

Лекция 7. Графические примитивы: многоугольники и шрифты

Определение, примеры, классификация,
способы компьютерного описания и хранения



Многоугольники (определение и примеры)

- Согласно математическому определению многоугольник – это плоская фигура, которая задается с помощью набора из 3-х или больше точек – вершин, последовательно соединенных прямолинейными отрезками, которые называются ребрами или сторонами многоугольника. Кроме того, в классической геометрии необходимо, чтобы ребра многоугольников не имели общих точек, кроме вершин.
- Примеры многоугольников:
 - треугольники,
 - прямоугольники,
 - восьмиугольники,
 - десятиугольники.
- В приложениях компьютерной графики возможны такие ситуации, когда не все перечисленные вершины многоугольника лежат строго в одной плоскости. Это может быть связано с ошибкой округления при вычислении значений, либо с ошибкой определения координат вершин или (что более характерно) с аппроксимацией криволинейной поверхности набором многоугольных участков. Один из способов исправления этой ситуации – просто разделить заданную сетку многоугольников на треугольники.

Классификация многоугольников

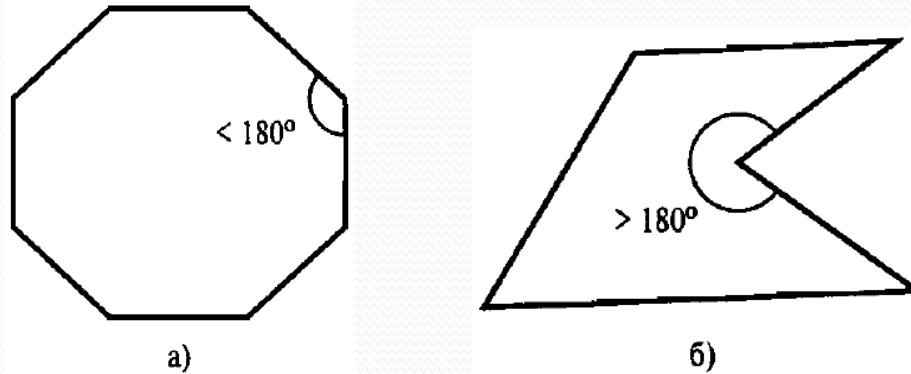


Рис. 1. Многоугольники: а) выпуклый; б) невыпуклый

Внутренним углом многоугольника называется угол внутри границы многоугольника, образованный двумя соседними сторонами. Если все внутренние углы многоугольника меньше или равны 180 градусам, то многоугольник выпуклый.

Другие эквивалентные определения выпуклого многоугольника:

- Весь многоугольник должен лежать по одну сторону от бесконечной прямой, являющейся продолжением любой из его сторон.
- Если выбрать любые 2 точки внутри выпуклого многоугольника, то прямолинейный отрезок, соединяющий эти точки, также будет полностью находиться внутри многоугольника.

Невыпуклые многоугольники создают дополнительные проблемы.

Реализация алгоритмов закрашивания и других графических процедур для невыпуклых многоугольников намного сложнее, поэтому, как правило, перед обработкой эффективнее разделить невыпуклый многоугольник на набор выпуклых. Как и другие алгоритмы предварительной обработки многоугольников, функции разделения невыпуклого многоугольника часто не входят в графическую библиотеку.

Для некоторых графических пакетов, в том числе и OpenGL, необходимо, чтобы все закрашиваемые многоугольники были выпуклыми. Кроме того, в некоторых системах допускается только закрашивание треугольных областей, что значительно облегчает многие алгоритмы создания изображений.

Распознавание невыпуклых многоугольников

Признаки невыпуклого многоугольника, которые можно положить в основу алгоритма распознавания невыпуклых многоугольников :

- по крайней мере один внутренний угол больше чем 180 градусов;
- продолжения некоторых сторон невыпуклого многоугольника будут пересекать другие ребра, а вершины находиться по разные стороны от такого продолжения;
- некоторые пары внутренних точек дадут отрезки, пересекающие границы многоугольника.

Таблицы многоугольников

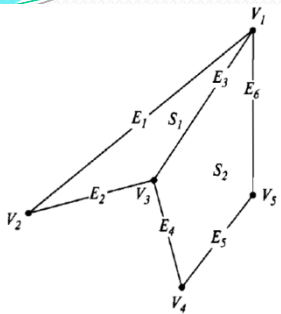


Таблица вершин	
$V_1:$	x_1, y_1, z_1
$V_2:$	x_2, y_2, z_2
$V_3:$	x_3, y_3, z_3
$V_4:$	x_4, y_4, z_4
$V_5:$	x_5, y_5, z_5

Таблица сторон	
$E_1:$	V_1, V_2
$E_2:$	V_2, V_3
$E_3:$	V_3, V_1
$E_4:$	V_3, V_4
$E_5:$	V_4, V_5
$E_6:$	V_5, V_1

Таблица поверхностных граней	
$S_1:$	E_1, E_2, E_3
$S_2:$	E_3, E_4, E_5, E_6

Таблица сторон, дополненная указателями на грани	
$E_1:$	V_1, V_2, S_1
$E_2:$	V_2, V_3, S_1
$E_3:$	V_3, V_1, S_1, S_2
$E_4:$	V_3, V_4, S_2
$E_5:$	V_4, V_5, S_2
$E_6:$	V_5, V_1, S_2

Рис. 2

Объекты сцены часто описываются как наборы многоугольных граней поверхностей. Графические пакеты часто предлагают функции определения формы поверхности в виде сетки многоугольных участков.

Описание каждого объекта включает информацию о координатах, с помощью которых задается геометрия многоугольных граней, и других параметров поверхности, таких как цвет, прозрачность и светоотражающие свойства. Эти данные размещаются в таблицах, которые используются при последующей обработке, выводе на экран и выполнении различных операций с объектами сцены.

Информационные таблицы многоугольников можно объединить в две группы: геометрические таблицы и таблицы атрибутов. В геометрических таблицах содержатся координаты вершин и параметры, которые позволяют определить пространственную ориентацию многоугольных поверхностей.

К атрибутам объекта, например, относятся величины, определяющие степень прозрачности объекта, отражающую способность его поверхности и текстурные характеристики.

Геометрическую информацию об объектах сцены удобно распределить по трем спискам: таблице вершин, таблице сторон и таблице граней поверхности граней (рис. 2).

Координаты всех вершин объекта записываются в таблицу вершин.

В таблице сторон содержатся ссылки на таблицу вершин, позволяющие определить вершины, принадлежащие каждой стороне многоугольника.

В таблице граней поверхности содержатся ссылки на таблицу сторон, определяющие границы каждого многоугольника.

Далее можно сформировать и другие таблицы:

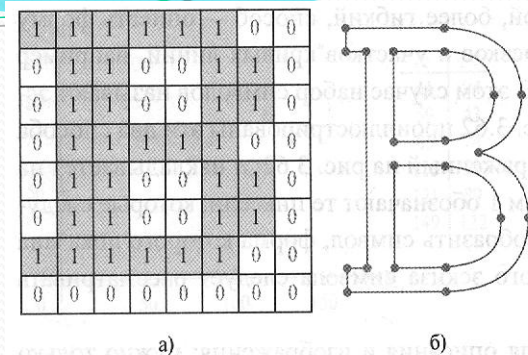
- Например, расширенную таблицу сторон, включающую указатели на таблицу поверхностных граней, чтобы быстрее определить общие стороны многоугольников. Это особенно удобно для процедур визуализации, в которых затенение поверхности при переходе через ребро от одного многоугольника к другому должно изменяться плавно.
- Аналогичным образом, для более быстрого доступа к данным можно расширить таблицу вершин, дополнив ее ссылками на соответствующие стороны.
- Можно составить расширенную таблицу граней (многоугольников) с указанием отдельных поверхностей в сцене, которые они формируют.

Примитивы символов

Рис. 3. Буква «В», представленная в виде:

а) узора битового отображения размером 8 на 8;

б) с помощью контура, состоящего из прямолинейных отрезков и участков кривых линий.



а)

б)

- Буквы, цифры и другие символы могут иметь разный размер и изображаться в различных стилях.
- Общий вид стиля набора символов называется начертанием. Сегодня существуют тысячи различных начертаний (шрифтов), доступных разным компьютерным приложениям. Примеры нескольких широко распространенных начертаний – Courier, Helvetica, Times New Roman, Arial и Tahoma.
- Высота всех символов шрифта 72 пункта приблизительно равна 2,54 см (1 дюйм), 14 пунктов составляет приблизительно 0.5 см.
- Шрифты можно разделить на две большие группы: с засечками и без засечек. У шрифта с засечками есть небольшие черточки, или штрихи, в конце главных штрихов знаков, тогда как при начертании без засечек таких штрихов нет. Например, этот текст набран шрифтом с засечками (Times New Roman). А это предложение напечатано шрифтом без засечек (Arial). Шрифт с засечками, как правило, более удобочитаемый; т.е. при таком шрифте легче читать большие фрагменты текста. С другой стороны, в шрифтах без засечек легче распознаются отдельные символы. По этой причине шрифты без засечек называют более разборчивыми. Эти шрифты хороши для обозначений и коротких названий.
- Также шрифты классифицируются по тому, являются ли они моноширинными или пропорциональными. При моноширинном начертании все символы имеют одинаковую ширину. Для пропорциональных шрифтов характерно изменение ширины символов.
- При хранении компьютерных шрифтов используются два различных вида (рис. 3). Простой способ описания формы символа определенного начертания – это задать узор из бинарных значений прямоугольной решетки. В таком случае набор символов называют растровым (битовым) шрифтом (рис. 3,а). Другой, более гибкий, способ – описать форму символов с помощью прямолинейных отрезков и участков кривых линий, например так, как это делается в шрифте PostScript. В этом случае набор символов называют векторным шрифтом (рис. 3,б).
- Растровые шрифты самые простые для описания и изображения: нужно только перенести сетку символа в буфер кадра. Однако в общем случае для записи растровых шрифтов необходимо больше памяти, поскольку нужно записывать каждую версию (с разными размерами и форматом) шрифта.
- Для записи векторных шрифтов требуется меньше памяти, поскольку для каждой версии не нужен отдельный кэш шрифта. Полуужирный шрифт, курсив или шрифты различных размеров можно создавать путем изменения определений кривых, образующих контуры символов. Однако для обработки векторных шрифтов нужно больше времени, поскольку для них необходимо выполнять процедуру растеризации кривых с закраской внутренней области.