точки, самой близкой к центру проецирования.

Второй метод отображения реализуется быстрее первого, но имеет меньшие изобразительные возможности.

В реальных графических системах применяется комбинация описанных методов, что также отражено на рисунке 2.2 (внизу). Сначала по второму методу

организуется проецирование точек примитивов (на рисунке – полигонов) на плоскость экрана и отыскание точек, ближайших к наблюдателю. Для моделирования освещенности объекта в этих точках находятся нормали к поверхности примитивов. Затем рассматриваются хорды или квады внутри проекций примитивов. Они путем обратной трассировки "привязываются" к примитивам, а через них – к текстурам. При считывании текселов выполняется усреднение их цветояркостей.

Рассмотренные графические операции образуют естественный конвейер: данные проходят последовательно несколько этапов обработки, причем выходные данные одного этапа сразу передаются на вход следующего. В современных графических системах этапы конвейера распределяются между центральным процессором (хост-процессором) системы и ее аппаратной частью – графическим процессором. Обобщенная структура графического конвейера приведена на рисунке 2.3. В качестве геометрических примитивов здесь предполагаются плоские полигоны.

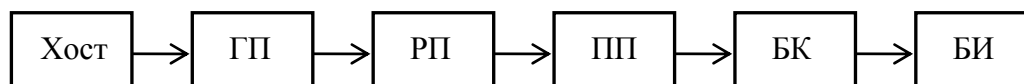


Рисунок 2.3 – Обобщенная структурная схема графической системы

Хост-процессор выполняет сценарные преобразования и формирует список примитивов. В геометрическом процессоре (ГП) примитивы подвергаются геометрическим преобразованиям. Дополнительно для вершин полигонов вычисляются атрибуты закрашки. Растровый процессор (РП) вычисляет адреса и атрибуты пикселей, в которые проецируются элементы примитивов в плоскости раstra. Для повышения качества текстурирования каждый пиксель может рассматриваться как набор более мелких элементов – субпикселей. Пакет данных, характеризующих один пиксель, называется фрагментом, поэтому растровый процессор еще называют фрагментным процессором. Описания фрагментов поступают на пиксельный процессор (ПП), выявляющий видимые наблюдателем пиксели с помощью алгоритма z-буфера и определяющий цветояркость каждого

видимого пикселя. Атрибуты видимых пикселей заносятся в буфер кадра (БК), из которого с частотой регенерации поступают на блок индикации (БИ). Если на каком-то этапе обработки производительности аппаратуры не хватает, прибегают к распараллеливанию этого этапа. Для отображения сложных сцен зачастую не хватает производительности одного графического конвейера, в большинстве случаев из-за ограниченной пропускной способности памяти БК. Тогда для получения желаемого ускорения необходимо поставить в параллель несколько конвейеров, например, повторив несколько раз все их компоненты.

3 ГЕОМЕТРИЧЕСКИЕ ПРИМИТИВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ

3.1 Полигоны

Плоскостные (полигональные) модели, или полигональные сетки, находят в компьютерной графике самое широкое применение. Они представляют поверхности геометрических объектов в виде набора состыкованных друг с другом плоских полигонов. Традиционное для компьютерной графики описание полигональной модели объекта является иерархическим и включает список вершин, список ребер и список полигонов объекта.

Нижний уровень иерархии – список вершин V :

$V = \{v_1, v_2, \dots, v_L\}$, где $v_i = (x_i, y_i, z_i)$, L – количество вершин в объекте;

средний уровень – список ребер E :

$E = \{e_1, e_2, \dots, e_M\}$, где $e_j = (pv_{j1}, pv_{j2}, f_{j1}, f_{j2})$, M – количество ребер в объекте;

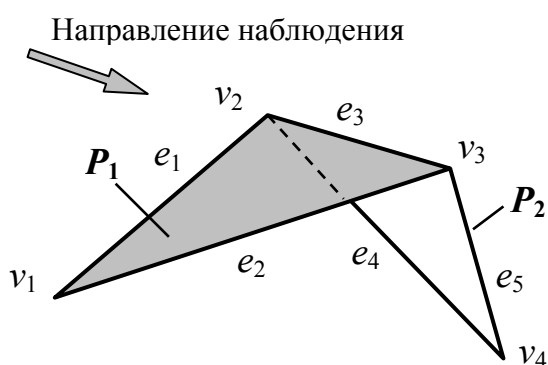
верхний уровень – список полигонов P :

$P = \{p_1, p_2, \dots, p_N\}$, где $p_k = (pe_{k1}, pe_{k2}, \dots, pe_{kR})$, N – количество полигонов в объекте, R – количество ребер в k -ом полигоне.

Каждая вершина v_i , внесенная в список V , описывается тройкой пространственных координат. Ребро объекта e_j ограничивается двумя вершинами и считается принадлежащим одному или двум полигонам. Это зависит от видимости смежных полигонов. Описание ребра включает, во-первых, две

ссылки pv_{j1}, pv_{j2} на элементы списка вершин, относящиеся к начальной и конечной вершинам j -го ребра, а во-вторых, – два признака f_{j1}, f_{j2} входимости ребра в смежные полигоны. Признаки входимости ребра в смежные полигоны в процессе динамики объекта получают нулевое или единичное значение и показывают, при визуализации какого полигона – первого или второго – должно быть нарисовано это ребро. Полигон объекта p_k представляется набором своих ребер и описывается ссылками на соответствующие элементы списка ребер.

В качестве примера на рисунке 3.1 показан простейший пространственный объект, состоящий из двух полигонов P_1, P_2 , и приведено его иерархическое описание.



$$V = \{(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3), (x_4, y_4, z_4)\};$$

$$E = \{(pv_1, pv_2, 1, 0), (pv_1, pv_3, 1, 0), (pv_2, pv_3, 1, 0), (pv_2, pv_4, 0, 0), (pv_3, pv_4, 0, 0)\};$$

$$P = \{(pe_1, pe_2, pe_3), (pe_3, pe_4, pe_5)\}.$$

Рисунок 3.1 – Полигональный объект и его иерархическое описание

При направлении наблюдения, принятом на рисунке 3.1, примитив P_2 является нелицевым и потому не должен отображаться. Об этом говорят нулевые значения вторых признаков входимости в описании каждого ребра P_2 . Иерархическое описание объектов позволяет снизить затраты памяти на хранение координат их вершин, исключить повторное вычерчивание одних и тех же ребер и, в то же время, просто отображать отдельные полигоны.

Для моделирования освещенности полигонального объекта необходимо знать направления нормалей к примитивам. В декартовой системе координат вектор нормали \bar{N} , как и любой другой вектор, задается тройкой значений n_x, n_y, n_z (координат нормали), которые представляют собой величины проекций \bar{N} на координатные оси. Математическое описание нормали имеет вид

$$\bar{N} = n_x i + n_y j + n_z k ,$$

где i, j, k – единичные векторы, направленные вдоль координатных осей x, y, z , соответственно (орты).

Из векторной алгебры известно, что нормальный вектор плоскости находится через векторное произведение двух векторов, лежащих в этой плоскости. В рассматриваемом случае в качестве таких векторов выступают векторы $\overline{R1}, \overline{R2}$, совпадающие с ребрами полигона, как это показано на рисунке 3.2. Вектор \overline{N} определяется выражением

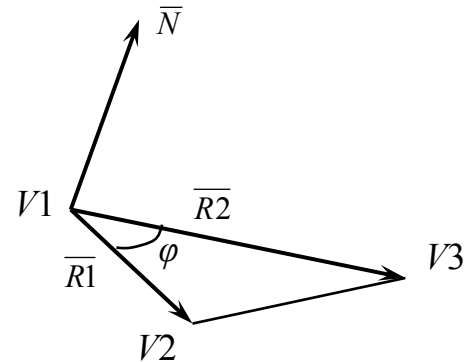


Рисунок 3.2 – Определение нормали к полигону

$$\overline{N} = \overline{R1} \times \overline{R2} = |\overline{R1}| \cdot |\overline{R2}| \cdot \sin \varphi,$$

но более рационально находить его с помощью определителя 3-го порядка [1]

$$\overline{N} = \begin{vmatrix} i & j & k \\ r_{1x} & r_{1y} & r_{1z} \\ r_{2x} & r_{2y} & r_{2z} \end{vmatrix},$$

где r_{1x}, \dots, r_{2z} – координаты векторов $\overline{R1}, \overline{R2}$.

Координаты n_x, n_y, n_z вектора нормали находятся из миноров элементов i, j, k , соответственно, то есть из определителей 2-го порядка, полученных из исходного определителя вычеркиванием первой строки и i -го, j -го, k -го столбца. Выражения имеют следующий вид

$$\begin{aligned} n_x &= \begin{vmatrix} r_{1y} & r_{1z} \\ r_{2y} & r_{2z} \end{vmatrix} = (y_2 - y_1)(z_3 - z_1) - (y_3 - y_1)(z_2 - z_1), \\ n_y &= \begin{vmatrix} r_{1x} & r_{1z} \\ r_{2x} & r_{2z} \end{vmatrix} = (x_2 - x_1)(z_3 - z_1) - (x_3 - x_1)(z_2 - z_1), \\ n_z &= \begin{vmatrix} r_{1x} & r_{1y} \\ r_{2x} & r_{2y} \end{vmatrix} = (x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1), \end{aligned}$$

где x_n, y_n, z_n – координаты n -ой вершины полигона ($n=1,2,3$).

Полигональное представление сложных поверхностей требует большого числа полигонов. Для снижения временных затрат на их обработку в

графических системах применяют различные модификации полигональных моделей. Например, сложные поверхности описывают аналитически или кусочно-аналитически, а затем разбивают на плоские треугольники, то есть проводят триангуляцию. В результате геометрические преобразования выполняются над поверхностями в целом, а вывод идет по простым алгоритмам визуализации полигонов. Для убыстрения визуализации используют последовательности связанных полигонов – стрипы (strip). Для визуализации каждого следующего треугольника в стрипе достаточно задать лишь одну дополнительную вершину, в то время как при визуализации его вне стрипа надо задать три вершины. Моделирование поверхностей на основе полигональных стрипов поддерживается популярными графическими библиотеками DirectX и OpenGL [8] и графическим процессором персонального компьютера.

3.2 Поверхности второго порядка

Основным недостатком полигональных моделей является необходимость в большом количестве полигонов для представления сложных, особенно криволинейных, поверхностей. Реалистичное отображение пространственной сцены может потребовать числа полигонов, достигающего до миллиона, и более. Это означает, что при синтезе динамических изображений геометрические параметры нужно пересчитывать у большого числа примитивов в режиме реального времени. Поэтому наряду с плоскими поверхностями в графических системах применяются криволинейные примитивы, в частности, поверхности второго порядка.

Криволинейно-поверхностные аналитические модели в виде уравнений второй степени используют для описания поверхностей второго порядка (квадрик): эллипсоида (в частном случае – сферы), параболоида, гиперболоида, цилиндра, конуса. Математически поверхности могут быть представлены в явной, неявной и параметрической формах. Со школы известны канонические уравнения поверхностей, например, уравнение сферы:

$$x^2 + y^2 + z^2 = R^2,$$

где R – значение радиуса.

Из канонической формы легко получить неявную (общую) и явную формы описания.

Неявная форма: $F(x, y, z) = x^2 + y^2 + z^2 - R^2 = 0,$

явная форма: $z(x, y) = \pm \sqrt{R^2 - x^2 - y^2}.$

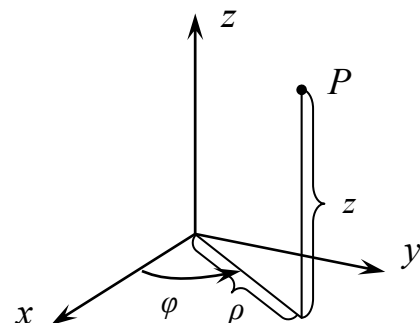
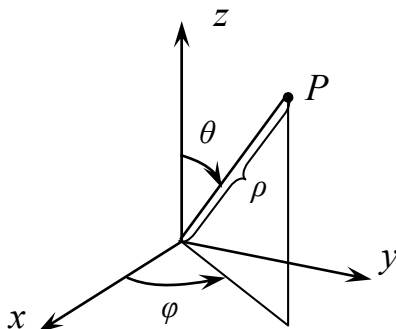
Неявная форма неудобна тем, что для визуализации требует применения ресурсоемкого метода обратной трассировки лучей, то есть нужно организовать в СКН перебор троек координат, для каждой тройки вычислять значение $F(x, y, z)$ и сравнивать его с нулем. Явная форма описания характеризуется многозначностью, так как каждой паре x, y ставит в соответствие не одно, а два значения z (положительное и отрицательное). Это усложняет дальнейший переход к полигональной модели поверхности, так как последовательно вычисляемые точки не являются соседними. В связи с названными причинами в компьютерной графике, как правило, используют параметрическую форму описания квадрик [5].

Вариантов параметрического описания одной и той же поверхности может быть несколько. Все зависит от выбранного закона обхода поверхности. Например, для описания квадрик в качестве аргументов-параметров могут быть выбраны сферические или цилиндрические координаты. Варианты описания сферы в этих координатах и их связь с декартовыми координатами показаны ниже.

Сферические координаты ρ, φ, θ | **Цилиндрические координаты ρ, φ, z**

ρ – радиус, φ – долгота, θ – полярное расстояние, z – высота

Представление точки



Уравнение сферы

Обход поверхности по параллелям
или меридианам

$$\rho = R, \quad \varphi = 0..2\pi, \quad \theta = 0..\pi.$$

Обход поверхности по сечениям,
перпендикулярным оси z

$$\rho = \sqrt{R^2 - z^2}, \quad \varphi = 0..2\pi, \quad z = -R..+R.$$

Связь с декартовыми координатами

$$x = \rho \sin \theta \cos \varphi,$$

$$y = \rho \sin \theta \sin \varphi,$$

$$z = \rho \cos \theta.$$

$$x = \rho \cos \varphi,$$

$$y = \rho \sin \theta,$$

$$z = z.$$

Поверхности второго порядка, за исключением эллипсоида, не локализованы в пространстве и простираются в бесконечность. По этой причине при включении в графические объекты они нуждаются в ограничении другими примитивами. На практике для ограничения поверхностей обычно используют ограничение предельных значений их параметров.

Для параметрического описания поверхностей второго порядка могут быть выбраны различные параметры. В любом случае координаты текущих точек отсчитываются в какой-то криволинейной системе координат. Координатная сетка криволинейной системы координат лежит на поверхности и для поверхностей второго порядка включает замкнутые координатные линии. Из-за этого при переходе к декартовой системе координат в описании поверхностей появляются тригонометрические функции параметров. Вычисление значений тригонометрических функций для компьютера является времязатратной операцией, поэтому в графике от них избавляются, например, используя работу с приращениями [3]. Дело в том, что текущие точки поверхности вычисляются в некотором порядке, и две последовательно обработанные точки оказываются на поверхности соседними. Это позволяет представить координаты следующей точки как функции координат предыдущей. В частности, очередное значение тригонометрических функций (для угла φ_{i+1}) можно найти, используя аргумент предыдущего значения φ_i и его приращение (шаг) $\Delta\varphi$:

$$\begin{aligned}\sin \varphi_{i+1} &= \sin(\varphi_i + \Delta\varphi) = \sin \varphi_i \cos \Delta\varphi + \cos \varphi_i \sin \Delta\varphi, \\ \cos \varphi_{i+1} &= \cos(\varphi_i + \Delta\varphi) = \cos \varphi_i \cos \Delta\varphi - \sin \varphi_i \sin \Delta\varphi.\end{aligned}$$

При выбранном шаге $\Delta\varphi$ значения $\sin \Delta\varphi$, $\cos \Delta\varphi$ являются константами, и вычисление очередного значения тригонометрической функции сводится к двум операциям умножения этих констант на уже известные предыдущие значения $\sin \varphi_i$, $\cos \varphi_i$ и суммированию полученных произведений. Начальные значения тригонометрических функций очевидны: $\sin \varphi_0=0$, $\cos \varphi_0=1$.

Если квадрики описаны в параметрической форме, то усложняется задача определения нормалей к поверхностям. Понятно, что для определения освещенности криволинейной поверхности требуется определить направление нормали в каждой ее точке. Компоненты вектора нормали, проведенной в i -ой точке к параметрически заданной поверхности, являются, в общем случае, дробно-рациональными функциями и в СКН описываются выражениями

$$n_{xi} = \frac{\begin{vmatrix} \left(\frac{\partial y}{\partial u}\right)_i & \left(\frac{\partial z}{\partial u}\right)_i \\ \left(\frac{\partial y}{\partial v}\right)_i & \left(\frac{\partial z}{\partial v}\right)_i \end{vmatrix}}{\begin{vmatrix} \left(\frac{\partial y}{\partial u}\right)_i & \left(\frac{\partial z}{\partial u}\right)_i \\ \left(\frac{\partial y}{\partial v}\right)_i & \left(\frac{\partial z}{\partial v}\right)_i \end{vmatrix}}, \quad n_{yi} = \frac{\begin{vmatrix} \left(\frac{\partial z}{\partial u}\right)_i & \left(\frac{\partial x}{\partial u}\right)_i \\ \left(\frac{\partial z}{\partial v}\right)_i & \left(\frac{\partial x}{\partial v}\right)_i \end{vmatrix}}{\begin{vmatrix} \left(\frac{\partial y}{\partial u}\right)_i & \left(\frac{\partial z}{\partial u}\right)_i \\ \left(\frac{\partial y}{\partial v}\right)_i & \left(\frac{\partial z}{\partial v}\right)_i \end{vmatrix}}, \quad n_{zi} = \frac{\begin{vmatrix} \left(\frac{\partial x}{\partial u}\right)_i & \left(\frac{\partial y}{\partial u}\right)_i \\ \left(\frac{\partial x}{\partial v}\right)_i & \left(\frac{\partial y}{\partial v}\right)_i \end{vmatrix}}{\begin{vmatrix} \left(\frac{\partial y}{\partial u}\right)_i & \left(\frac{\partial z}{\partial u}\right)_i \\ \left(\frac{\partial y}{\partial v}\right)_i & \left(\frac{\partial z}{\partial v}\right)_i \end{vmatrix}}. \quad (3.1)$$

где u и v – параметры, например, φ и θ .

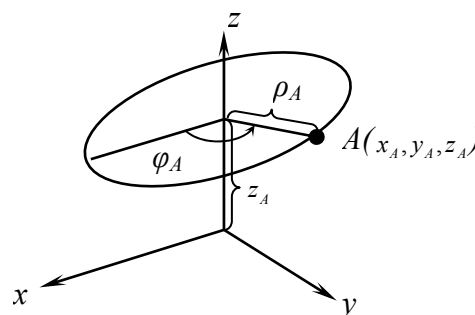
При разворачивании поверхности по координатным линиям u или v две частные производные в каждом определителе постоянны, однако вычисление компонентов n_x , n_y , n_z в целом требует выполнения нескольких операций умножения для каждой точки поверхности, что для режима реального времени является недопустимо затратным. Задачу можно упростить, если иметь в виду, что направление нормали зависит от вида самой поверхности, а не от формы ее математического представления. Тогда для нахождения координат нормали можно использовать общую форму описания квадрики:

$$n_{xi} = \frac{\partial F(x, y, z)}{\partial x}, \quad n_{yi} = \frac{\partial F(x, y, z)}{\partial y}, \quad n_{zi} = \frac{\partial F(x, y, z)}{\partial z}.$$

Например, для сферы координаты нормали после пропорционального деления на два, что не изменяет направление вектора, принимают вид: $n_x=x$, $n_y=y$, $n_z=z$.

Характерной особенностью рассматриваемых квадрик является наличие замкнутых сечений хотя бы одной координатной плоскостью (или плоскостью, параллельной ей). Эта особенность позволяет привести описание квадрик к одной форме и выполнять их развертывание по замкнутым координатным линиям, в общем случае – эллипсам. Под термином "развертывание" будем понимать последовательный расчет точек поверхности в процессе сканирования аргументов координатной системы. Во избежание описания многозначными функциями, квадрики следует описывать параметрически связанными уравнениями. В качестве параметров описания удобно использовать координаты z и φ цилиндрической системы координат. Пусть ось z цилиндрической системы координат совпадает с осью аппликат декартовой системы координат. Тогда координата z точки в этих двух системах координат будет одной и той же, и для ее обозначения можно использовать один и тот же символ. На рисунке 3.3 показана текущая точка A , которая лежит на эллипсе сечения квадрики плоскостью, перпендикулярной оси z . Она имеет цилиндрические координаты ρ_A , φ_A , z_A , причем координата ρ_A изменяется по некоторому закону и зависит от высоты сечения.

Рисунок. 3.3 – Сечение квадрики в цилиндрических и декартовых координатах



В общем случае параметрическое описание квадрик имеет вид

$$\left. \begin{aligned} x &= f_x \cos \varphi, \\ y &= f_y \sin \varphi, \\ z &= z, \quad z = -z_{\max}, \dots, z_{\max}, \quad \varphi = 0, \dots, 2\pi, \end{aligned} \right\} \quad (3.2)$$

где f_x , f_y – функциональные (зависящие от z) коэффициенты, определяющие геометрические параметры поверхности.

Описание (3.2) представляет поверхность квадрики в виде набора эллиптических сечений. Например, эллипсоид можно образно представить в виде дыни,

состоящей из эллиптических плоских ломтиков, которые нарезаны перпендикулярно одной из осей дыни. Коэффициенты f_x , f_y для выбранных квадрик приведены в таблице 3.1, где, кроме того, показаны координаты вектора нормали $\bar{N} (n_x, n_y, n_z)$.

Таблица 3.1 – Параметрическое описание квадрик

Параметр	Поверхность второго порядка				
	Эллипсоид	Эллиптический параболоид	Конус	Эллиптический цилиндр	Однополостной гиперболоид
f_x	$a\sqrt{1-(z/c)^2}$	$a\sqrt{z}$	$(a/c)z$	a	$a\sqrt{1+(z/c)^2}$
f_y	$b\sqrt{1-(z/c)^2}$	$b\sqrt{z}$	$(b/c)z$	b	$b\sqrt{1+(z/c)^2}$
n_x	x/a^2	x/a^2	x/a^2	x/a^2	x/a^2
n_y	y/b^2	y/b^2	y/b^2	y/b^2	y/b^2
n_z	z/c^2	$-1/2$	$-z/c^2$	0	$-z/c^2$

В таблице 3.1 приняты следующие обозначения: a и b – полуоси эллипсов-сечений, перпендикулярных оси z и проведенных для эллипсоида и однополостного гиперболоида на уровне $z=0$, для эллиптического параболоида – на уровне $z=1$, для конуса – на уровне $z=c$, а для эллиптического цилиндра – на любом уровне; c – третья полуось эллипсоида, мнимая полуось однополостного гиперболоида и величина, влияющая на угол при вершине конуса.

Недостатком примитивов-квадрик является большое количество операций с точками, выполняемых в пространстве наблюдателя. Например, для моделирования освещенности поверхности или выявления ее нелицевых точек необходимо анализировать положение нормали в каждой точке примитива, при наложении текстуры для каждой точки поверхности нужно вычислить текстурные координаты, по точкам выполняется и перспективное проецирование поверхности на плоскость экрана. Для снижения вычислительных затрат

поверхности второго порядка на определенном этапе преобразований подвергают триангуляции и далее работают с плоскими треугольниками по сравнительно простым алгоритмам.

3.3 Бикубические сплайны

Часто отображаемые объекты, особенно природные, имеют довольно сложную форму, не допускающую универсального аналитического описания в целом. Их форма задается набором характерных (опорных) точек, принадлежащих поверхности объекта. Опорные точки получаются в результате замеров на реальных объектах, их сканирования с помощью 3D-сканеров или назначаются разработчиками. В качестве примера можно назвать составленную геодезистами карту высот участка земной поверхности. В процессе геометрического моделирования исходная поверхность должна быть восстановлена с заданной точностью. Она должна проходить как можно ближе к опорным точкам, а лучше – через них. При этом должен быть сохранен характер (топология) исходной поверхности. Самый простой подход – соединить опорные точки участками плоскости, то есть применить полигональную модель. Однако для достижения реалистичности отображения объекта его полигональная модель должна насчитывать тысячи и десятки тысяч полигонов, что повышает требования к объему памяти и производительности графической системы. Не приносит успеха и применение квадрик, так как в этом случае возникает проблема их гладкой стыковки в единую поверхность. Поверхности неаналитических форм представляют кусочно-полиномиальными функциями – сплайнами.

Слово «сплайн» (spline) пришло из кораблестроения. Так в свое время в Англии называли длинную и тонкую металлическую линейку. Ее прижимали к ребрам (опорным точкам) будущего судна и, благодаря ее упругости, получали обводы бортов. В геометрическом моделировании сплайны – это степенные функции одного или двух переменных, графическими образами которых являются кривые линии или криволинейные поверхности. Они служат, в частности, для

решения задачи интерполяции, то есть нахождения промежуточных точек кривой линии или поверхности, заданной опорными точками. Уравнения сплайнов имеют, обычно, степень не выше третьей, так как именно такая степень является минимально необходимой для гладкой стыковки криволинейных участков. Покажем это на примере сплайн-функции одной переменной.

Пусть две точки $P1$ и $P2$, показанные на рисунке 3.4, нужно соединить кривой S таким образом, чтобы она проходила через эти точки и гладко сопрягалась с соседними участками кривой. Соседние участки на рисунке показаны толстыми сплошными линиями, а желаемый вид кривой S – штриховой линией.

Очевидно, для решения задачи нужно наложить на кривую S четыре ограничения:

1) и 2) – кривая проходит через точки $P1$ и $P2$, тогда составная кривая не будет иметь разрывов первого рода;

3) и 4) – сопряжение кривой S с соседними участками в точках $P1$ и $P2$ является плавным (гладким), тогда составная кривая не будет иметь разрывов второго рода.

Для этого требуется иметь математическое описание кривой S в виде многочлена, имеющего не менее четырех коэффициентов. Самым простым из таких многочленов является многочлен третьей степени, который, в общем случае, имеет вид

$$y = ax^3 + bx^2 + cx + d,$$

включающий как раз четыре коэффициента формы a, b, c и d . Тогда наложение четырех упомянутых ограничений дает систему четырех уравнений с четырьмя неизвестными:

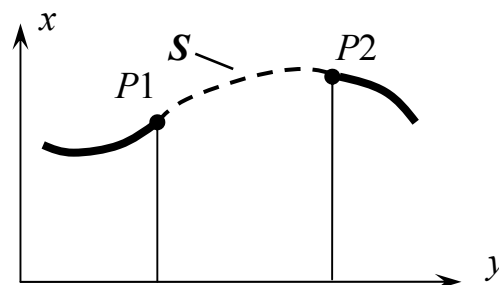


Рисунок 3.4 – Сплайн-интерполяция по двум точкам

- 1) прохождение через $P1$ – $y_1 = ax_1^3 + bx_1^2 + cx_1 + d$,
- 2) прохождение через $P2$ – $y_2 = ax_2^3 + bx_2^2 + cx_2 + d$,
- 3) равенство первых производных соседних сплайнов в точке $P1$ –

$$\dot{y}_1 = 3ax_1^2 + 2bx_1 + c,$$

- 4) равенство первых производных соседних сплайнов в точке $P2$ –

$$\dot{y}_2 = 3ax_2^2 + 2bx_2 + c.$$

Сплайновая поверхность, в отличие от кривой, должна проходить через четыре точки, являющиеся для нее угловыми. Поверхность можно представить как результат движения кубической кривой параллельно самой себе. При этом конечные точки этой кривой в процессе движения скользят по двум другим (боковым) кубическим кривым. В результате получается поверхность, которая описывается бикубическим степенным многочленом. Каждое слагаемое многочлена включает два аргумента, имеющих в различных сочетаниях степени от 0 до 3. Существует много разновидностей сплайновых поверхностей, обладающих различными свойствами и формируемых с использованием различных условий и геометрических параметров [3].

Для описания сплайнов можно использовать явную, неявную и параметрическую формы. В компьютерной графике обычно используют параметрическое описание сплайнов. Явная форма описания в декартовой системе координат по ряду причин используется редко. Во-первых, вид описания поверхности в явной форме зависит от выбранного положения системы координат. Во-вторых, некоторые участки поверхности могут иметь вертикальные касательные векторы, то есть, стремящиеся к бесконечности производные. В этом случае невозможно задать условия стыковки отсеков поверхности. Наконец, в-третьих, параметрическое представление, в отличие от явной формы, описывает естественный последовательный обход участков поверхности в процессе ее развертывания во времени, что было пояснено на примере квадрики. В общем случае, отсек сплайновой поверхности описывается бикубическими выражениями вида

$$\begin{aligned}
x(u, v) = & (C_{X00} + C_{X01}u + C_{X02}u^2 + C_{X03}u^3)v^0 + \\
& + (C_{X10} + C_{X11}u + C_{X12}u^2 + C_{X13}u^3)v^1 + \\
& + (C_{X20} + C_{X21}u + C_{X22}u^2 + C_{X23}u^3)v^2 + \\
& + (C_{X30} + C_{X31}u + C_{X32}u^2 + C_{X33}u^3)v^3 = \sum_{i=0}^3 \sum_{j=0}^3 C_{Xij} u^j v^i,
\end{aligned} \tag{3.3}$$

$$u = 0, \dots, 1, \quad v = 0, \dots, 1,$$

где u, v – параметры описания (аргументы сплайн-функции);

C_{X00}, \dots, C_{X33} – коэффициенты формы, определяющие геометрические характеристики поверхности.

Аналогичный вид имеют выражения $y(u, v)$ и $z(u, v)$, включающие наборы коэффициентов C_{Y00}, \dots, C_{Y33} и C_{Z00}, \dots, C_{Z33} , соответственно.

Смысл выражений (3.3) следующий. Аргументы u, v представляют собой координаты криволинейной координатной системы, расположенной на поверхности сплайна. В ней каждая точка поверхности задается парой числовых значений. Для отображения поверхности координаты ее точек с помощью (3.3) переводятся в декартову систему координат, в которой расположена экранная плоскость. Соответствие координат текущей точки CP сплайновой поверхности в двух упомянутых координатных системах показано на рисунке 3.5.

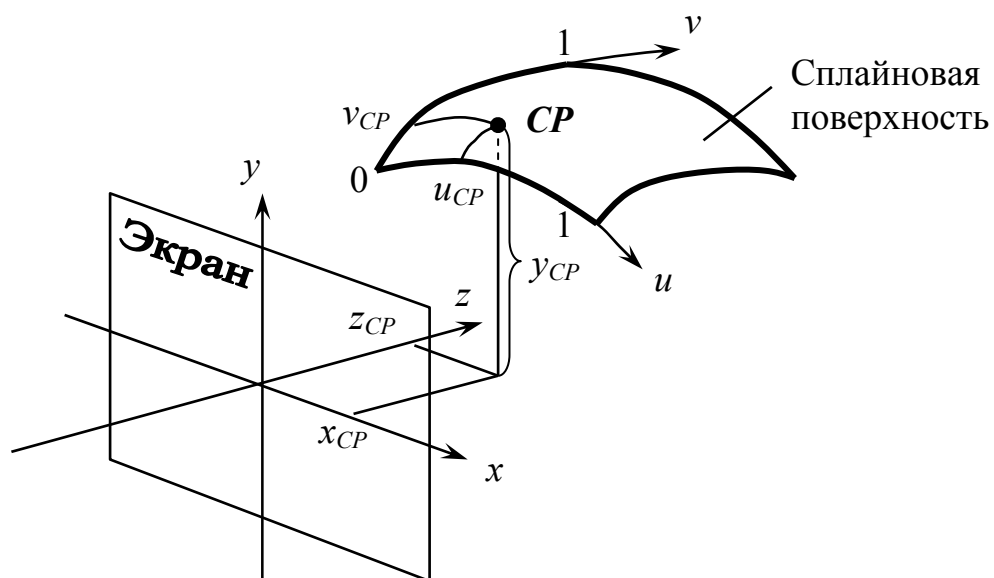


Рисунок 3.5 – Соответствие между параметрическими и декартовыми координатами

Коэффициенты многочленов отыскиваются при наложении ограничений на форму отсека поверхности. В зависимости от выбора ограничений поверхность получает ту или иную форму описания. Например, ограничениями для поверхности Кунса (частный случай – поверхность Эрмита, поверхность Фергюсона) являются условия ее прохождения через заданные угловые точки, а также соответствие заданным значениям частных производных (то есть наклонов) в угловых точках поверхности и смешанных частных производных (то есть кручений) в этих точках [1,3]. Использование таких ограничений является геометрически понятным, но сложно реализуемым. В компьютерных системах геометрического моделирования в качестве ограничений обычно используется повторение сплайном формы некоторой многогранной опорной поверхности (характеристического многогранника), заданной шестнадцатью опорными точками. Поверхность должна проходить вблизи опорных точек или через некоторые из них, и изменение их координат должно приводить к изменению формы поверхности. Поэтому описание отсека поверхности представляют не в форме (3.3), а в другом виде, выражая коэффициенты многочленов через координаты опорных точек. Описание в координатной форме имеет вид

$$\begin{aligned} x(u, v) &= \sum_{i=0}^3 \sum_{j=0}^3 f_i(v) f_j(u) P_{Xij}, \\ y(u, v) &= \sum_{i=0}^3 \sum_{j=0}^3 f_i(v) f_j(u) P_{Yij}, \\ z(u, v) &= \sum_{i=0}^3 \sum_{j=0}^3 f_i(v) f_j(u) P_{Zij}, \quad u = 0..1, v = 0..1, \end{aligned} \quad (3.4)$$

где P_x, P_y, P_z – массивы x, y и z -координат опорных точек;

$f_i(u), f_j(v)$ – функциональные коэффициенты, имеющие свой вид для каждой разновидности сплайн-функций.

Значения функций $f_i(u), f_j(v)$ выступают как весовые коэффициенты координат опорных точек, поэтому эти функции называют весовыми или смешивающими.

В компьютерной графике обычно применяется описание сплайна в матричной форме, которое выглядит следующим образом:

$$\begin{aligned}
x(u, v) &= U \cdot M \cdot P_x \cdot M^T \cdot V^T, \\
y(u, v) &= U \cdot M \cdot P_y \cdot M^T \cdot V^T, \\
z(u, v) &= U \cdot M \cdot P_z \cdot M^T \cdot V^T,
\end{aligned} \tag{3.5}$$

где U, V – векторы степеней параметров u и v :

$$U = \begin{vmatrix} u^3 & u^2 & u & 1 \end{vmatrix}, \quad V = \begin{vmatrix} v^3 & v^2 & v & 1 \end{vmatrix};$$

P_x, P_y, P_z – геометрические матрицы, содержащие координаты x, y и z опорных точек, например, для нумерации опорных точек, принятой на рисунке 3.6,

$$P_x = \begin{vmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{vmatrix};$$

M – базисная матрица поверхности, которая содержит числовые коэффициенты, определяющие своеобразие поверхности.

Сплайны характеризуются рядом полезных свойств. Было уже упомянуто, что форма сплайнового отсека следует за формой характеристического многогранника. Если все его опорные точки лежат в одной плоскости, то все текущие точки сплайна тоже лежат в этой плоскости. Характеристический многогранник является описанным вокруг сплайновой поверхности, следовательно, попадание этой поверхности в некоторый объем (например, объем видимости наблюдателя) легко проверяется по шестнадцати точкам. Кроме того, сплайны инвариантны по отношению к аффинным преобразованиям. Это означает, что при необходимости сдвига, поворота, масштабирования и отражения сплайна не нужно подвергать этим преобразованиям все текущие точки отсека. Достаточно выполнить преобразования только над опорными точками, а потом просто применить алгоритм развертывания сплайна (выражения (3.5)) на этих преобразованных опорных точках.

В геометрическом моделировании часто применяется бикубическая поверхность Безье. Ограничениями при построении этой поверхности является ее прохождение через угловые точки характеристического многогранника и заданные на его границах наклоны касательных в направлениях u, v . На рисунке

3.6 показана бикубическая поверхность Безье и ее характеристический многогранник, вершинами которого являются 16 опорных точек P_{00}, \dots, P_{33} .

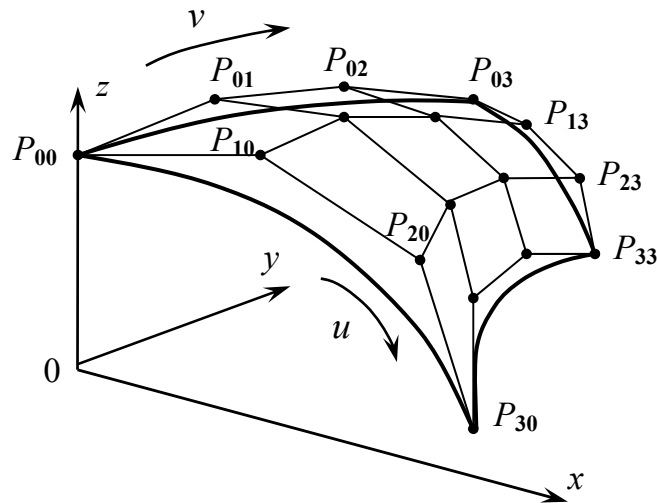


Рисунок 3.6 – Сплайн Безье с характеристическим многогранником

Поверхность размещена в своей локальной декартовой системе координат. Индексы обозначений опорных точек привязаны к параметрам u, v (P_{uv}) и нарастают в направлении нарастания этих параметров (они показаны на рисунке стрелками). Предполагается, что поверхность сплайна разворачивается по линиям $v=\text{const}$, то есть сначала вычисляются текущие точки, лежащие на линии $v=0$, затем – на линии $v=\Delta v$, далее – на линии $v=2\Delta v$ и так далее, где Δv – шаг по параметру v . Угловые точки используются для привязки примитива к моделируемой поверхности, а промежуточные – для «изгибания» отсека в различных направлениях.

Для поверхности Безье функциональные коэффициенты в (3.4) определяются выражениями

$$f_i(u) = C_m^i u^i (1-u)^{m-i}, \quad f_j(v) = C_n^j v^j (1-v)^{n-j},$$

где C_m^i, C_n^j – биномиальные коэффициенты,

а базисная матрица в системе (3.5) выглядит следующим образом:

$$M = \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}.$$

Можно убедиться в том, что сплайн Безье проходит через угловые опорные точки, например, через точку P_{00} . Покажем это на примере координаты x . Для этого подставим в первое выражение (3.5) матрицу M и значения параметров $u=v=0$, а затем последовательно выполним матричные операции:

$$\begin{aligned}
 x(u, v) &= \begin{vmatrix} 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \cdot \begin{vmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{vmatrix} \cdot \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \cdot \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \end{vmatrix} = \\
 &= \begin{vmatrix} 1 & 0 & 0 & 0 \end{vmatrix} \cdot \begin{vmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{vmatrix} \cdot \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \cdot \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \end{vmatrix} = \begin{vmatrix} x_{00} & x_{01} & x_{02} & x_{03} \end{vmatrix} \cdot \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} \cdot \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \end{vmatrix} = \\
 &= \begin{vmatrix} (-x_{00} + 3x_{01} - 3x_{02} + x_{03}) & (3x_{00} - 6x_{01} + 3x_{02}) & (-3x_{00} + 3x_{01}) & x_{00} \end{vmatrix} \cdot \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \end{vmatrix} = x_{00}.
 \end{aligned}$$

Основной задачей при использовании сплайновых примитивов в геометрическом моделировании пространственных объектов является расстановка опорных точек. В случае сплайна Безье четыре угловые точки примитива находятся легко: они принадлежат поверхности объекта. Остальные точки должны выбираться таким образом, чтобы обеспечить гладкую стыковку примитива с соседними. Нетрудно сообразить, что таких соседних примитивов будет восемь: четыре будут соприкасаться с данным примитивом ребрами и четыре – угловыми точками. Для каждой угловой точки примитива нужно выполнить три условия гладкости: по направлению координаты u , по направлению координаты v и по диагональному направлению. Для этого крайние опорные точки соседних примитивов должны лежать на одной прямой в каждом из названных направлений, что иллюстрируется рисунком 3.7.

На рисунке толстыми линиями показаны общие границы четырех сплайновых примитивов $S1, \dots, S4$. Их углы сходятся в одной точке, которая в каждом примитиве имеет свое обозначение в соответствии с обозначениями на рисунке 3.6 (первый нижний индекс обозначения показывает номер примитива).

Штрихами показаны прямые линии, на которых лежат крайние опорные точки соседних примитивов.

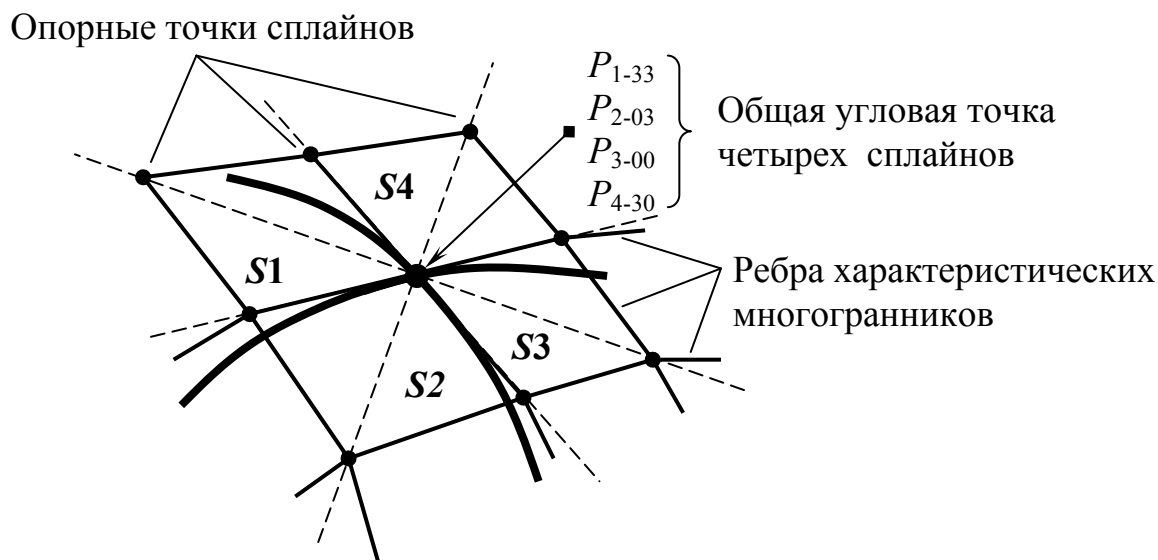


Рисунок 3.7 – Гладкая стыковка сплайновых примитивов

Математическими условиями гладкой стыковки будет равенство частных производных, взятых в соответствующих направлениях для сплайн-функций, которые описывают соседние примитивы. Для диагонального направления берется смешанная производная. При построении математической модели сложной поверхности нужно знать, какими будут эти частные производные, а в распоряжении проектировщика имеется только сама эта поверхность или ее характерные точки. Чтобы упростить процесс описания поверхности, прибегают к диалоговым системам геометрического моделирования. Они располагают готовыми сплайновыми примитивами, форму которых можно изменять в режиме реального времени. Это делается путем перемещения («перетаскивания») опорных точек. Угловые точки привязываются к моделируемой поверхности, а положение промежуточных точек интерактивно задается таким образом, чтобы примитив получил желаемую конфигурацию. Одной из известных систем геометрического моделирования является 3DStudioMAX.

Следует обратить внимание на то, что при изменении аргументов-параметров u, v в диапазоне $(0,1)$ сплайн Безье занимает пространство в пределах

всего характеристического многогранника и проходит через его угловые точки. В приложении А приведена иллюстрирующая программа моделирования сплайна Безье, написанная в среде математического пакета MathCAD.

Кроме сплайнов Безье, в компьютерной графике широко применяются базовые сплайны, или *B*-сплайны («би-сплайны»). Их математическая модель в матричной форме также описывается выражениями (3.5). Форма *B*-сплайна однозначно задается координатами опорных точек, однако, отсек, в общем случае, только приближается к опорным точкам и может не проходить ни через одну из них. Для *B*-сплайновой поверхности третьей степени функциональные коэффициенты в выражениях (3.4) определяются формулами [3]

$$\begin{aligned} f_0(u) &= \frac{1}{6}(1-u)^3, & f_1(u) &= \frac{1}{6}(3u^3 - 6u^2 + 4), \\ f_2(u) &= \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1), & f_3(u) &= \frac{1}{6}u^3, \end{aligned} \quad (3.6)$$

(формулы для многочленов $f_j(v)$ отличаются от приведенных только тем, что в них всюду вместо параметра u стоит параметр v), а базисная матрица M имеет вид

$$M = \frac{1}{6} \begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{vmatrix}.$$

Подставив матрицу M в выражения (3.5) и проделав матричные операции, как это было сделано для сплайна Безье, можно убедиться, что *B*-сплайн не проходит через опорные точки. Например, текущая точка с параметрическими координатами $u=v=0$ получает следующее значение координаты x :

$$x(u, v) = \frac{1}{36}(x_{00} + 4x_{10} + x_{20} + 4x_{01} + 16x_{11} + 4x_{21} + x_{02} + 4x_{12} + x_{22}). \quad (3.7)$$

Оно показывает, что координата текущей точки не совпадает с координатой ни одной опорной точки, а является суммой «вкладов», которые вносят несколько опорных точек. Наибольший вклад вносит опорная точка P_{11} , следовательно, именно к ней и будет ближе текущая точка при нулевых значениях аргументов u и v .

Применение B -сплайнов упрощает процесс моделирования сложных поверхностей. Любые 16 характерных точек поверхности, образующие четырехугольник 4×4 точки, могут быть приняты за опорные точки B -сплайнового примитива. При этом сформированный отсек займет не весь характеристический многогранник, а расположится вблизи четырех средних точек $P_{11}, P_{12}, P_{21}, P_{22}$. Только что было показано аналитически, что при нулевых значениях параметров u и v текущая точка приближается не к угловой опорной точке, а к одной из центральных – к точке P_{11} . В приложении Б приведена MathCAD-программа моделирования B -сплайнового отсека на тех же опорных точках, что и сплайна Безье в приложении А. По цифровым значениям координат видно, что при изменении параметров u, v в диапазоне $(0,1)$ B -сплайн «привязывается» к центральной части характеристического многогранника и занимает в пространстве примерно в девять раз меньше места, чем сплайн Безье. Следовательно, для моделирования одинаковых участков поверхности с помощью B -сплайновых примитивов требуется в девять раз больше вычислений, чем в случае сплайнов Безье. Этот недостаток B -сплайнов компенсируется их повышенной гладкостью, достигаемой без введения дополнительных опорных точек. Отсюда вытекает простая методика моделирования пространственных поверхностей сложной формы:

- 1) поверхность представляется набором принадлежащих ей характерных точек. Выбирается расположение координатных осей u, v . Характерные точки принимаются за опорные точки B -сплайновых примитивов;
- 2) образуется «окно» включающее 16 опорных точек (по 4 вдоль каждой координатной оси). По выражениям (3.5) происходит вычисление координат текущих точек примитива;
- 3) по окончании развертывания очередного примитива окно перемещается на один ряд опорных точек по оси u или v . В результате перемещения в окно входят 12 прежних опорных точек и 4 новые точки. Для этого набора опорных точек по выражениям (3.5) вычисляются координаты текущих точек;

4) пункт 3) выполняется до тех пор, пока не будут использованы все опорные точки. После этого выполняется визуальный контроль смоделированной поверхности. Для коррекции рельефа по результатам контроля изменяется положение опорных точек.

Привязка *B*-сплайнового отсека к центральной части характеристического многогранника приводит к тому, что смоделированная поверхность не проходит через крайние (граничные) характерные точки исходной поверхности. Чтобы получить отсеки, примыкающие к границам поверхности нужно иметь дополнительные ряды опорных точек, о расположении которых нет никакой информации. Для решения задачи прибегают к так называемым кратным граничным опорным точкам. Кратными называются опорные точки, имеющие различные обозначения, но одинаковые координаты. Рассмотрим их применение на примере одного сплайнового отсека.

Кратные опорные точки образуются повторением координат граничных опорных точек и присвоением новым точкам уникальных обозначений. Каждая граничная опорная точка получает одну дополнительную, кратную себе, а четыре угловые точки получают по три дополнительные опорные точки каждая. Всего при кратности, равной двум, поверхность, имеющая $m \times n$ опорных точек, получает $(2m+2n+4)$ дополнительных опорных точек. Схематически это показано на рисунке 3.8 (для одного отсека *B*-сплайна).

Маленькими кружками обозначены 16 опорных точек *B*-сплайна (P_{00} , P_{30} , P_{03} , P_{33} – угловые точки). Сплошными линиями показаны участки характеристического многогранника, соответствующие девяти возможным отсекам поверхности. При использовании имеющихся (исходных) опорных точек будет развернут центральный отсек, расположенный вблизи заштрихованного участка. Отсеки, приближенные к граничным участкам, например, к участку с точечной заливкой, развернуты быть не могут. Большими кружками обозначены дополнительные опорные точки. Стрелками показано, какие исходные опорные точки дают им свои координаты. Теперь опорных точек формально стало больше. Границы новой поверхности обозначены штриховой линией. Она проходит через

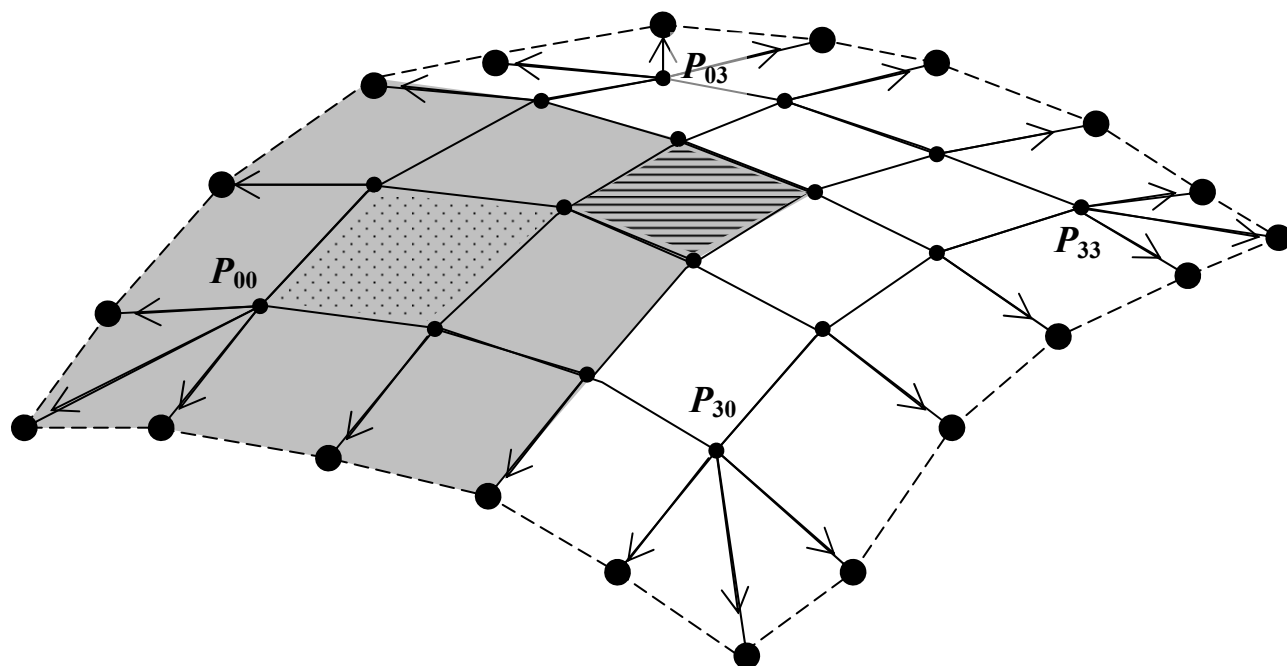


Рисунок 3.8 – Образование дополнительных опорных точек второй кратности

новые опорные точки, но они имеют те же координаты, что и старые граничные опорные точки, следовательно, геометрические границы сплайновой поверхности не изменились, изменилось лишь их формализованное описание. Благодаря дополнительным опорным точкам граничные отсеки исходной поверхности становятся центрами новых сплайновых примитивов и могут быть развернуты. Например, участок, отмеченный точечной заливкой, стал центром характеристического многоугольника, выделенного серым цветом. Можно показать, что угловая точка отсека, привязанного к этому участку, теперь стала ближе к опорной точке P_{00} . Координата x угловой точки описывается выражением (3.7). В силу кратности опорных точек $x_{00}=x_{10}=x_{01}=x_{11}$, и опорная точка P_{00} получает самый большой вес, равный $\frac{25}{36}$, а до введения кратных точек она имела вес, равный $\frac{1}{36}$.

Кратность опорных точек, равная двум, приближает точки B -сплайна к граничным точкам моделируемой поверхности, но не дает их совпадения. Для совпадения необходима кратность, равная трем. Она достигается повторным добавлением кратных точек к граничным опорным точкам. Принцип добавления

тот же, что и показанный на рисунке 3.8. Нетрудно видеть, что количество добавляемых во второй раз опорных точек равно $(2m+2n+12)$. В результате к участкам характеристического многогранника, показанным на рисунке 3.8, добавляется еще по одному ряду с каждой стороны поверхности (к каждой ее границе). В то же время геометрически границы поверхности не изменяются. При использовании опорных точек третьей кратности граничные отсеки поверхности представляют собой узкие полоски, проходящие через граничные точки. Наконец, образовав граничные опорные точки с кратностью, равной четырем, получим граничные отсеки, вырожденные в кривые линии. Эти кривые линии будут строго проходить через граничные точки характеристического многогранника.

В приложении В приведены изображения, полученные MathCAD-программой и иллюстрирующие применение кратных опорных точек. В качестве исходного набора опорных точек использованы те же точки, что и в приложении А. Обратив внимание на численные значения координат на полученных изображениях, можно судить о расположении отсеков в пространстве и о влиянии на это кратности опорных точек.

Таким образом, применение кратных опорных точек улучшает качество моделирования сложных поверхностей, хотя и значительно увеличивает расход вычислительных ресурсов графической системы.

Для моделирования освещения сплайновых поверхностей необходимо знать направления нормалей, проведенных через их точки. Как уже отмечалось, координаты нормалей к параметрически заданным поверхностям описываются выражениями (3.1), которые довольно сложны. В связи с этим, после геометрических преобразований сплайновые поверхности, как правило, подвергаются тесселяции и далее обрабатываются в полигональной форме.

Широкими изобразительными возможностями обладают рациональные бикубические сплайны. В компьютерной графике обычно применяют рациональные *B*-сплайны на неравномерной сетке (Non-Uniform Rational *B*-Splines – NURBS). В их описание входят числовые параметры формы (весовые коэффициенты), позволяющие управлять формой поверхности без изменения

координат опорных точек. Описание рациональной бикубической сплайновой поверхности имеет следующий вид

$$x(u, v) = \frac{\sum_{i=0}^3 \sum_{j=0}^3 w_{ij} f_i(u) f_j(v) P_{xij}}{\sum_{i=0}^3 \sum_{j=0}^3 w_{ij} f_i(u) f_j(v)}, \quad (3.8)$$

где w_{ij} – весовые коэффициенты, в общем случае свои для каждой опорной точки.

Координаты $y(u, v)$ и $z(u, v)$ описываются аналогично (с использованием наборов координат P_y и P_z). Функциональные коэффициенты $f_i(u), f_j(v)$ могут быть такими же, как и у предыдущих сплайн-функций. Своеобразием рациональных B -сплайнов является их инвариантность относительно перспективного проецирования. Это означает, что перспективное проецирование на плоскость экрана всех точек сплайна может быть заменено разворачиванием сплайна по выражениям (3.8), в которых координаты опорных точек (P_x, P_y, P_z) пересчитаны по закону перспективы. Нерациональные сплайны таким свойством не обладают.

Описание координаты текущей точки рационального сплайна в матричной форме может выглядеть следующим образом:

$$x(u, v) = \frac{U \cdot M \cdot WX \cdot M^T \cdot V^T}{U \cdot M \cdot W \cdot M^T \cdot V^T},$$

$$WX = \begin{pmatrix} (w_{00} \cdot x_{00}) & (w_{01} \cdot x_{01}) & (w_{02} \cdot x_{02}) & (w_{03} \cdot x_{03}) \\ (w_{10} \cdot x_{10}) & (w_{11} \cdot x_{11}) & (w_{12} \cdot x_{12}) & (w_{13} \cdot x_{13}) \\ (w_{20} \cdot x_{20}) & (w_{21} \cdot x_{21}) & (w_{22} \cdot x_{22}) & (w_{23} \cdot x_{23}) \\ (w_{30} \cdot x_{30}) & (w_{31} \cdot x_{31}) & (w_{32} \cdot x_{32}) & (w_{33} \cdot x_{33}) \end{pmatrix}, \quad W = \begin{pmatrix} w_{00} & w_{01} & w_{02} & w_{03} \\ w_{10} & w_{11} & w_{12} & w_{13} \\ w_{20} & w_{21} & w_{22} & w_{23} \\ w_{30} & w_{31} & w_{32} & w_{33} \end{pmatrix},$$

где WX – матрица координат опорных точек с соответствующими коэффициентами формы;

W – матрица коэффициентов формы сплайна.

Вид приведенных выражений показывает, что моделирование поверхностей на основе рациональных сплайнов требует больших вычислительных затрат. Макрокоманды рисования NURBS есть в современных графических библиотеках.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Фокс А., Пратт М. Вычислительная геометрия. Применение в проектировании и на производстве: Пер. с англ. – М.: Мир, 1982. – 304 с.
2. Иванов В.П., Батраков А.С. Трехмерная компьютерная графика /Под ред. Г.М. Полищука. – М.: Радио и связь, 1995. – 224 с.
3. Роджерс Д., Адамс Дж. Математические основы машинной графики: Пер. с англ. – М.: Мир, 2001. – 604 с.
4. Никулин Е.А. Компьютерная геометрия и алгоритмы машинной графики. – СПб.: БХВ-Петербург, 2003. – 560 с.
5. Голованов Н.Н. Геометрическое моделирование. – М.: Издательство Физико-математической литературы, 2002. – 472 с.
6. Роджерс Д. Алгоритмические основы машинной графики: Пер. с англ. – М.: Мир, 1989. – 512 с.
7. Эйнджел Э. Интерактивная компьютерная графика. Вводный курс на базе OpenGL, 2 изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 592 с.
8. Блинова Т.А., Порев В.Н. Компьютерная графика /Под ред. В.Н.Порева – К.: Издательство Юниор, 2005. – 520 с.

ПРИЛОЖЕНИЯ

Приложение А. Моделирование сплайна Безье по его параметрическому описанию

Исходными данными для моделирования являются координаты шестнадцати опорных точек и геометрическая (базисная) матрица Безье. Нумерация опорных точек соответствует рисунку 3.6. Для примера возьмем координаты опорных точек, приведенные в таблице А1.

Таблица А1 – Координаты опорных точек сплайнового примитива

Точка	P_{00}	P_{10}	P_{20}	P_{30}	P_{01}	P_{11}	P_{21}	P_{31}	P_{02}	P_{12}	P_{22}	P_{32}	P_{03}	P_{13}	P_{23}	P_{33}
x_{uv}	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
y_{uv}	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
z_{uv}	0	1	-1	0.5	2	1.5	0	0.5	2	1.8	1.5	1.8	1.5	2	1.5	1.5

Координаты текущих точек сплайна Безье обозначим xx , yy , zz . Они вычисляются по параметрическим выражениям (3.5)

$$xx = U \cdot M \cdot X \cdot M^T \cdot V^T,$$

$$yy = U \cdot M \cdot Y \cdot M^T \cdot V^T,$$

$$zz = U \cdot M \cdot Z \cdot M^T \cdot V^T.$$

Координаты опорных точек представляются в форме матриц

$$X := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \end{bmatrix}$$

$$Y := \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

$$Z := \begin{bmatrix} 0 & 2 & 2 & 1.5 \\ 1 & 1.5 & 1.8 & 2 \\ -1 & 0 & 1.5 & 1.5 \\ 0.5 & 0.5 & 1.8 & 1.5 \end{bmatrix}$$

Геометрическая матрица Безье

$$\mathbf{M} := \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Для графических построений координаты текущих точек должны сохраняться в матрицах **xx,yy,zz**. Их строки и столбцы обозначим номерами **nu,nv**, лежащими в диапазоне (0,10). Им в соответствие поставим шаги по аргументам **u,v** параметрического описания сплайна. Таких шагов будет 10 по каждому аргументу, а величина каждого шага составит 0.1.

Программа моделирования

Транспонирование геометрической матрицы Безье

$$\mathbf{M}^T := \mathbf{M}^T$$

Задание сетки шагов по аргументам параметрической системы координат. В программе наборы шагов задаются в виде ранжированных переменных:

$$nu := 0..10 \quad nv := 0..10$$

Получение абсолютных значений аргументов

$$u_{nu} := 0.1 \cdot nu \quad v_{nv} := 0.1 \cdot nv$$

Для вычисления текущих точек необходимы матрицы-строки **U** и **V**, содержащие степени параметров. Для каждого сочетания параметров **u** и **v** они принимают различные значения. Чтобы облегчить графические построения, эти наборы значений для всех сочетаний **u** и **v** запоминаются. В результате возникают двумерные матрицы. Ниже для этих матриц вычисляются четыре элемента каждой строки.

$$\begin{aligned} U_{nu,0} &:= (u_{nu})^3 & U_{nu,1} &:= (u_{nu})^2 & U_{nu,2} &:= (u_{nu}) & U_{nu,3} &:= 1 \\ V_{nv,0} &:= (v_{nv})^3 & V_{nv,1} &:= (v_{nv})^2 & V_{nv,2} &:= (v_{nv}) & V_{nv,3} &:= 1 \end{aligned}$$

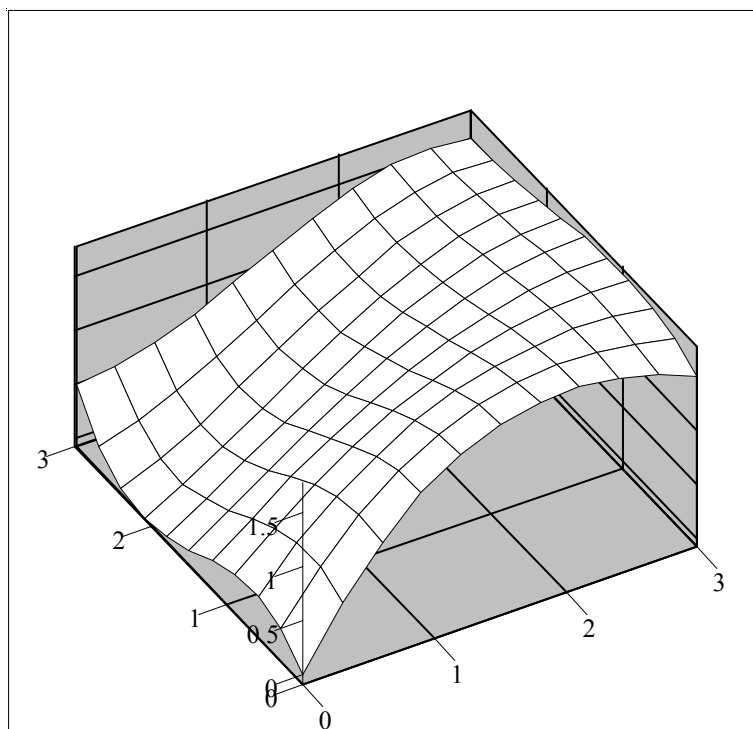
Транспонирование матрицы аргумента v .

$$VT := V^T$$

Вычисление координат текущих точек в матричной форме

$$xx := U \cdot M \cdot X \cdot MT \cdot VT \quad yy := U \cdot M \cdot Y \cdot MT \cdot VT \quad zz := U \cdot M \cdot Z \cdot MT \cdot VT$$

Построение трехмерного графика (Surface Plot) сплайновой поверхности



xx, yy, zz

Приложение Б. Моделирование *B*-сплайна по его параметрическому описанию

Исходными данными для моделирования являются координаты шестнадцати опорных точек и геометрическая (базисная) матрица *B*-сплайна. Нумерация опорных точек соответствует рисунку 3.6, а их координаты взяты из приложения А:

$$X := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \end{bmatrix} \quad Y := \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{bmatrix} \quad Z := \begin{bmatrix} 0 & 2 & 2 & 1.5 \\ 1 & 1.5 & 1.8 & 2 \\ -1 & 0 & 1.5 & 1.5 \\ 0.5 & 0.5 & 1.8 & 1.5 \end{bmatrix}$$

Геометрическая матрица B -сплайна

$$M := \frac{\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}}{6}$$

Координаты текущих точек сплайна вычисляются по параметрическим выражениям (3.5). Программа использует тот же алгоритм и те же обозначения, что и в приложении А.

Программа моделирования

Транспонирование геометрической матрицы

$$MT := M^T$$

Задание сетки шагов по аргументам параметрической системы координат. В программе наборы шагов задаются в виде ранжированных переменных:

$$nu := 0..10 \quad nv := 0..10$$

Получение абсолютных значений аргументов

$$u_{nu} := 0.1 \cdot nu \quad v_{nv} := 0.1 \cdot nv$$

Вычисление элементов матриц, содержащих степени аргументов-параметров.

$$U_{nu,0} := (u_{nu})^3 \quad U_{nu,1} := (u_{nu})^2 \quad U_{nu,2} := (u_{nu}) \quad U_{nu,3} := 1$$

$$V_{nv,0} := (v_{nv})^3 \quad V_{nv,1} := (v_{nv})^2 \quad V_{nv,2} := (v_{nv}) \quad V_{nv,3} := 1$$

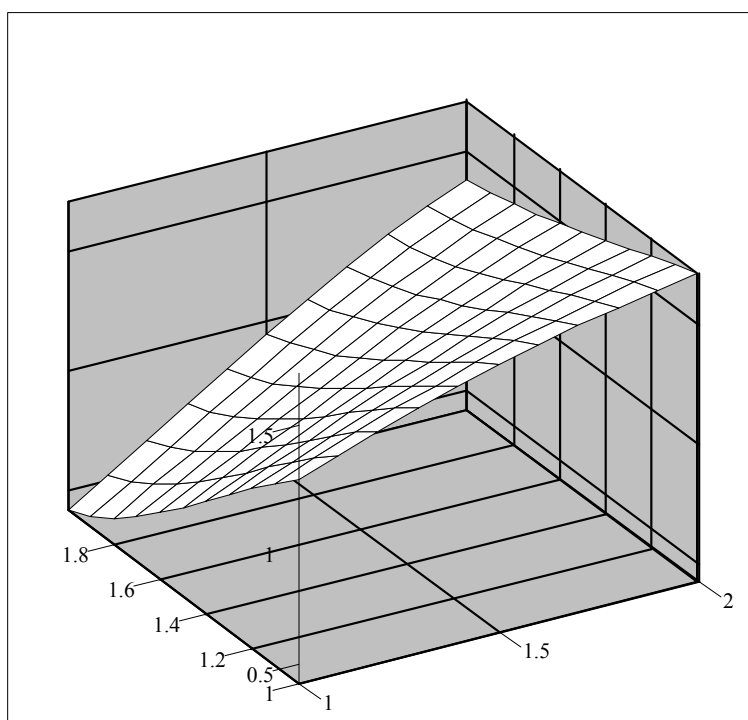
Транспонирование матрицы аргумента v .

$$VT := V^T$$

Вычисление координат текущих точек в матричной форме

$$xx := U \cdot M \cdot X \cdot MT \cdot VT \quad yy := U \cdot M \cdot Y \cdot MT \cdot VT \quad zz := U \cdot M \cdot Z \cdot MT \cdot VT$$

Построение трехмерного графика (Surface Plot) сплайновой поверхности



xx, yy, zz

Приложение В. Формирование отсека B -сплайновой поверхности на основе опорных точек различной кратности

На рисунках слева показаны изображения отсека B -сплайна, полученные MathCAD-программой при различной кратности опорных точек. На рисунках справа серым цветом отмечены формируемые отсеки на исходной поверхности, соответствующей набору опорных точек из приложения Б.

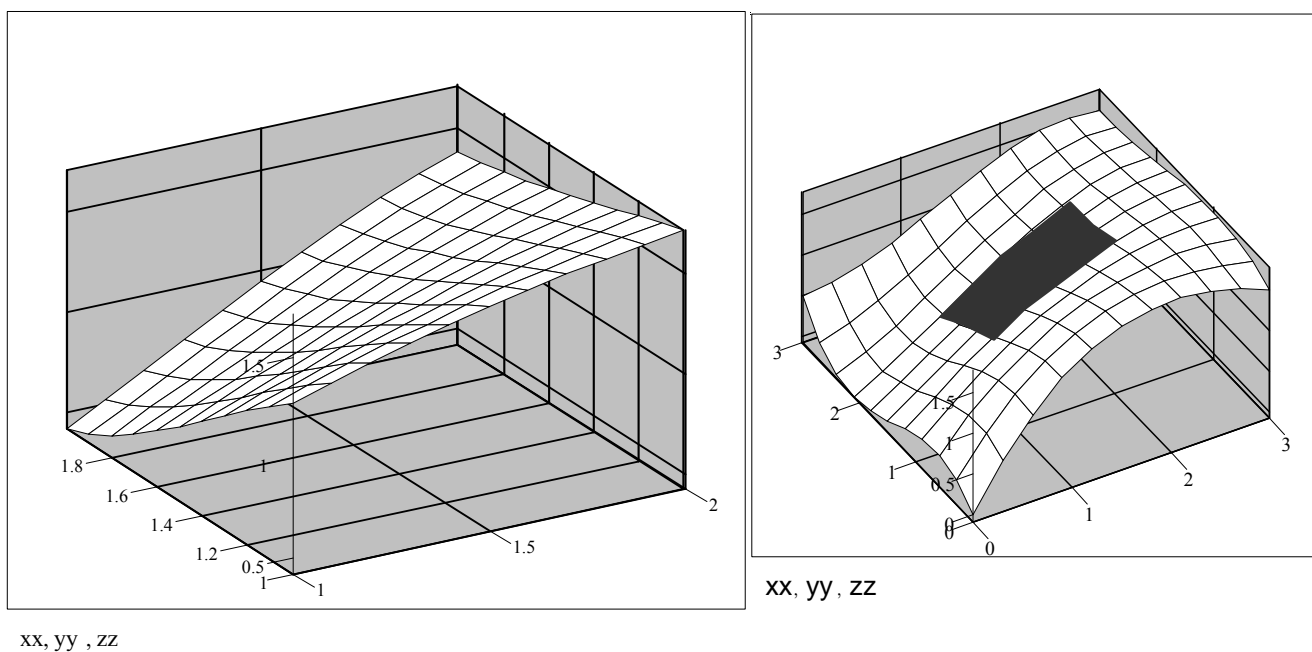


Рисунок В1 – Формирование отсека на основе исходного набора опорных точек (кратность граничных точек равна единице)

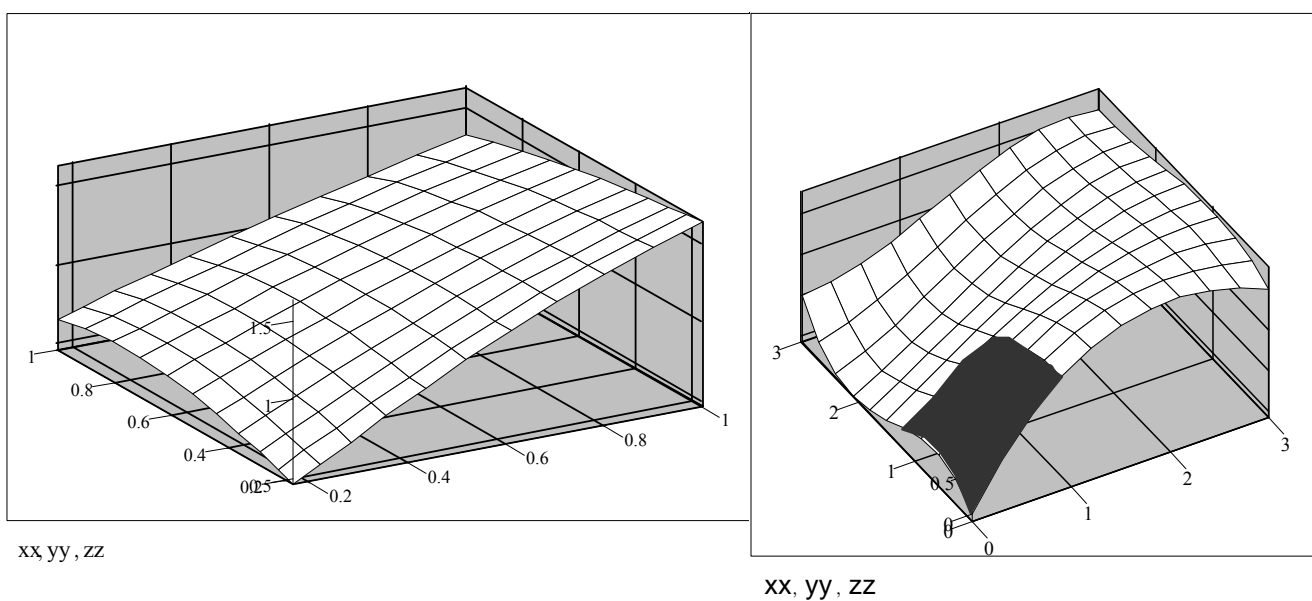
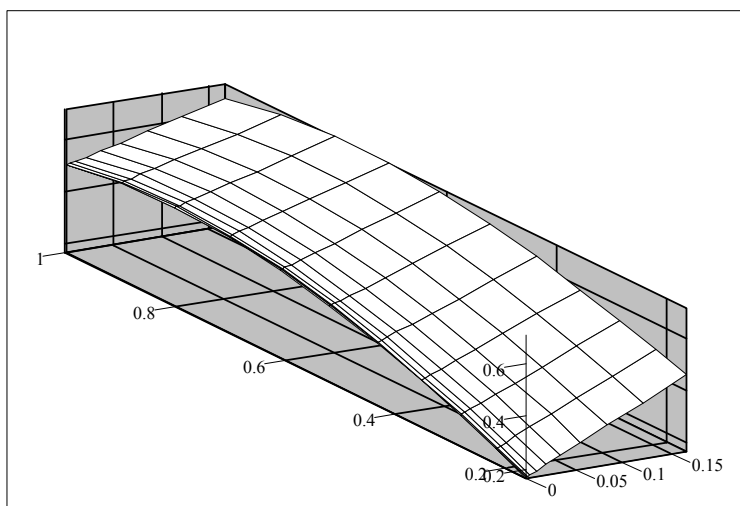
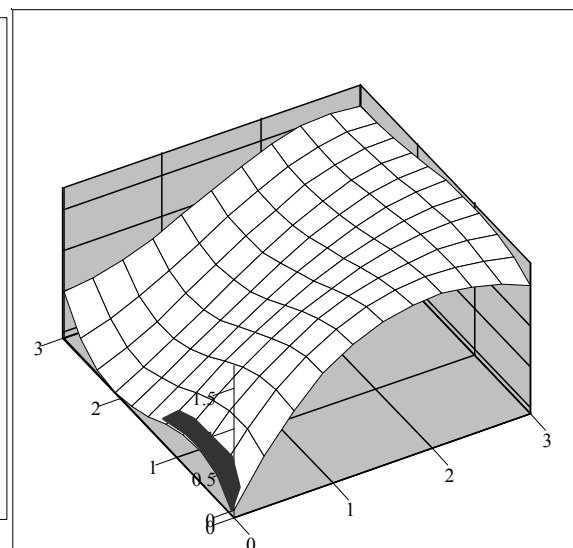


Рисунок В2 – Формирование отсека на основе набора кратных граничных опорных точек (кратность равна двум)

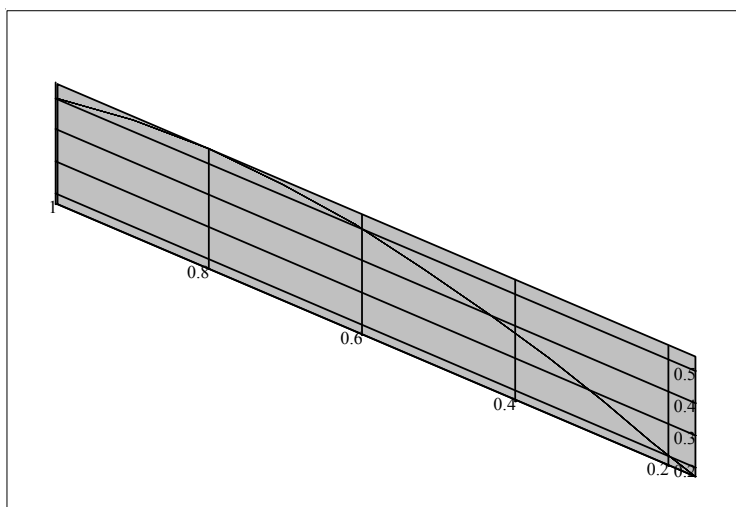


xx, yy, zz

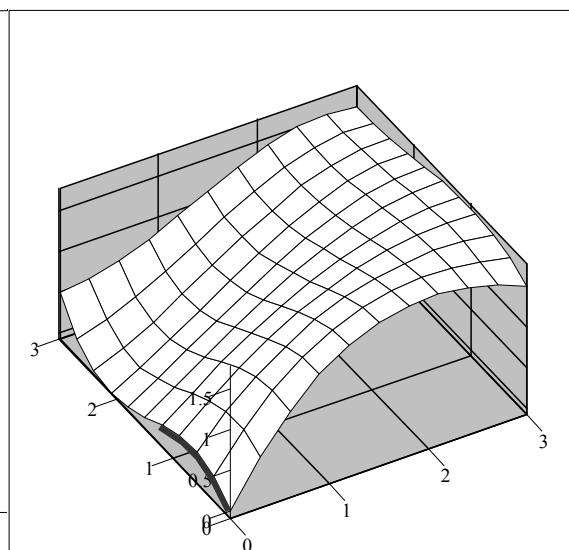


xx, yy, zz

Рисунок В3 – Формирование отсека на основе набора кратных граничных опорных точек (кратность равна трем)



xx, yy, zz



xx, yy, zz

Рисунок В4 – Формирование отсека на основе набора кратных граничных опорных точек (кратность равна четырем). Отсек выродился в кривую линию