

ОПТИМИЗАЦИЯ ТРЕХМЕРНЫХ ОБЪЕКТОВ В СИСТЕМАХ ВИЗУАЛИЗАЦИИ РЕАЛЬНОГО ВРЕМЕНИ

Обсуждается задача оптимизации трехмерных объектов. Рассматриваются методы уменьшения количества полигонов без видимых искажений поверхностей.

Большинство геометрических примитивов могут быть исключены из списка обрабатываемых данных, если они находятся далеко от наблюдателя, не попадают в пирамиду видимости или повернуты от наблюдателя. Отбраковка не актуальных в данный момент времени объектов происходит с помощью ограничивающих сфер (невидимых оболочек), которые полностью окружают каждый объект в базе данных. Чем дальше наблюдатель находится от объекта, тем менее требуется его детальное описание. Если генератор изображений будет обрабатывать как можно меньше граней, принадлежащим далеким моделям, то его вычислительные ресурсы могут быть направлены на отображение большего числа граней для объектов с высокой степенью детальности важных моделей на переднем плане. Для этого заранее создаются несколько версий одного и того же объекта с разными уровнями детализации (LOD-levels-of-detail) [1-5].

Для самого низкого уровня детальности (LOD) объект состоит из ограниченного количества граней, для каждого более детального уровня, объект обрастает все новыми подробностями. Для этого необходимо определять размер объекта и расстояние до него, чтобы сделать оценку того, какой уровень детальности соответствует объекту, чтобы отобразить его должным образом.

Резкие изменения в сцене, в результате внезапного добавления и удаления граней плохо сказывается на качестве изображений в динамике. Сглаживание делает изменения плавными и, следовательно, менее заметными при добавлении или удалении граней. Это достигается с помощью функции прозрачности. Осуществляется переход от полностью прозрачной сцены (и, следовательно, невидимой) до непрозрачной (или до любого уровня прозрачности, который указан с основным цветом). Еще плавный переход осуществляется с помощью геометрической операции морфинга.

Для систем реального времени, одной из важнейших задач является правильный подбор объектов для оптимизации. Детали большого объекта заметнее деталей маленького к тому же расположенного в глубине сцены, а количество граней определяет необходимость оптимизации данного объекта. Исходя из этого характеристикой оптимизации выбрано отношение площади поверхности объекта к количеству граней объекта.

Площадь поверхности объекта вычисляется через площади граней, так как в трехмерных объектах грани состоят из треугольников и четырехугольников, площадь каждой отдельной грани найти не сложно.

Расчет количества граней объекта на его собственную поверхность, то есть плотность граней, берем как соотношение количества граней этого объекта на площадь

объекта. Эта характеристика объекта принимается как основная, по которой будет происходить выделение объекта для оптимизации. Объекты с большой плотностью будем оптимизировать, с маленькой плотностью можем оставить без оптимизации.

Выделение в объекте граней для объединения происходит из принципа - удаляем грани по возрастанию площади. Так как площади граней объекта у нас уже рассчитаны, то выбираем грань с минимальной площадью и объединяем ее с гранью имеющей с ней общее ребро. Полученная после преобразования грань может представлять собой шестиугольник, что абсолютно не приемлемо, поэтому удаляем вершины. Удалению будут подлежать вершины с большими внутренними углами между ребрами. Если из удаляемой вершины исходят ребра других граней, то ребра друг их граней переносятся в вершину находящуюся по часовой стрелке от удаляемой вершины. После этой процедуры площади граней с измененными ребрами пересчитываются, упорядочиваются, и оптимизация продолжается дальше.

Выводы.

Рассмотрен метод уменьшения количества обрабатываемых примитивов в зависимости от отношения площади поверхности объекта к количеству граней объекта. Данный метод позволяет снизить вычислительную нагрузку на генератор изображений без потери качества изображений.

Список литературы

1. Асмус А.Э., Богомяков А.И., Вяткин С.И. и др. Видеопроцессор компьютерной системы визуализации "Альбатрос" // Автометрия. N 6. 1994. С. 39.
2. Великохатный Р.И., Вяткин С.И., Гимаутдинов О.Ю., Чижик С.Е. и др. "Ариус" - семейство 3D графических систем реального времени для PC платформ // Тр. 7-й Междунар. конф. "Графикон-97". Москва, 1997.
3. W. Straber, A. Schilling, G. Knittel, "High Performance Graphics Architectures", // Graphicon'95 Proceedings, S. Klimenko et al. (Eds). St-Petersburg 1995.
4. K. Akeley, "Reality Engine Graphics", SIGGRAPH'93 Conference Proceedings, Computer Graphics, Vol. 27, August 1993, pp. 109-116.
5. Computer Image Generation. Edited by Bruce J.Schachter. A John Wiley&Sons, Inc. 1983.
6. Вяткин С. И. Моделирование сложных поверхностей с применением функций возмущения // Автометрия, - 2007, - т. 43, - № 3. - С. 40–47.
7. G.Knittel Voxel Engine for Real-time Visualization and Examination, Eurographics'93, - 1993. - Volume 12, - number 3, - P. 37.