АЛГОРИТМЫ ТРИАНГУЛЯЦИИ

Александр РОМАНЮК, Александр СТОРЧАК

Опубликовано в журнале "КОМПЬЮТЕРЫ+ПРОГРАММЫ", № 1, 2001 г.

Будущее машинной графики связано с формированием трехмерных изображений с высокой степенью реализма. Необходимость их формирования возникает во многих областях деятельности человека, например, в машиностроительном и архитектурном моделировании, дизайне, рекламе, мультипликации и т.д. Трехмерные (*3D*) изображения наиболее информативны, поскольку позволяют оценить конструктивные и эстетические особенности объектов. Системы синтеза реалистических изображений должны обеспечивать передачу всех свойств моделируемого объекта: объемность, расположение, передачу полутонов, тени, освещение, текстуры поверхности. Чем выше степень реалистичности изображения, тем больше требуется вычислений для его формирования.

Генерация объемных изображений представляет сложную вычислительную задачу, в связи этим на практике выполняют ее декомпозицию. Сложные изображения формируют из фрагментов объектов, для чего их разбивают на составные части. Процесс разбиения поверхности объектов на полигоны получил название *тесселяции*. Эта стадия на данном этапе развития машинной графики проводится полностью программно вне зависимости от технического уровня 3D-аппаратуры.

В настоящее время появилось большое разнообразие графических акселераторов, которые имеют различные аппаратные графические функции для закраски трехмерных объектов, удаления невидимых частей, наложения текстур и т.п. Для использования преимуществ 3D-ускорителей необходимо сначала программно произвести тесселяцию исходных объектов, а затем передать полученные полигональные области для дальнейшей обработки акселератору.

На практике наиболее часто производится разбиение изображений на треугольники. Это объясняется следующими причинами:

- **треугольник является простейшим полигоном, вершины которого однозначно задают грань;**
- **любую область можно гарантировано разбить на треугольники;**

- **вычислительная сложность алгоритмов разбиения на треугольники существенно меньше, чем при использовании других полигонов;**

- **реализация процедур рендеринга наиболее проста для области, ограниченной треугольником;**

- **для треугольника легко определить три его ближайших соседа, имеющих с ним общие грани.**

Процесс разбиения полигональной области со сложной конфигурацией в набор треугольников называется*триангуляцией*. При анализе или синтезе сложных поверхностей их аппроксимируют сеткой треугольников, и впоследствии оперируют с простейшими полигональными областями, т.е. с каждым из треугольников.

Особый интерес к алгоритмам триангуляции определяется тем, что они используются во многих процедурах машинной графики, таких как формирование поверхностей, закраска, удаление невидимых частей, отсечение.

Приведем пример создания полусферы, проиллюстрировав этапы рендеринга. Полигональная область (рис. 1,а) представляется совокупностью треугольников (рис. 1,б). При этом важно достаточно точно аппроксимировать внешний контур.

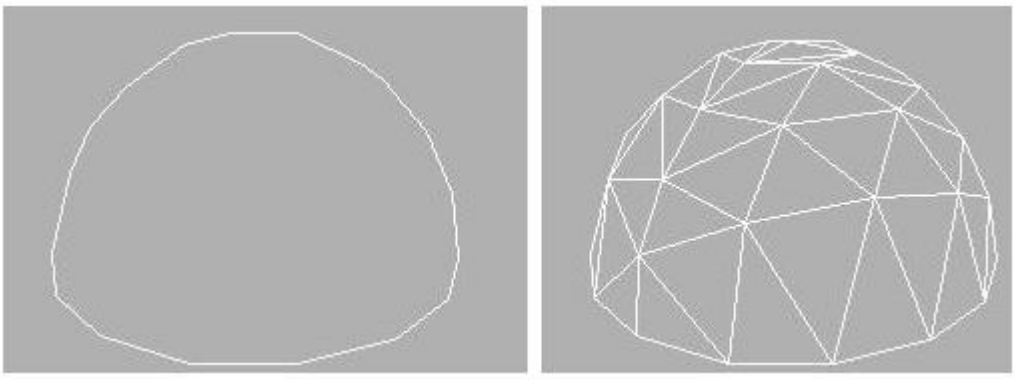

Рис. 1 а) Полигональная область б) Триангуляционная сеть

Любая поверхность может быть аппроксимирована с необходимой точностью сеткой треугольников. Точность аппроксимации определяется количеством треугольников и способом их выбора. Для качественной визуализации объекта, находящегося вблизи точки наблюдения, требуется учесть во много раз больше треугольников, чем в ситуации, когда тот же объект расположен на удалении. Даже достаточно грубая сетка полезна на практике, так как методы сглаживания, применяемые в процессе отображения,

могут значительно улучшить представление о кривизне поверхности.

Следующий этап – закрашивание граней (рис. 2), ограниченных треугольниками.

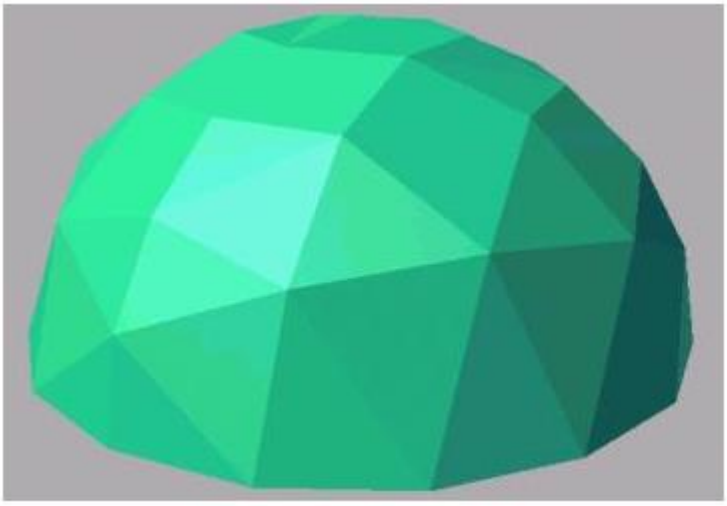

Рис. 2 Закрашивание граней объекта.

Последним этапом является применение алгоритмов сглаживания, которые устраняется ребристость поверхности, эффект Маха. На рисунке 3 представлена уже гладкая полусфера с плавным переходом полутонов и областями освещения.

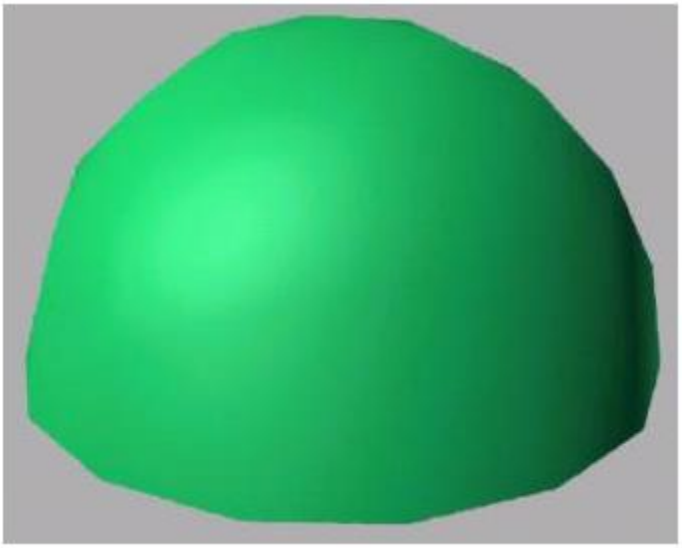

Рис. 3 Сглаживание поверхности

Сколько треугольников нужно?

При моделировании объекта возникает задача, какое количество треугольников нужно использовать? С одной стороны нужно

достичь максимального качества при окончательной визуализации, а с другой снизить нагрузку на процессор и 3D-акселератор. Для снижения нагрузки можно использовать менее детализированную триангуляционную сеть, но при этом качество будет далеко не самым лучшим.

При формировании динамических изображений точность представления объектов в кадре из-за его перемещения не воспринимается человеком, в связи с чем нецелесообразно использовать детальную триангуляцию.

При выборе подходов и алгоритмов триангуляции следует руководствоваться целью, которая ставится конкретной задачей. Например, при моделировании объекта, находящегося вдалеке, совершенно нет смысла применять очень детальную сеть для его описания и, наоборот, для близких объектов это очень важно, например, для построения модели головы человека требуется около 60 тыс. треугольников.

Триангуляция полигонов

Рассмотрим наиболее распространенные алгоритмы триангуляции полигонов. Простейшее решение задачи триангуляции состоит в расщеплении полигона вдоль некоторой хорды на два полигона и в дальнейшем рекурсивном разбиении их до ситуации, когда подлежащий триангуляции полигон является треугольником.

Данный способ применим лишь для триангуляции выпуклых полигонов. Выпуклым считается полигон, если отрезок прямой, соединяющий любые его две точки, полностью лежит во внутренней области (рис 4,а). Полигон на рисунке 4,б не является выпуклым, так как отрезок *ab* выходит за пределы многоугольника.

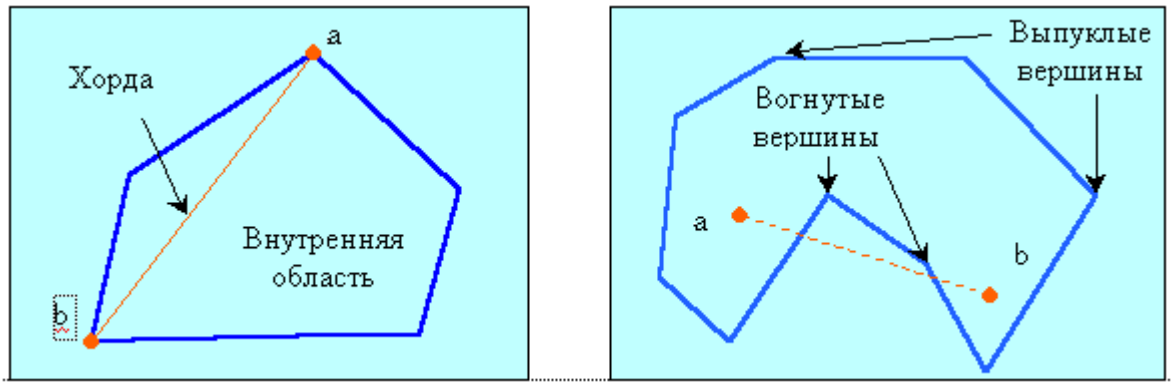

Рис. 4 а) Выпуклый полигон б) Не выпуклый полигон

Триангуляция невыпуклых полигонов не так проста, поэтому предварительная разбивка невыпуклых многоугольников на

выпуклые существенно упрощает алгоритмы их последующей обработки. Проще всего это можно сделать последовательным переносом и поворотом многоугольника так, чтобы одна из его вершин совпадала с началом координат, а исходящая из нее сторона — с осью OX. При расположении каких-либо сторон ниже оси, происходит их отсечение, и алгоритм рекурсивно повторяют для полученных полигонов, пока они не станут выпуклыми.

Приведем универсальный алгоритм. Триангуляцию любого полигона можно осуществить по следующему универсальному алгоритму.

Выбирается крайняя левая вершина и между двумя ее смежными сторонами проводится диагональ. При этом могут иметь место следующие два случая:

- **диагональ ac является хордой (рис. 5,а);**
- **диагональ ac — не хорда (рис. 5,б), так как внутрь треугольника abc попадает вершина d полигона (в общем случае их может быть несколько).**

Из всех вершин внутри треугольника abc вершина d наиболее удалена от стороны ac. Эту вершину будем называть*вторгающейся*. В случае, когда вторгающейся вершины нет, то полученный треугольник заносят в сетку треугольников, и алгоритм рекурсивно обрабатывает оставшийся полигон до тех пор, пока он не выродится в треугольник. При обнаружении вторгающейся вершины проводится диагональ из текущей до вторгающейся вершины. Полученные полигоны рекурсивно обрабатываются до получения треугольников.

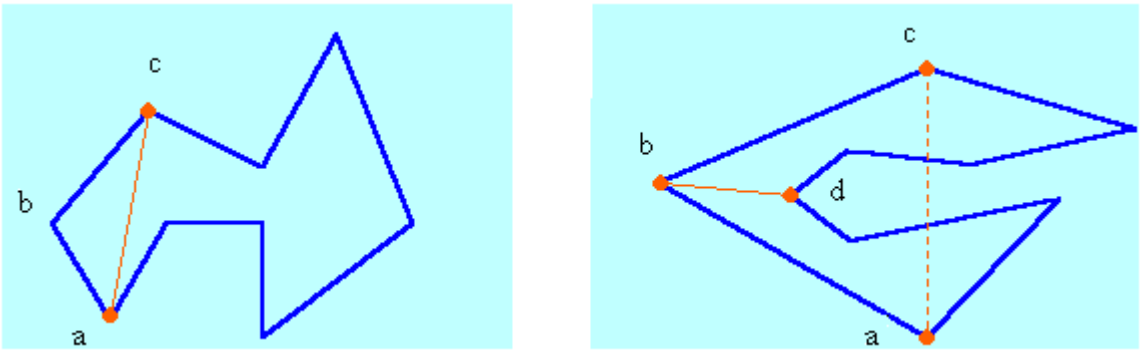

Рис. 5. а) диагональ ac — хорда б) обнаружена вторгающаяся вершина

Один из подходов к триангуляции состоит в нахождении внутренней точки области, ограниченной полигоном, и соединении с ней всех вершин (рис.6).

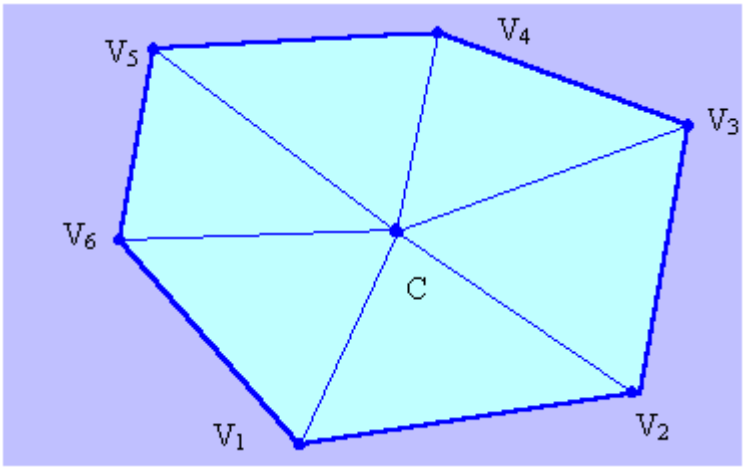

Рис. 6 Триангуляция с использованием внутренней точки

Учитывая, что триангуляция является подготовительным этапом рендеринга, к выбору внутренней точки могут предъявляться определенные требования. Так, например, при многопроцессорной обработке целесообразно внутреннюю точку выбрать таким образом, чтобы площадь составляющих треугольников была примерно одинаковой. В этом случае будет обеспечена одинаковая степень вычислительной загрузки аппаратуры.

При триангуляции в качестве исходных данных наиболее часто задаются вершины полигона. Задача состоит в том, чтобы представить этот полигон совокупностью составных непересекающихся треугольников. В одном из самых распространенных программных интерфейсов (API) для разработки приложений в области двумерной и трехмерной графики стандарте OpenGL для формирования триангуляционной сетки используются следующие команды:

- **GL_TRIANGLES. Треугольники. Команда рисует серию треугольников, используя тройки вершин v_0, v_1 и v_2, затем v_3, v_4 и v_5 и т.д (рис. 7). Если количество вершин не кратно 3, оставшиеся точки игнорируются.**

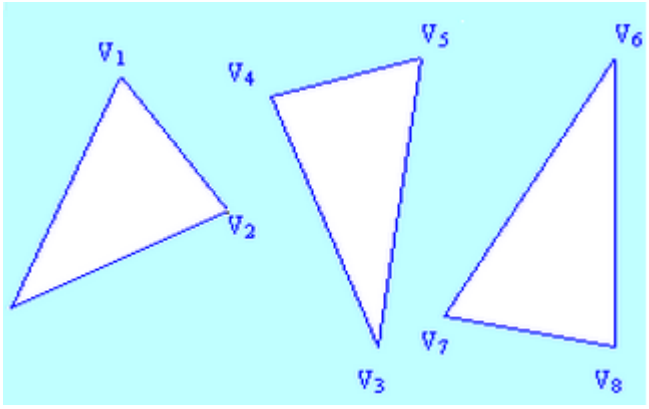

Рис.7.

Команду удобно использовать при описании поверхности, которая в результате тесселяции представляется кроме треугольников и другими геометрическими фигурами, например, прямоугольниками. В этом случае треугольники могут быть не соприкасаться между собой.

Команда может использоваться для описания всех треугольников полигона, независимо от того, связаны они или нет между собой, однако из-за дублирования информации о координатах общих вершин память используется нерационально.

- **GL_TRIANGLE_STRIP. Полоса треугольников. Команда рисует серию треугольников, используя вершины в следующем порядке: v_0, v_1, v_2, затем v_2, v_1, v_3, затем v_2, v_3, v_4, и т. д (рис. 8). При таком подходе каждая следующая вершина вместе с двумя предыдущими задает треугольник. Порядок вершин должен гарантировать, что треугольники имеют правильную ориентацию. Полоса может использоваться для формирования части поверхности.**

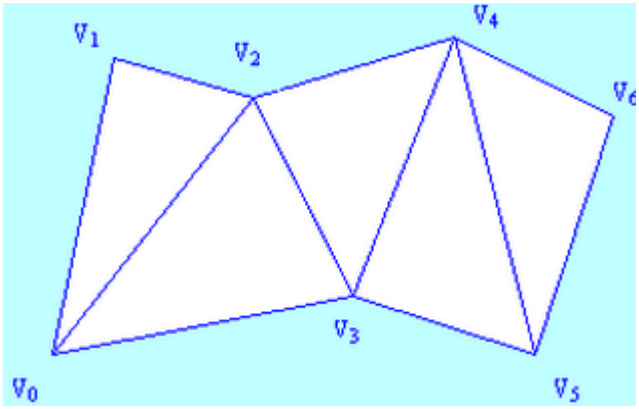

Рис.8

- **GL_TRIANGLE_FAN. Команда похожа на GL_TRIANGLE_STRIP, но при формировании треугольников используют вершины в таком порядке: v_0, v_1, v_2, затем v_0, v_2, v_3, затем v_0, v_3, v_4, и т. д (рис. 9). Все треугольники имеют общую вершину v_0. Команда позволяет разбить n-угольник на n-2 треугольника. Для этого нужно выбрать одну из n вершин полигона и соединить ее с каждой из**

  - **n-3 несоседних вершин**

Поверхности, заданные набором точек

В машинной графике задачу триангуляции можно рассматривать в двух направлениях – это триангуляция полигональных областей и триангуляции набора точек. Последняя имеет место при описании поверхности набором точек и интенсивностями их цветов.

Поточечное описание поверхностей применяют в тех случаях, когда поверхность очень сложна и не обладает гладкостью, а детальное представление многочисленных геометрических особенностей важно для практики. К поверхностям такого рода можно отнести участки грунта, формы малых небесных тел, микрообъекты, снятые с помощью электронного микроскопа и другие образования со сложной формой.

Среди методов триангуляции для конечного набора точек, которые задают поверхность, широко используют *метод Делоне*. Метод предполагает соединение между собой набора точек непересекающимися отрезками прямых линий таким образом, чтобы сформированные треугольники стремились к равноугольности (рис. 10,а). Триангулированное изображение на рисунке 10,б не может быть отнесено к триангуляции Делоне.

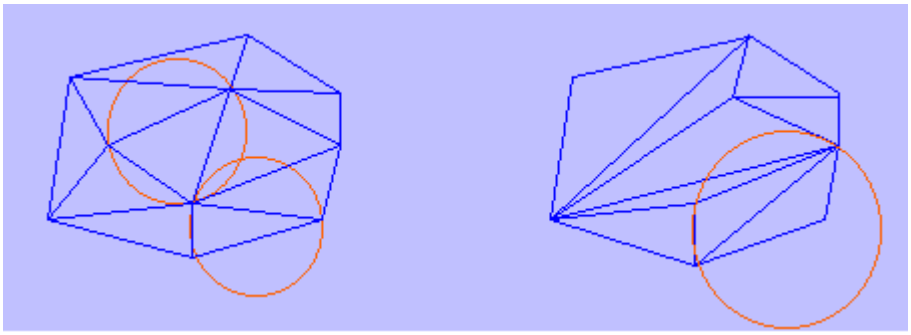

Рис. 10. Два варианта триангуляции для одного набора точек

Триангуляция набора точек будет триангуляцией Делоне, если описанная окружность для каждого треугольника будет свободна от точек, то есть внутри ее не будет больше ни одной точки из набора. На рисунке 10,а показаны две окружности, которые не содержат внутри себя других точек (можно провести окружности и для других треугольников, чтобы убедиться, что они также свободны от точек набора). На рисунке 10,б это правило не соблюдается – внутри области, ограниченной окружностью, попала одна точка другого треугольника. Следовательно, эта триангуляция не относится к типу Делоне.

Алгоритм работает путем постоянного наращивания к текущей триангуляции по одному треугольнику за шаг. Вначале триангуляция состоит из одного ребра, по окончании работы триангуляция является триангуляцией Делоне. На каждой итерации алгоритм ищет новый треугольник, который подключается к границе текущей триангуляции. Поиск и подключение треугольника образно можно представить, как надувание двумерного пузыря, привязанного к одному из ребер границы. Если такой пузырь достигает некоторой, еще не включенной в триангуляцию точки из

набора, то образуется треугольник. В случае, когда точка не встречена, обрабатывается следующее ребро границы.

Заключение

Алгоритмы триангуляции реализованы во многих профессиональных графических пакетах 3D-моделирования, таких как 3D Studio MAX, OpenGL Optimizer, LightWave и др. Эти пакеты позволяют не только использовать различные алгоритмы для триангуляции, но и добавлять пользователям свои.

Учитывая, что алгоритмы триангуляции являются неотъемлемой частью практически всех программных продуктов 3D-графики, интенсивно ведутся работы по их усовершенствованию. Можно предположить, что в недалеком будущем они будут реализованы аппаратно в графических акселераторах.

Опубликовано в журнале "КОМПЬЮТЕРЫ+ПРОГРАММЫ", № 1, 2001 г.