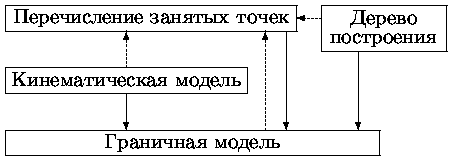
**Раздел 3. Типы геометрических моделей**

**Тема 3.1.** Типы представления геометрических 3D-моделей: граничное представление, в виде дерева построений, кинематическое представление, гибридные типы. Способы представления поверхности модели. Геометрические модели хранения и визуализации. Способы описания геометрических моделей.

**3.1.1. Типы представления геометрических 3D-моделей: граничное представление, в виде дерева построений, кинематическое представление, гибридные типы.**

Во многих приложениях машинной графики возникает потребность в представлении трехмерных тел (вычислительный эксперимент, автоматизация проектирования, роботизация, вычислительная томография, тренажеры, видеографика и т.д.). . Можно выделить два основных типа представлений 3D моделей: **- *граничное***, когда в модели хранятся границы объекта, например, вершины, ребра, грани, **-** в виде ***дерева построения***, когда хранятся базовые объекты (призма, пирамида, цилиндр, конус и т.п.), из которых формировалось тело и использованные при этом операции; в узле дерева сохраняется операция формирования, а ветви представляют объекты.Предельным случаем граничной модели является модель, использующая ***перечисление***всех точек занимаемого ею пространства. В частности, тело может быть аппроксимировано набором "склеенных" друг с другом параллелепипедов, что может быть удобно для некоторых вычислений (веса, объемы, расчеты методом конечных элементов и т.д.).Часто используются ***гибридные модели***, в которых в различной мере смешиваются эти два основных типа представления. В частности, в граничной модели может сохраняться информация о способе построения, например, информация о контуре и траектории его перемещения для формирования заданной поверхности (это так называемые ***кинематические* модели)**.В моделях в виде дерева построения в качестве элементарных могут использоваться не только базовые объекты, но также и сплошные тела, заданные с помощью границ.В общем случае нельзя утверждать, что одна модель во всем лучше другой. Так, например, граничная модель удобна для выполнения операций визуализации (удаление невидимых частей, закраска и т.п.), с другой стороны модель в виде дерева построения естественным образом может обеспечить параметризацию объекта, т.е. модификацию объекта изменением тех или иных отдельных параметров, вплоть до убирания каких-либо составных частей, но не удобна для визуализации, так как требует перевычисления объекта по дереву построения. Поэтому необходимы средства взаимного преобразования моделей. Понятно, что из более общей можно сформировать более простую, обратное преобразование далеко не всегда возможно или целесообразно, что и иллюстрируется сплошными и штриховыми линиями на **рис. 3.1.1**



**Рис. 3.1.1: Преобразования моделей представления**

Из **рис. 3.**[**1**](http://bourabai.ru/graphics/0209.htm#r10_1)**.1**  видно особое место граничной модели, преобразование в которую возможно из любых других. Учитывая это, а также и то, что эта модель наиболее удобна для визуализации дальнейшее рассмотрение будет, в основном, относиться к этой модели.

**3.1.2. Способы представления поверхности модели**.

Важной составной частью [геометрических моделей](javascript:termInfo(%22%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D1%85%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B5%D0%B9%22)) является описание поверхностей. Если поверхности детали — плоские грани, то модель может быть выражена достаточно просто определенной информацией о гранях, ребрах, вершинах детали. При этом обычно используется [**метод конструктивной геометрии**](javascript:termInfo(%22%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%20%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B9%20%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D0%B8%22))**.** Представление с помощью плоских граней имеет место и в случае более сложных поверхностей, если эти поверхности аппроксимировать множествами плоских участков —[***полигональными сетками***](javascript:termInfo(%22%D0%BF%D0%BE%D0%BB%D0%B8%D0%B3%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC%D0%B8%20%D1%81%D0%B5%D1%82%D0%BA%D0%B0%D0%BC%D0%B8%22))**.** Тогда можно [поверхностную модель](javascript:termInfo(%22%D0%BF%D0%BE%D0%B2%D0%B5%D1%80%D1%85%D0%BD%D0%BE%D1%81%D1%82%D0%BD%D1%83%D1%8E%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C%22)) задать одной из следующих форм:

1. модель есть список граней, каждая грань представлена упорядоченным списком вершин (циклом вершин); эта форма характеризуется значительной избыточностью, так как каждая вершина повторяется в нескольких списках.
2. модель есть список ребер, для каждого ребра заданы инцидентные вершины и грани.

Процесс построения 3D-изображения в виде полигональных сеток можно представить состоящим из трех этапов. На первом этапе поверхность преобразуется в множество многоугольников (полигонов). Далее выполняются геометрические преобразования и установки освещения. На заключительном третьем этапе, так называемом "*[рендеринг](javascript:termInfo(%22%D1%80%D0%B5%D0%BD%D0%B4%D0%B5%D1%80%D0%B8%D0%BD%D0%B3%22))*" (rendering), создается двумерное изображение из полученных на предыдущих этапах многоугольников.  
 Однако аппроксимация [полигональными сетками](javascript:termInfo(%22%D0%BF%D0%BE%D0%BB%D0%B8%D0%B3%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%BC%D0%B8%20%D1%81%D0%B5%D1%82%D0%BA%D0%B0%D0%BC%D0%B8%22)) при больших размерах ячеек сетки дает заметные искажения формы, а при малых размерах ячеек оказывается неэффективной по вычислительным затратам. Поэтому более популярны описания неплоских поверхностей кубическими уравнениями в форме Безье или B-сплайнов.

Знакомство с этими формами удобно выполнить, показав их применение для описания геометрических объектов первого уровня — пространственных кривых.

**Примечание 1.**

Геометрическими объектами нулевого, первого и второго уровней называют соответственно точки, кривые, поверхности.

В подсистемах графики и геометрического моделирования используются параметрически задаваемые кубические кривые  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=1/?k=10

|  |  |
| --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=2/?k=10 | (1) |

http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=3/?k=10

где http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=4/?k=10. Такими кривыми описывают сегменты аппроксимируемой кривой, т.е. аппроксимируемую кривую разбивают на сегменты и каждый сегмент аппроксимируют уравнениями (1).

Применение кубических кривых обеспечивает (соответствующим выбором четырех коэффициентов в каждом из трех уравнений) выполнение четырех условий сопряжения сегментов. В случае [*модели Безье*](javascript:termInfo(%22%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%20%D0%91%D0%B5%D0%B7%D1%8C%D0%B5%22)) этими условиями являются прохождение кривой сегмента через две заданные концевые точки и равенство в этих точках касательных векторов соседних сегментов. В случае [*B-сплайнов*](javascript:termInfo(%22B-%D1%81%D0%BF%D0%BB%D0%B0%D0%B9%D0%BD%D0%BE%D0%B2%22)) выполняются условия непрерывности касательного вектора и кривизны (т.е. первой и второй производных) в концевых точках сегментов. Использование [*сплайнов*](javascript:termInfo(%22%D1%81%D0%BF%D0%BB%D0%B0%D0%B9%D0%BD%D0%BE%D0%B2%22)) обеспечивает высокую степень "гладкости" кривой, хотя прохождение аппроксимирующей кривой через заданные точки здесь не обеспечивается. Применение полиномов выше третьей степени не рекомендуется, так как велика вероятность появления "волнистости".

В случае формы Безье коэффициенты в (1) определяются, во-первых, подстановкой в (1) значений http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=5/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=6/?k=10 и координат заданных концевых точек http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=7/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=8/?k=10соответственно, во-вторых, подстановкой в выражения производных  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=9/?k=10  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=10/?k=10  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=11/?k=10  
тех же значений http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=5/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=6/?k=10 и координат точек http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=12/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=13/?k=10, задающих направления касательных векторов (рис. 1). В результате для формы Безье получаем  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=14/?k=10

|  |  |
| --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=15/?k=10 | (2) |

http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=16/?k=10

где http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=17/?k=10 — вектор-строка, матрица http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=18/?k=10 представлена в табл. 1, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=19/?k=10 — вектор координат http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=20/?k=10 точек http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=7/?k=10, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=12/?k=10, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=13/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=8/?k=10, аналогично http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=21/?k=10, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=22/?k=10 — векторы координат http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=23/?k=10, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=24/?k=10 тех же точек.

**Таблица 1**

|  |  |  |  |
| --- | --- | --- | --- |
| -1 | 3 | -3 | 1 |
| 3 | -6 | 3 | 0 |
| -3 | 3 | 0 | 0 |
| 1 | 0 | 0 | 0 |

|  |
| --- |
| http://bigor.bmstu.ru/?img/?doc=100_Graph/4006.mod/?n=1 |

**Рис. 1.  Кривая Безье**

В случае B-сплайнов аппроксимируемая кривая делится на  http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=25/?k=10  участков, выделяемых последовательными точками http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=26/?k=10.  
 Участок между парой соседних точек http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=27/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=28/?k=10 аппроксимируется B-сплайном, построенным с использованием четырех точек http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=29/?k=10. B-сплайн на участке http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=30/?k=10 может быть представлен выражениями, аналогичными

(2)  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=14/?k=10  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=31/?k=10  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=16/?k=10

для которых матрица http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=18/?k=10 имеет иной вид и представлена в табл. 2, а векторы http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=19/?k=10, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=21/?k=10, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=22/?k=10 содержат соответствующие координаты точек http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=29/?k=10.

**Примечание 2.**

Покажем, что в точках сопряжения для первой и второй производных аппроксимирующего выражения выполняются условия непрерывности, что требуется по определению B-сплайна. Обозначим участок аппроксимирующего B-сплайна, соответствующий участку http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=32/?k=10 исходной кривой, через http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=33/?k=10. Тогда для этого участка и координаты х в точке сопряжения http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=34/?k=10 имеем http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=35/?k=10 и  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=36/?k=10  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=37/?k=10

Для участка http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=38/?k=10 в той же точке http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=34/?k=10 имеем http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=39/?k=10 и

http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=40/?k=10  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=41/?k=10

т.е. равенство производных в точке сопряжения на соседних участках подтверждает непрерывность касательного вектора и кривизны. Естественно, что значение http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=42/?k=10 координаты http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=43/?k=10 точки http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=34/?k=10 аппроксимирующей кривой на участке http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=33/?k=10

http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=44/?k=10  
равно значению http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=42/?k=10, подсчитанному для той же точки на участке http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=38/?k=10, но значения координат узловых точек http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=42/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4006.mod/?n=45/?k=10 аппроксимирующей и аппроксимируемой кривых не совпадают.

**Таблица 2**

|  |  |  |  |
| --- | --- | --- | --- |
| -1/6 | 1/2 | -1/2 | 1/6 |
| 1/2 | -1 | 1/2 | 0 |
| -1/2 | 0 | 1/2 | 0 |
| 1/6 | 2/3 | 1/6 | 0 |

Аналогично можно получить выражения для форм Безье и B-сплайнов применительно к поверхностям с учетом того, что вместо (1) используются кубические зависимости от двух переменных.

**3.1.3. Геометрические модели хранения и визуализации.**

Одним из наиболее ярких применений геометрического моделирования является компьютерная графика. Исходная информация для получения изображения некоторого объекта на экране компьютера поставляется геометрической моделью этого объекта. На практике используются различные способы визуализации геометрических объектов. Наиболее простой из них — отображение линиями. Он позволяет получить общее представление об объекте, его размерах и форме. С помощью компьютера можно получить изображение моделируемого объекта, близкое к его фотографии. Геометрической модели или ее частям можно придать любой цвет и освещенность. Кроме того, объект на экране можно заставить двигаться в реальном времени и тем самым получить видеофильм.

Для того чтобы увидеть, как выглядит объект, нужно смоделировать поведение потока лучей света, идущего от источников к модели и от поверхности модели к наблюдателю. При этом граням модели можно придать необходимый цвет, зеркальность, прозрачность, излучение, фактуру и другие физические свойства взаимодействия со световыми потоками. Модель можно осветить с разных сторон светом различного цвета и интенсивности. Реалистические отображения объектов строятся из отдельных точек определенного цвета и яркости, причем точки должны быть сравнительно небольшой величины и располагаться достаточно близко друг к другу (расстояние между точками не должно превосходить размеры точек).

Изображения геометрических объектов линиями называются векторными, а изображения геометрических объектов точками называются растровыми. Примеры векторного и растрового изображений приведены на рис. 9.1.1 и 9.1.2.

Растровая графика более информативна, чем векторная, так как позволяет получать тоновые изображения, но требует больше ресурсов и склонна к искажению при редактировании. В обоих случаях мы видим проекции геометрических объектов на выбранную плоскость.

Проекции могут быть построены с помощью линий, перпендикулярных проекционной плоскости, или с помощью линий, проходящих через общую точку. Первые называются параллельными проекциями (рис. 9.1.3), вторые — центральными проекциями или перспективными изображениями (рис. 9.1.4). Центральные проекции могут быть созданы на плоскости или криволинейной поверхности. Мы рассмотрим центральные проекции на плоскости.

Центральные проекции геометрических объектов ближе к тому изображению, которое возникает на сетчатке глаза, поэтому они дают более реалистические картины.

Хотя центральные проекции ближе к тому, что можно наблюдать в реальности, размеры отдельных частей объекта на центральных проекциях зависят от их расстояния до точки наблюдения, поэтому в качестве конструкторской документации чаще используются параллельные проекции.

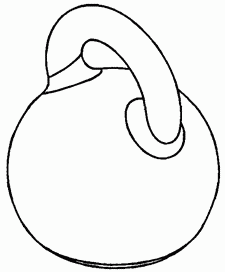


Рис. 9.1.1. Векторное изображение

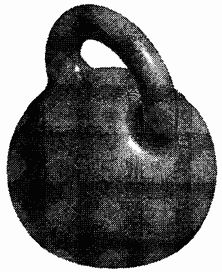


Рис. 9.1.2. Растровое изображение

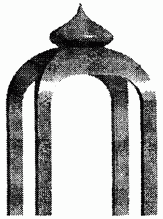


Рис. 9.1.3. Параллельная проекция

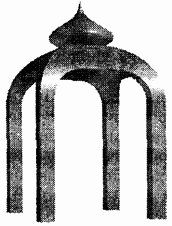


Рис. 9.1.4. Центральная проекция

Растровая и векторная графика, а также параллельные и центральные проекции не являются альтернативами. Они дополняют друг друга. В данной главе мы рассмотрим методы визуализации геометрических объектов с помощью компьютера или методы компьютерной графики.

**3.1.3.1. Методы и алгоритмы компьютерной графики**

К методам [компьютерной графики](javascript:termInfo(%22%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B9%20%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B8%22)) относят методы преобразования графических объектов, представления (развертки) линий в растровой форме, выделения окна, удаления скрытых линий, проецирования, закраски изображений.  
 Преобразование графических объектов выполняется с помощью операций переноса, масштабирования, поворота.  
 Перенос точки из положения http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=1/?k=10 в новое положение http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=2/?k=10 можно выполнять по формулам типа:  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=3/?k=10  
где http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=4/?k=10 — приращение по координате http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=5/?k=10. Однако удобнее операции преобразования представлять в единой матричной форме:

|  |  |
| --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=6/?k=10 | (1) |

где http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=7/?k=10 — преобразующая матрица. При этом точки http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=2/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=1/?k=10 в двумерном случае изображают векторами-строками 1×3, в которых кроме значений двух координат, называемых при таком представлении однородными, дополнительно указывают масштабный множитель http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=8/?k=10. Тогда перенос для случая 2D можно выразить в виде (1), где http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=7/?k=10 есть табл. 1, а http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=9/?k=10.

**Таблица 1**

|  |  |  |
| --- | --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=10/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=10/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=12/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=13/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=10/?k=10 |

Для операций масштабирования и поворота матрицы http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=7/?k=10 представлены в  
табл. 2 и табл. 3 соответственно, где http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=14/?k=10, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=15/?k=10 — масштабные множители, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=16/?k=10 — угол поворота.

Удобство (1) объясняется тем, что любую комбинацию элементарных преобразований можно описать формулой (1). Например, выражение для сдвига с одновременным поворотом имеет вид  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=17/?k=10  
где http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=18/?k=10, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=19/?k=10 — матрица сдвига, http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=20/?k=10 — матрица поворота.

**Таблица 2**

|  |  |  |
| --- | --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=21/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=22/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=10/?k=10 |

**Таблица 3**

|  |  |  |
| --- | --- | --- |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=23/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=24/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=25/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=23/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 |
| http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=11/?k=10 | http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=10/?k=10 |

Представление графических элементов в растровой форме требуется для отображения этих элементов на битовую карту растровой [видеосистемы](javascript:termInfo(%22%D0%B2%D0%B8%D0%B4%D0%B5%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B%22)). Пусть требуется развернуть отрезок http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=26/?k=10 прямой http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=27/?k=10, причем http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=28/?k=10 (при других значениях http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=29/?k=10 рассматриваемый ниже алгоритм остается справедливым после определенных модификаций). Введем обозначения: http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=30/?k=10 http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=31/?k=10 за величину дискрета (пиксела) примем единицу. В алгоритме развертки номера строк и столбцов карты, на пересечении которых должны находиться точки отрезка, определяются следующим образом:

1. http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=32/?k=10
2. http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=33/?k=10
3. если http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=34/?k=10, то http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=35/?k=10 иначе http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=36/?k=10
4. http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=37/?k=10
5. переход к пункту 3, пока не достигнута точка http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=38/?k=10.

Экономичность этого алгоритма обусловливается отсутствием длинных арифметических операций типа умножения.  
 [Выделение окна](javascript:termInfo(%22%D0%92%D1%8B%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BE%D0%BA%D0%BD%D0%B0%22)) требуется при определении той части сцены, которая должна быть выведена на экран  [дисплея](javascript:termInfo(%22%D0%B4%D0%B8%D1%81%D0%BF%D0%BB%D0%B5%D1%8F%22)).  
 Пусть окно ограничено линиями http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=39/?k=10 (рис. 1). Поочередно для каждого многоугольника проверяется расположение его вершин и ребер относительно границ окна. Так, для многоугольника http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=40/?k=10  (см. рис. 1) при отсечении по границе http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=41/?k=10 просматривается множество вершин в порядке обхода по часовой стрелке. Возможны четыре ситуации для двух последовательных вершин http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=1/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=42/?k=10:

1. если http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=43/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=44/?k=10, то обе вершины и инцидентное им ребро находятся вне окна и исключаются из дальнейшего анализа;
2. если http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=45/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=46/?k=10, то обе вершины и инцидентное им ребро остаются для дальнейшего анализа;
3. если http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=45/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=44/?k=10, то вершина http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=47/?k=10 остается в списке вершин, а вершина http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=48/?k=10 заменяется новой вершиной с координатами http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=49/?k=10 в нашем примере такой новой вершиной будет http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=50/?k=10;
4. если http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=43/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=46/?k=10, то вершина http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=47/?k=10 заменяется новой вершиной с координатами http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=51/?k=10 а вершина http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=48/?k=10 остается в списке вершин; в нашем примере новой вершиной будет http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=52/?k=10.

После окончания просмотра применительно ко всем границам в окне оказываются оставшиеся в списке вершины.

|  |
| --- |
| http://bigor.bmstu.ru/?img/?doc=100_Graph/4001.mod/?n=1 |

**Рис. 1.  Выделение окна**

Применяя эти правила в нашем примере, получаем сначала многоугольник http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=53/?k=10, а после отсечения по верхней границе http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=54/?k=10 — многоугольник http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=55/?k=10 (см. рис. 1). Однако правильный результат несколько иной, а именно многоугольник http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=56/?k=10. Этот правильный результат получается при двойном обходе вершин сначала по часовой стрелке, затем против с включением в список новых вершин, появляющихся при каждом обходе.  
 Применяют ряд алгоритмов [***удаления скрытых линий***](javascript:termInfo(%22%D1%83%D0%B4%D0%B0%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F%20%D1%81%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D1%85%20%D0%BB%D0%B8%D0%BD%D0%B8%D0%B9%22))**.** Один из наиболее просто реализуемых алгоритмов — алгоритм z-буфера, где [*z-буфер*](javascript:termInfo(%22z-%D0%B1%D1%83%D1%84%D0%B5%D1%80%22)) — область памяти, число ячеек в которой равно числу пикселов в окне вывода. Предполагается, что ось http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=57/?k=10 направлена по нормали к видовой поверхности и наблюдатель расположен в точке http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=58/?k=10.  
 В начале исполнения алгоритма все пикселы соответствуют максимальному значению http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=57/?k=10, т.е. максимальному удалению от наблюдателя, что приводит к помещению во все ячейки z-буфера значений пикселов фона картины (чертежа). Далее поочередно для всех точек граней рассчитываются значения координаты http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=57/?k=10. Среди точек, относящихся к одному и тому же пикселу (одной и той же ячейке z-буфера http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=59/?k=10), выбирается точка с наименьшим значением http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=57/?k=10 и ее код (т.е. цвет и яркость) помещается в http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=59/?k=10. В итоге z-буфер будет содержать пикселы наиболее близких к наблюдателю граней.  
 Алгоритмы построения проекций преобразуют трехмерные изображения в двумерные. В случае построения центральной проекции каждая точка трехмерного изображения отображается на картинную поверхность путем пересчета координат http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=60/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=61/?k=10 (рис. 2). Так, координату http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=62/?k=10 точки http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=63/?k=10 вычисляют по очевидной формуле  
http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=64/?k=10 аналогично рассчитывается координата http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=65/?k=10 точки http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=63/?k=10.

|  |
| --- |
| http://bigor.bmstu.ru/?img/?doc=100_Graph/4001.mod/?n=2 |

**Рис. 2.  Построение центральной проекции точки A**

В параллельных проекциях  http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=66/?k=10  и координаты http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=60/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=61/?k=10 точек http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=63/?k=10 и http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=67/?k=10 совпадают. Поэтому построение параллельных проекций сводится к выделению окна, при необходимости к повороту изображения и возможно к удалению скрытых линий.

Моделирование эффектов отражения света от поверхности объекта в геометрических моделях называют [*рендерингом*](javascript:termInfo(%22%D1%80%D0%B5%D0%BD%D0%B4%D0%B5%D1%80%D0%B8%D0%BD%D0%B3%D0%BE%D0%BC%22)). Закраска матовых поверхностей основана на законе Ламберта, согласно которому яркость отраженного от поверхности света пропорциональна http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=68/?k=10, где http://bigor.bmstu.ru/?frm/?doc=100_Graph/4001.mod/?n=69/?k=10 — угол между нормалью к поверхности и направлением луча падающего света. В [*алгоритме Гуро*](javascript:termInfo(%22%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B5%20%D0%93%D1%83%D1%80%D0%BE%22)) яркость внутренних точек рассматриваемой поверхности определяется линейной интерполяцией яркости в вершинах многоугольника. При этом сначала проводится интерполяция в точках ребер, а затем по строкам горизонтальной развертки. Более реалистичными получаются изображения в [*алгоритме Фонга*](javascript:termInfo(%22%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B5%20%D0%A4%D0%BE%D0%BD%D0%B3%D0%B0%22)), основанном на линейной интерполяции векторов нормалей к поверхности. Один из алгоритмов рендеринга заключается в [*трассировке лучей*](javascript:termInfo(%22%D1%82%D1%80%D0%B0%D1%81%D1%81%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B5%20%D0%BB%D1%83%D1%87%D0%B5%D0%B9%22)) — моделировании прохождения лучей света между источниками, поверхностями и наблюдателем.

**3.1.3.2. Программы компьютерной графики**

[*Графические редакторы*](javascript:termInfo(%22%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D1%8B%22)) предназначены для создания, сохранения и воспроизведения графических изображений. Различают редакторы векторные и растровые .  
 Векторные редакторы позволяют создавать изображения типа схем, чертежей, диаграмм, простых рисунков. Одним из наиболее известных векторных редакторов является Corel Draw. Характерные возможности этого редактора — привязка линий к определенной позиции на изображении, создание выносок и размерных линий, работа с многостраничными изображениями, стандартный интерфейс Windows, многоуровневый откат, кроссплатформенность (имеются версии для Windows, OS/2, MAC, различных вариантов UNIX). Примеры других векторных редакторов — Microsoft Paint, iGrafx Designer, Visio 2000, Adobe Illustrator.  
 Растровые графические редакторы выполняют такие функции, как воспроизведение сложных изображений типа фотографий, их редактирование с помощью операций закрашивания, обрезания, маскирования, изменения параметров всего изображения или его участков. В этом классе графических редакторов лидирующее положение занимает Adobe Photoshop. При его использовании все изображение строится из набора отдельных слоев-картинок, имеющих прозрачные и закрашенные участки. Можно создавать, удалять, копировать, комбинировать участки, регулировать прозрачность и порядок расположения слоев. В пакете предусмотрены возможности построения и редактирования также векторных изображений. Благодаря подключению дополнительных модулей можно расширять функциональность пакета Adobe Photoshop. В частности, поэтому с помощью Adobe Photoshop удается выполнять полный цикл работ по созданию Web-страниц. Один из подключаемых модулей PhotoGraphics — дополнительное средство для рисования векторных объектов непосредственно в Photoshop. В модуле пользователю предоставляются все основные инструменты рисования таких объектов, как линии, прямоугольники, овалы, текст и т.п.  
 Другой растровый редактор Painter 6 позволяет создавать изображения, имитирующие картины художников. Наряду с этим в нем реализованы обычные функции растровых редакторов, есть возможность создавать анимационные изображения.  
 Для воспроизведения сложных изображений, например фотографий, используют просмотрщики популярных 2D и 3D форматов; графические и видео просмотрщики и конверторы соответствующих форматов.  
 Для синтеза трехмерных изображений в САПР преимущественно используют средства таких известных систем, как CATIA, Unigraphics, Inventor, Solidworks и др. Однако кроме них, существуют и применяются [*графические библиотеки*](javascript:termInfo(%22%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B8%22)) и пакеты [компьютерной графики](javascript:termInfo(%22%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D0%B9%20%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B8%22)) и [геометрического моделирования](javascript:termInfo(%22%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B3%D0%BE%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%22)), непосредственно не связанные с проектированием в технике. К числу графических библиотек, получивших наибольшее распространение, относится библиотека графических процедур OpenGL.  
 [*OpenGL*](javascript:termInfo(%22OpenGL%22)) (Open Graphics Library) — открытая графическая библиотека и спецификация (стандарт) на интерфейсы программирования трехмерной графики. Библиотеку выпускают такие корпорации, как Microsoft, Silicon Graphics, а также просто группы программистов. Стандарт OpenGL разрабатывается фирмой Silicon Graphics с 1992 г. В библиотеке OpenGL имеются процедуры построения [*графических примитивов*](javascript:termInfo(%22%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D1%85%20%D0%BF%D1%80%D0%B8%D0%BC%D0%B8%D1%82%D0%B8%D0%B2%D0%BE%D0%B2%22)) (к числу которых относят точки, линии, полигоны), удаления невидимых линий, [Z-буферизации](javascript:termInfo(%22z-%D0%B1%D1%83%D1%84%D0%B5%D1%80%22)), реалистичной засветки, задания положения тел и камеры наблюдения и т.п. Примитивы могут быть как векторными, так и растровыми. Важной особенностью технологии OpenGL является инвариантность к аппаратно-программным платформам.  
 Единицей информации в OpenGL является вершина, из вершин формируются более сложные объекты. Программист создает вершины, указывает, как их соединять (линиями или многоугольниками), устанавливает координаты и параметры камеры и ламп, а библиотека OpenGL берет на себя работу создания изображения на экране. С помощью OpenGL создаются трехмерные поверхности, текстуры, моделируются источники света, создаются эффекты тумана, прозрачности, смешивания цветов, имеются операции анимации с передвижением объектов сцены, ламп и камер по заданным траекториям .  
 Альтернативой OpenGL для платформы Microsoft является комплекс *[DirectX](javascript:termInfo(%22DirectX%22))* (Direct eXtension). Он представлет собой [API](javascript:termInfo(%22API%22)) для решения задач, относящихся к мультимедийным приложениям, в частности, для разработки компьютерных игр. В DirectX входит несколько API, один из них Direct3D используется для 3D-графики, другие - для звука, музыки, устройств ввода и т.д.

Среди других систем 3D-графики выделяют редактор Studio Max 4. Это высокоразвитый, но дорогой и довольно сложный в освоении комплекс моделирования, визуализации и анимации объектов, разработанный фирмой Autodesk для операционной среды Windows. Для генерации ландшафтов типа горных пейзажей, водных поверхностей, фантастических картин подходит 3D-редактор Bryce. К числу 3D-редакторов можно отнести соответствующие компоненты из программного обеспечения САПР, а также программы Adobe Dimention, CorelDream 3D, CosmoWorlds и ряд других.

**3.1.4. Способы описания геометрических моделей.**

В настоящее время применяют следующие подходы к построению и представлению [геометрических моделей](javascript:termInfo(%22%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D1%85%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D0%B5%D0%B9%22)).

1. **[Граничное представление](javascript:termInfo(%22%D0%93%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%22))** (Boundary-representation или B-rep) — задание граничных элементов детали – поверхностей (граней), ребер, вершин. Например,[модель](javascript:termInfo(%22%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C%22)) B-rep с плоскими поверхностями может быть задана списком граней вместе с инцидентными им ребрами и списком ребер с инцидентными им вершинами. Поверхности сложной формы дополнительно задаются или уравнениями поверхностей или результатами применения функций создания примитивов. К числу таких функций относятся заметание (sweeping), натягивание (skinning), сопряжение (blending). [*Заметание*](javascript:termInfo(%22%D0%97%D0%B0%D0%BC%D0%B5%D1%82%D0%B0%D0%BD%D0%B8%D0%B5%22)) (называемое также протягиванием) составляет основу кинематического метода синтеза поверхностей, согласно которому задают двумерный контур и траекторию его перемещения, а след от перемещения контура принимают в качестве поверхности детали. [*Натягивание*](javascript:termInfo(%22%D0%9D%D0%B0%D1%82%D1%8F%D0%B3%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%22)) заключается в натягивании поверхности на заданные плоские поперечные сечения тела. Сопряжение – функция скругления острых ребер, образуемых при пересечении поверхностей.

1. **[Позиционный метод](javascript:termInfo(%22%D0%9F%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9%20%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%22))** (называемый также декомпозиционным), в соответствии с которым рассматриваемое пространство разбивают на ячейки (позиции) и деталь задают указанием ячеек, принадлежащих детали. Ячейки могут иметь форму параллелепипедов одинаковых размеров (воксельное представление), более экономную форму параллелепипедов кратных размеров (октантное представление) или ячейки могут быть неодинаковой формы (ячеечное представление). Очевидно, что с ростом числа ячеек увеличивается точность моделирования, но модели становятся весьма громоздкими.

1. **[Метод конструктивной геометрии](javascript:termInfo(%22%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%20%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B9%20%D0%B3%D0%B5%D0%BE%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D0%B8%22))** (Constructive Solid Geometry)— представление сложной детали в виде совокупностей [*базовых элементов формы*](javascript:termInfo(%22%D0%B1%D0%B0%D0%B7%D0%BE%D0%B2%D1%8B%D1%85%20%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%20%D1%84%D0%BE%D1%80%D0%BC%D1%8B%22)) (БЭФ) и выполняемых над ними теоретико-множественных операций. Этот подход называют также объектно-ориентированным моделированием или feature-based modeling. Это основной способ конструирования сборочных узлов в современных САПР-К. К БЭФ относятся заранее разработанные модели простых тел, это, в первую очередь, модели параллелепипеда, цилиндра, сферы, призмы. Типичными теоретико-множественными операциями являются объединение, пересечение, разность. Например, модель плиты с отверстием в ней может быть получена вычитанием цилиндра из параллелепипеда.  
    Метод конструктивной геометрии порождает еще один способ построения геометрических моделей, называемый конструктивным представлением. [*Конструктивное представление*](javascript:termInfo(%22%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%22)) объемной геометрии — это описание последовательности применения операций при создании геометрической модели. Обычно история синтеза модели из БЭФ соответствует последовательности операций при изготовлении деталей, что удобно при проектировании технологических процессов в [системах CAM](javascript:termInfo(%22%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%D1%85%20CAM%22)).

Рассмотренные модели хранятся и обрабатываются в векторной форме. Однако для визуализации в современных рабочих станциях в связи с использованием в них растровых дисплеев необходима [*растеризация*](javascript:termInfo(%22%D1%80%D0%B0%D1%81%D1%82%D0%B5%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F%22)) — преобразование модели в растровую форму. Обратную операцию перехода к векторной форме называют векторизацией, векторная форма характеризуется меньшими затратами памяти. В частности, векторизация должна выполняться по отношению к данным, получаемым сканированием изображений в устройствах автоматического ввода.