

Об одной адаптивной модели представления объектов в задачах трехмерной компьютерной графики

А.Д. Капустин, Ю.Г. Федорова, Т.В. Фирсова, А. В. Чурбанов,

Нижегородская Лаборатория Программных Технологий, ЗАО, Нижний Новгород, Россия

Аннотация

В компьютерной графике существует множество подходов к построению адаптивных моделей поверхностей, которые позволяют поднять производительность отрисовки без потери качества изображения объекта. Целью нашей работы было построение программной системы, позволяющей качественно воспроизводить на компьютерах класса PC-workstation в режиме реального времени поверхности, описываемые десятками тысяч полигонов.

В реализованной адаптивной модели, как и в большинстве других, базовым представлением поверхности является триангуляция. В качестве структуры хранения множества уровней детализации выбрана multi-триангуляция (*multi-triangulation*) [1]. Для построения графа multi-триангуляция мы использовали модификации известных алгоритмов упрощения сеток поверхностей: уточнения (*refinement*) [2] и прореживания (*decimation*) [3]. На этапе извлечения триангуляции переменного разрешения из структуры графа использовались алгоритмы извлечения методом спуска [4] и с текущего сечения [4,5]. В работе описываются алгоритмы построения модели и извлечения триангуляции требуемого разрешения. Демонстрируется работа программы на нескольких практических моделях. Приводятся временные характеристики работы программы.

Ключевые слова: *упрощение сетки, уровень детализации, многомасштабная модель, multi-триангуляция.*

1. ВВЕДЕНИЕ

Отображение геометрических моделей с высокой частотой кадров (*frame-rate*) является одной из основных задач современной трехмерной графики. Для детального описания геометрии объектов в CAD, ГИС и приложениях других областей строятся модели, состоящие из десятков и сотен тысяч элементов.

Поскольку в общем случае *frame-rate* находится в обратной зависимости от числа отображаемых элементарных примитивов, значительный выигрыш в скорости получается при использовании адаптивного подхода. В этом случае, для отображения конкретного кадра используется аппроксимация объекта, содержащая меньшее число элементов, но такая, что визуальное качество изображения не ухудшается по сравнению с самым детальным представлением, либо ухудшается, но контролируемым образом. Такая дискретизация зависит от взаиморасположения объекта

и точки наблюдения. Возможно определение меры качества отображения модели и построение функциональной связи между качеством картинки и числом элементов дискретизации. При этом, очевидно, что более грубые аппроксимации, содержащие малое число элементов, будут обеспечивать высокий *frame-rate* при низком качестве изображения, в то время как более точные аппроксимации будут качественно отображать модель, но с низким *frame-rate*. Модель объекта, предоставляющая множество аппроксимаций, принято называть многомасштабной (многоуровневой) моделью¹.

Задание объектов посредством полигональных сеток их поверхностей является стандартом в современной 3D графике. Далее мы будем говорить о моделях, где в качестве полигональной сетки используется триангуляция.

Данная статья представляет программную реализацию многомасштабной модели (MRM) поверхности. Основные требования к базовым алгоритмам и программе формулировались следующим образом:

- Структура MRM должна быть адекватна для поверхностей типа *terrain* (однозначно проектируемую на некоторую плоскость) и произвольных поверхностей (*free-form*).
- Возможность построения уровней детализации поверхности со значительным уменьшением числа полигонов.
- Динамическая генерация уровней детализации в реальном времени, т.е. во время отображения.
- Гладкий, непрерывный переход между различными уровнями детализации в процессе отображения последовательных кадров.
- Возможность регулировать качество изображения.

В качестве структуры MRM нами была выбрана *multi-триангуляция* [1,5] представляющая собой ациклический направленный граф. Работа с MRM осуществляется в два этапа:

- построение MRM на базе итеративного алгоритма упрощения поверхности, поставляющего последовательные локальные изменения в

¹ Чаще чем *многомасштабная* или *многоуровневая модель*, встречается английский термин *multiresolution model* и его сокращение MRM. Синонимом *аппроксимации* в данном контексте являются термины *уровень детализации* и *уровень разрешения*, их английский эквивалент – *level of detail*.

триангуляции; этот этап, как правило, требует значительных вычислительных затрат, и поэтому выполняется в off-line;

- извлечение триангуляции переменного разрешения; этот этап выполняется в on-line для каждого фрейма.

Статья построена следующим образом. Работы по *multi-triangulation* обсуждаются в разделе 2. Кроме того там же приводятся источники, описывающие алгоритмы упрощения поверхности, выбранные нами в качестве базовых для построения MRM.

В разделе 3 описывается граф *multi-triangulation*. Его построение методом спуска на основе алгоритма аппроксимации с заданной точностью поверхности, описываемой бивариантной функцией, представлено в разделе 4. В разделе 5 мы описываем построение *multi-triangulation* методом подъема с использованием алгоритма упрощения поверхности произвольной формы. Раздел 6 посвящен алгоритму динамического извлечения триангуляции поверхности переменного разрешения из *multi-triangulation*. В разделе 7 обсуждаются результаты.

2. ОБЗОР ИСПОЛЬЗУЕМЫХ РАБОТ

Обзор работ по построению MRM и извлечению триангуляции с переменным разрешением содержится в [6].

В [5] E.Purpo вводит понятие *multi-triangulation* поверхностей. *Multi-triangulation* определена, как много разрешающая модель поверхности, основанная на наборе фрагментов плоских триангуляций, размещенных в частично упорядоченном множестве. Различные аппроксимации поверхности могут быть получены комбинированием различных фрагментов модели. Описан алгоритм извлечения из MRM триангуляции аппроксимированной поверхности с переменным разрешением. Этот алгоритм стартует всегда с самой грубой дискретизации поверхности. Расширение понятия *multi-triangulation* для поверхности произвольной формы, было введено L.De Floriani, P.Magillo и E.Purpo в [7].

В работе [1] описано два подхода к построению *multi-triangulation* MRM. Первый основывается на методе спуска (*top-down*), где построение модели начинается с грубой триангуляции поверхности, построенной на малой части исходных точек, которая модифицируется последовательным введением в нее все новых и новых вершин из исходного множества точек. Второй основывается на методе подъема (*bottom-up*), где построение поверхности начинается с самой подробной исходной сетки.

В работе [8] рассматриваются две различных MRM модели. Первая основана на обобщенной Делоне триангуляции и может быть использована для аппроксимаций поверхностей. Приведен алгоритм извлечения триангуляции поверхности с переменным разрешением для такой MRM. Вторая является

адаптацией *multi-triangulation* модели Purpo E. [5] для поверхностей произвольной формы.

В [9] строится классификация существующих MRM: иерархические (*hierarchical*) и пирамидальные (*pyramidal*) модели. Здесь же вводится обобщенная формальная модель Multiresolution Simplicial Model (MSM). Описываются основные операции над ней. Показывается, что в рамках классификации *multi-triangulation* является пирамидальной моделью.

Klein R., Kramer J. [4] приводят алгоритм извлечения триангуляции с переменным разрешением для *multi-triangulation*, обобщенной на поверхности произвольной формы. В отличие от алгоритма Purpo E. [5], который всегда стартует с самой грубой триангуляции и проходит по иерархии до тех пор, пока каждый треугольник построенной триангуляции не будет удовлетворять условию аппроксимации, определяемому пороговой функцией, алгоритм Klein и Kramer позволяет динамически улучшать или огрублять текущую триангуляцию к любому переменному разрешению.

Основной составляющей этапа построения *multi-triangulation* является процедура, осуществляющая локальные изменения в триангуляции поверхности, соответствующие последовательным уровням детализации. В качестве такой процедуры может быть выбран некий алгоритм упрощения поверхности.

Качество построенной *multi-triangulation*, а затем и извлеченных из нее триангуляций (аппроксимаций поверхности), будет зависеть от характеристик используемого алгоритма упрощения. Мы рассматриваем 2 различных алгоритма упрощения поверхности, имеющих самостоятельное значение, которые, вместе с тем, здесь позволяют строить *multi-triangulation* различными способами и для различных поверхностей.

Первый алгоритм является модификацией алгоритма жадного введения M.Garland и P.Heckbert [2]. MRM по этому алгоритму строится методом спуска. Второй алгоритм является модификацией алгоритма W. Schroeder [3]. MRM по этому алгоритму строится методом подъема. Эти алгоритмы объединяет использование структуры приоритетной очереди и структура данных триангуляции, основанная на треугольниках.

3. ОБЩАЯ СХЕМА МОДЕЛИ

Граф multi-триангуляции

В данном разделе мы дадим общее представление о том, что такое *multi-triangulation* и ее граф. Строгое определение вводится в статьях [1,5].

Multi-triangulation определяется как структура, основанная на наборе фрагментов плоских триангуляций, размещенных в частично упорядоченном множестве. Она получается последовательным применением алгоритма модификации триангуляции и запоминанием

фрагментов модификаций в некой пирамидальной структуре – направленном ациклическом графе (directed acyclic graph - DAG). Тем самым описывается взаимосвязь между локальными модификациями триангуляции.

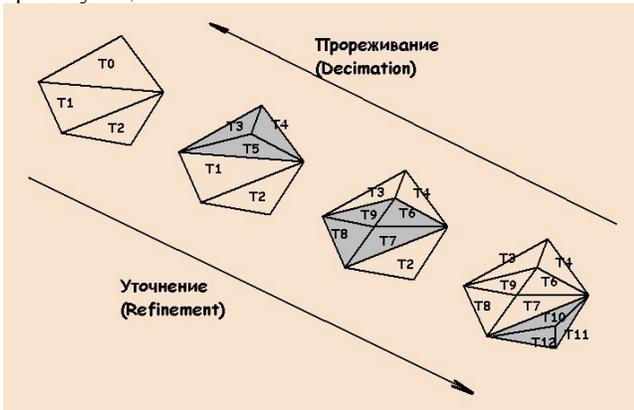


Рис. 1. Последовательность локальных модификаций триангуляции. Области модификации на каждом шаге показаны серым цветом.

Рисунок 1 показывает три шага модификации триангуляции. Граф, описывающий эту последовательность модификаций триангуляций, представлен на рис. 2.

Каждый узел q графа соответствует локальной модификации некоторого фрагмента триангуляции. В данной MRM локальной модификацией является введение в триангуляцию одной вершины v . Каждому узлу (кроме двух – см. ниже) соответствуют два набора треугольников, один определяет триангуляцию области без вершины v – треугольниками этого набора помечены дуги, входящие в узел q . Эти дуги называются дугами потолка. Второй набор определяет триангуляцию этой же области, но с вершиной v – треугольниками этого набора помечены дуги, выходящие из узла q . Эти дуги называются дугами пола. Единственным узлом DAG, в который не входит ни одна дуга, является узел-источник (*source node*), а единственным узлом, из которого не выходит ни одна дуга является узел-сток (*drain node*).

Сечением DAG называется множество дуг, содержащих точно одну дугу на каждом пути от *source* к *drain node*. Множество треугольников, соответствующих сечению, т.е. включающее в себя все треугольники, которыми помечены дуги сечения, является триангуляцией. На рис. 2 показано сечение и соответствующая ему триангуляция. В зависимости от того вносятся в сетку изменения алгоритмом уточнения или прореживания, DAG может быть построен методом спуска или методом подъема. Граф на рис. 2 был построен методом спуска. Номера дуг и узлов соответствуют порядку внесения модификаций с рис. 1.

Ниже в общих чертах будут рассмотрены алгоритмы уточнения и прореживания, являющиеся базовыми для модификации триангуляции. Затем мы

опишем построение DAG методом спуска и методом подъема (на основе прореживания).

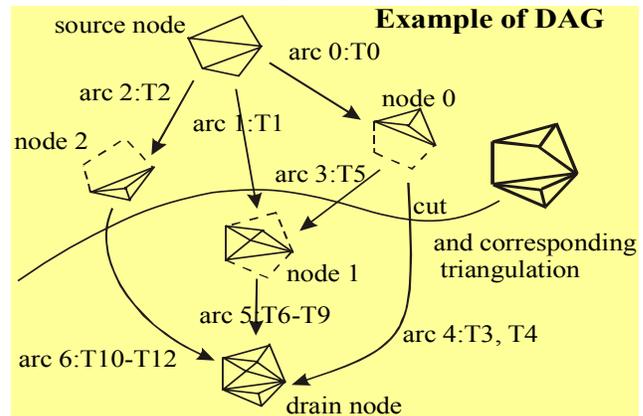


Рис. 2 Граф multi-триангуляции, построенный по последовательности модификации триангуляции показанной на Рисунке 1.

4. ПОСТРОЕНИЕ MRM НА ОСНОВЕ АЛГОРИТМА УПРОЩЕНИЯ.

Алгоритм уточнения (refinement).

Алгоритм аппроксимации поверхности с заданной точностью является модификацией алгоритма жадного введения [2]. Наша реализация работает на регулярном и рассеянном множествах начальных точек.

Алгоритм стартует с построения в плоскости X, Y выпуклой оболочки множества точек и ее триангуляции. Результат этой процедуры – *CreateInitTIN_R*, – начальная аппроксимация поверхности.

На каждой итерации – процедура *UpdateTIN_R*, выбирается исходная точка с наибольшим отклонением от текущей аппроксимации и вводится в триангуляцию. Отклонение (*triangleError*) в точке вычисляется как $|f(x, y) - h(x, y)|$, где $h(x, y)$ – значение z -координаты на аппроксимированной поверхности в точке (x, y) . Алгоритм основан на Делоне триангуляции и на каждой итерации строит аппроксимированную поверхность, проекция которой на плоскость X, Y является Делоне триангуляцией.

Быстродействие достигается за счет использования приоритетной очереди (*heap*), а также за счет хранения информации о принадлежности всех исходных точек к треугольникам текущей триангуляции. Упорядочение в очереди проводится по *triangleError*. Использование *heap* позволяет отказаться от процедуры локализации вводимой точки в текущей триангуляции. Хранение информации о принадлежности всех исходных точек текущим треугольникам позволяет быстро пересчитывать отклонения для точек, попавших во вновь образованные треугольники при ретриангуляции, и таким образом обновлять информацию о точках-кандидатах для модифицированных треугольников в области влияния (*update region*) вводимой точки.

Процесс модификации поверхности завершается либо при достижении заданного порога отклонения, либо после выполнения определенного количества итераций.

Алгоритм построения DAG методом спуска

[Инициализация DAG]

- Создать начальную триангуляцию поверхности вызвать *CreateInitTIN_R*.
- Инициализировать DAG: создать узлы *source* и *drain* и дугу между ними, помеченную всеми треугольниками начальной триангуляции.

[Введение очередного узла в DAG]

До тех пор, пока не выполнилось условие завершения процесса модификации триангуляции выполнять:

- Вызвать *UpdateTIN_R*.
- Взять множество индексов треугольников, составляющих $S \leftarrow \text{update region}$.
- Создать новый узел q_{new} , соответствующий S .
- Расщепить каждую дугу a_i , направленную в узел *drain* и помеченную хотя бы одним индексом из S , на две дуги: a_{i1} и a_{i2} . Одна из новых дуг (например, a_{i1}) будет по-прежнему направлена в *drain*, но из нее будут удалены индексы треугольников из S . Эта дуга может быть удалена, если все ее индексы принадлежат S . Другая дуга (a_{i2}) будет направлена в узел q_{new} и помечена удаленными индексами из S .
- Создать новую дугу DAG, направленную из узла q_{new} в узел *drain* и помеченную всеми треугольниками списка S .

Заметим, что в треугольниках DAG запоминаются ошибки *triangleErrors*. По ним в дальнейшем будет строиться сечение, соответствующее триангуляции требуемого разрешения (раздел 6).

5. ПОСТРОЕНИЕ MRM НА ОСНОВЕ АЛГОРИТМА УПРОЩЕНИЯ ДЛЯ ПОВЕРХНОСТИ ПРОИЗВОЛЬНОЙ ФОРМЫ

Построение MRM методом подъема реализовано нами на базе алгоритма упрощения поверхности произвольной формы последовательно удаляющим вершины триангуляции. Исходная поверхность задается произвольным набором треугольников в трехмерном пространстве, задающих, вообще говоря, несвязные поверхности, содержащие в том числе и неманифолдные сочленения.

Для работы алгоритма необходимо иметь структуру триангуляции, т.е. информацию о смежности треугольников. Если она отсутствует, то выполняется шаг доработки, восстанавливающий ее.

Алгоритм упрощения (decimation) для поверхности произвольной формы

Алгоритм упрощения является модификацией алгоритма decimation W.J.Schroeder [3]. Классификация вершин текущей треугольной сетки по топологическим и геометрическим признакам выполняется как в [3] и [10]. В случае обнаружения неманифолдного сочленения сетка разрезается на манифолдные части, как показано на рисунке 3.

После анализа исходной поверхности - процедура *CreateInitTIN_D*, на каждой итерации выполняется процедура *UpdateTIN_D*: выбирается вершина треугольной сетки с наименьшей ошибкой *vertexError* и для соответствующего ребра *edgeForCollapse*, инцидентного этой вершине, выполняется операция схлопывания ребра - *edge collapse* (рисунок 4). Аналогично алгоритму *refinement* (раздел 4) для ускорения выбора удаляемой вершины используется структура *heap*, где упорядочение проводится по ошибке в вершине (*vertexError*).

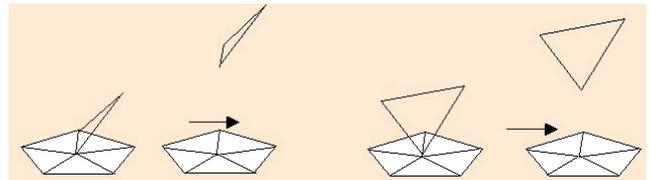


Рис. 3 Два случая неманифолдного сочленения и их «разрезание».

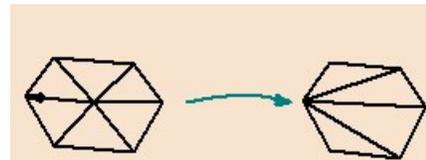


Рис. 4 Схлопывание ребра (edge collapse).

Основное отличие между алгоритмом, реализованным нами, и алгоритмом [3] в вычислении ошибки в вершине и выборе ребра для *edge collapse*. В качестве основной метрики ошибки мы выбрали усредненное расстояние между поверхностями объекта до и после удаления вершины (объем, заключенный между измененной и неизменной поверхностями, деленный на площадь измененного участка). При удалении вершины в результате *edge collapse* ошибка удаленной вершины добавляется к ошибкам всех инцидентных ей вершин.

Алгоритм построения DAG методом подъема

DAG MRM, которая строится методом подъема, не отличается от структуры, конструируемой методом спуска. Имеется разница только в схеме построения – граф строится снизу-вверх. В данном случае локальной модификацией является одна операция *edge collapse* с удалением вершины – процедура *UpdateTIN_D*. Для каждого треугольника, вводимого в DAG, *triangleError* полагается равной *vertexError* соответствующего

edgeCollapse. Также как и в DAG, построенном методом спуска, каждый узел описывает две разные триангуляции одного и того же фрагмента поверхности.

6. ИЗВЛЕЧЕНИЕ ТРИАНГУЛЯЦИИ ПЕРЕМЕННОГО РАЗРЕШЕНИЯ

Напомним, что каждое сечение DAG MRM, отделяющее узел *source* от узла *drain*, определяет некоторую триангуляцию поверхности. Для извлечения триангуляции некоторого разрешения определяется пороговая функция $\tau(P): R^3 \rightarrow R$, которая в каждой точке P пространства задает порог ошибки аппроксимации поверхности в этой точке. Треугольник T является допустимым тогда и только тогда, когда его $triangleError < \min(\tau(P))$, \min берется по всем $P \subset T$.

Пороговая функция $\tau(P)$, как правило, является параметром алгоритма извлечения. При $\tau(P) = const$ извлекается триангуляция с фиксированной ошибкой аппроксимации по всей поверхности. Определив $\tau = \tau(dist(P, viewPoint))$, где *viewPoint* точка положения наблюдателя, мы будем получать, так называемую, *view-dependent* триангуляцию, ошибка аппроксимации которой будет различаться для различных областей поверхности.

Построения сечения методом, стартующим с самой грубой аппроксимации, подробно разобрано, например, в [5], поэтому мы не будем его здесь приводить.

Опишем алгоритм извлечения триангуляции с текущего сечения. Задача состоит в том, чтобы от кадра к кадру динамически поддерживать текущее сечение минимального размера, треугольники которого имеют допустимое разрешение, минимизируя при этом само время, затрачиваемое на построение сечения. Кроме того, необходимо поддерживать гладкий переход между кадрами.

Динамическое извлечение реализуется двумя процедурами – *Refine* и *Coarsen* с использованием двух очередей *Queue0*, *Queue1*.

Сначала *Refine* добавляет к текущему сечению все дуги, расположенные ниже текущего сечения и выше результирующего сечения шага *Refine*, включая дуги результирующего сечения. Затем все дуги, не входящие в результирующее сечение, удаляются из текущего сечения, для этого выполняется их маркировка. При работе *Refine Queue0* содержит верхние узлы дуг, составляющих текущее сечение. При первом обращении к *Refine* текущее сечение инициализируется пустым списком дуг. На каждой итерации очередной узел q извлекается из очереди и тестируются треугольники дуг пола и потолка узла q . Каждая немаркированная дуга потолка маркируется и ее верхний узел добавляется к *Queue0*. Для каждой немаркированной дуги пола выполняется:

- если ее нижний узел – член *Queue0*, то дуга маркируется,

- иначе, если все треугольники дуги допустимы, то она добавляется к текущему сечению;
- иначе (если есть хотя бы один недопустимый треугольник) то дуга маркируется и ее нижний узел добавляется к *Queue0*.

Основной цикл процедуры огрубления триангуляции *Coarsen* выполняется по узлам очереди *Queue1*. *Queue1* содержит узлы DAG, у которых все дуги пола входят в текущее сечение. На каждой итерации из очереди *Queue1* извлекается очередной узел q . Если для узла q все треугольники дуг потолка допустимы, то в текущем сечении дуги пола узла q заменяются на дуги потолка и к *Queue1* добавляются те узлы, для которых все дуги пола входят в новое текущее сечение.

7. РЕЗУЛЬТАТЫ

Программная реализация всех алгоритмов была выполнена под Microsoft Visual C++ 6.0. Результаты были получены на компьютерах PentiumII/300MHz/128Mb и PentiumIII/500MHz/192Mb.

В таблице 1 приведено время работы программы для построения DAG MRM для некоторых моделей поверхностей. в off-line.

Таблица 1

Построение DAG. PentiumIII, 500MHz, 192Mb RAM

Модель	Число тр-ов в Drain Node	Время построения
Terrain 1	70000	17.3 sec
Terrain2	20000	2.5 sec
Free-form1-Horse	39700	1.5 min
Free-form2-Capitol	40000	1.66 min

При извлечении сечений нами сравнивались реализации *сверху-вниз* и *с текущего сечения*. Тесты показали что, если параметры извлечения (ошибка аппроксимации, точка наблюдения и направления взгляда) меняются плавно, то метод *с текущего сечения* работает вдвое быстрее метода *сверху-вниз*, который каждый раз стартует с самой грубой аппроксимации. Типичные времена для DAG с ~60000 в *drain node* при этом не превышали 0.05 sec на PentiumII/300MHz/128Mb при числе извлеченных треугольников <~10000. Для размера извлекаемой триангуляции ~5000 треугольников время не превышало 0.02 sec, что позволяет значительно поднять frame-rate (до 15 кадров в секунду) при использовании графических карт среднего класса. При этом расчет аппроксимации модели не превышал 20% от общего времени на кадр. Камера висит над поверхностью, направлена в центр поверхности и равномерно удаляется от нее.

На рис. 5 приведены графики зависимости времени извлечения триангуляции от положения камеры, количества треугольников в сечении и количества заметных (обработанных) треугольников для вариантов работы с треугольниками (1) и дугами (2).

Пороговая функция в этом тесте была вида $\tau = \tau(dist^2(P, viewPoint))$.

Из зависимости (1) рис.5. следует, что время извлечения текущего сечения пропорционально количеству обрабатываемых треугольников. Можно ускорить извлечение сечения, уменьшив тем самым относительную долю построения аппроксимации поверхности относительно расчета и отображения всего кадра.

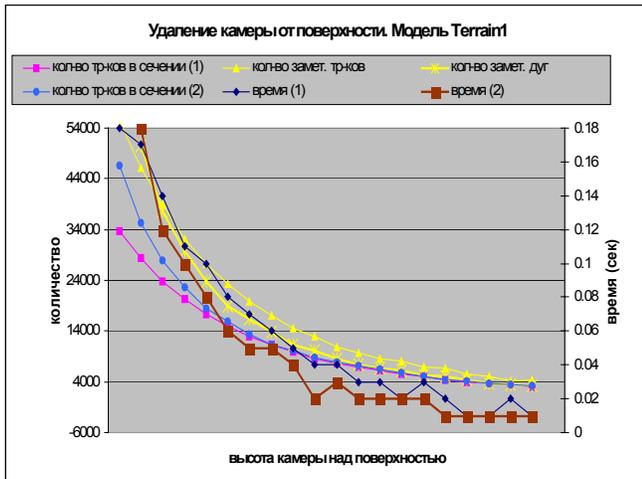


Рис. 5. PC PentiumII, 300MHz,128Mb RAM. Извлечение триангуляции для двух модификаций DAG.

Реализация (2) хранит в дуге графа ошибку равную максимальной из ошибок треугольников, маркирующих в эту дугу.

При работе с дугами извлекаемая триангуляция более подробна, что объясняется некоторой «перестраховкой» при оценке ошибки в дуге; в то же время в этом варианте меньше треугольников подвергается обработке и за счет этого извлечение быстрее. В результате за то же время ~0.05 сек стало возможным извлекать ~15000 треугольников.

На рисунках 6-10 демонстрируется работа алгоритмов на моделях Terrain1 и Horse.

8. ЗАКЛЮЧЕНИЕ И НАПРАВЛЕНИЯ ДАЛЬНЕЙШИХ РАБОТ

Представлена реализация MRM на базе *multi-триангуляции*. Показано, что на ее основе могут создаваться адаптивные модели широкого множества полигональных поверхностей с использованием различных алгоритмов упрощения сеток.

Нами модель использовалась для оптимизации визуализации. Однако, применение соответствующих предметной области алгоритмов модификации сетки, функции ошибки и пороговой функции, позволяет использовать данный подход в любых задачах, где требуется уменьшение числа элементов сетки поверхности.

С практической точки зрения, одна из важнейших задач – осуществить построение адаптивных моделей для поверхностей с числом элементов 1000000 и

больше. В этом случае хранение *multi-триангуляции* как одной структуры в оперативной памяти неприемлемо. Необходимо разработать алгоритм обработки и соответствующие структуры, которые сделают возможным хранение отдельных уровней иерархии модели во вторичной памяти их эффективную загрузку и обработку.

Литература

- [1] L.De Floriani, P.Magillo, E. Puppo, 1997, *Building and traversing a surface at variable resolution*, Proceedings IEEE Visualization'97, Phoenix, AZ
- [2] M.Garland,P.Heckbert, *Fast Triangular Approximation of Terrain and Height Fields*. Course Notes for SIGGRAPH'97 LosAngeles,1997
- [3] W.J. Schroeder, *A Topology Modifying Progressive Decimation Algorithm*, Course Notes for SIGGRAPH'97, Los Angeles, 1997
- [4] R.Klein,J.Kramer, 1997, *Building multiresolution models for fast interactive visualization*, In Proceedings of the SCCG, 1997
- [5] E.Puppo, *Variable Resolution of Terrain Surfaces*. Proceedings Eight Canadian Conf. on Comp.Geo.,pp.202-210, 1996
- [6] E.Puppo, R.Scopigno, 1997, Simplification, LOD, and multiresolution – Principles and applications, Eurographics'97 Tutorial Notes
- [7] L.De Floriani, P.Magillo, E. Puppo, *Selective Refinement of Surface Meshes: Data Structures and Algorithms*, 1998.
- [8] R.Klein,J.Kramer, 1997, *Multiresolution representations for surfaces meshes*, In Proceedings of the SCCG, pp. 57-66, 1997
- [9] L.De Floriani, P.Magillo, E. Puppo, *A Formal Approach to Multiresolution Hypersurface Modeling*.
- [10] W.J. Schroeder, J.A. Zarge, W.E. Lorensen, *Decimation of triangle meshes*, Computer Graphics 26(2): 65-70, July 1992

Об авторах

Капустин Александр (e-mail: kap@nsth.nnov.ru),
 Федорова Юлия (e-mail: juf@nsth.nnov.ru),
 Фирсова Татьяна (e-mail: tafir@nsth.nnov.ru),
 Чурбанов Андрей (e-mail: anv@nsth.nnov.ru).

AN ADAPTIVE MODEL FOR OBJECTS REPRESENTATION IN 3D COMPUTER GRAPHICS

A. Kapustin, J. Fedorova, T. Firsova, A. Churbanov

Abstract

There are a lot of approaches in computer graphics for building surface adaptive models that provide high frame rate without significant losses of object image quality. We present the program system for fast rendering of surfaces

including tens thousands polygons on PC-workstations. Triangulation is the standard representation of surface and it is used in our model also. We have taken multi-triangulation [1] (directed acyclic graph - DAG) as a structure for a keeping surface levels-of-detail hierarchy.

For DAG construction we use modifications of two well-known triangle-mesh reduction algorithms: refinement [2] and decimation [3]. To retrieve variable resolution

triangulation the “top-down” and “from-current-cut” extracting techniques are used [1,4].

In this paper we describe in details model construction and extraction algorithms. Program results are demonstrated on several practical models. We also discuss performance numbers.

Keywords: *mesh simplification, level-of-detail, multi-resolution model, multi-triangulation.*

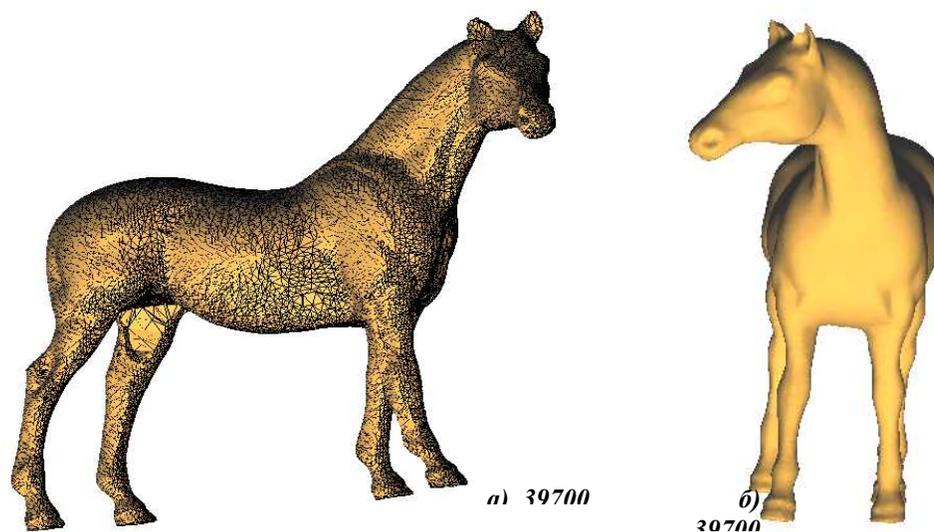


Рис.6. Модель Horse. Полное разрешение. Вид сбоку и спереди. 39700 треугольников. На а) видна сетка.

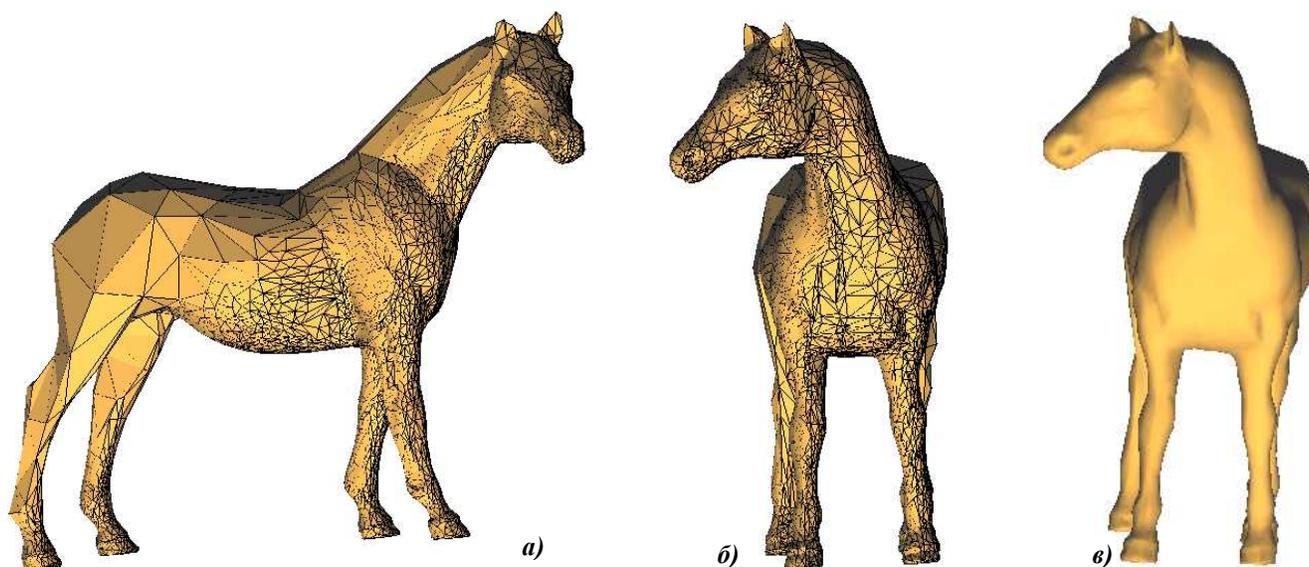


Рис. 7. Модель Horse. Аппроксимация построенная MRM для вида спереди. 8700 треугольников. Видна а),б) сетка.

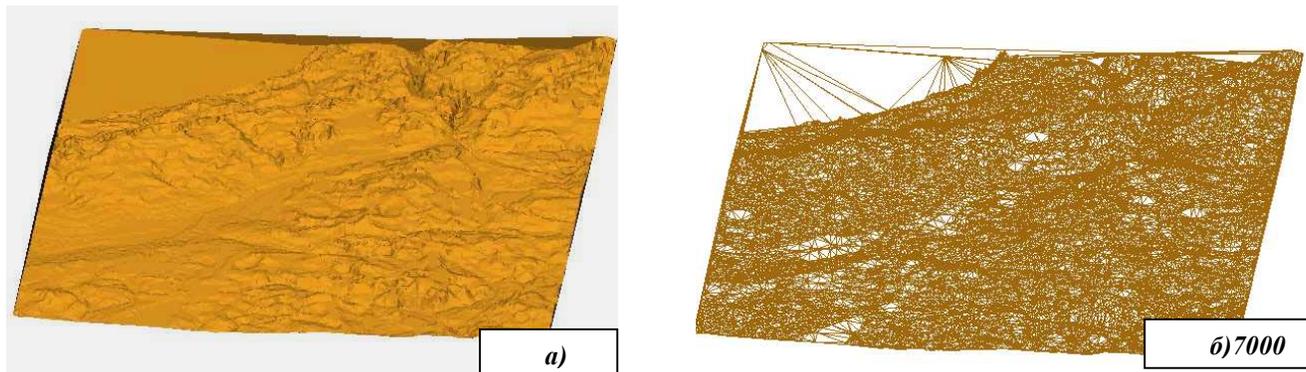


Рис.8. Модель Terrain 1. 70000 треугольников. Самое детальное разрешение.
На б) показана сетка.

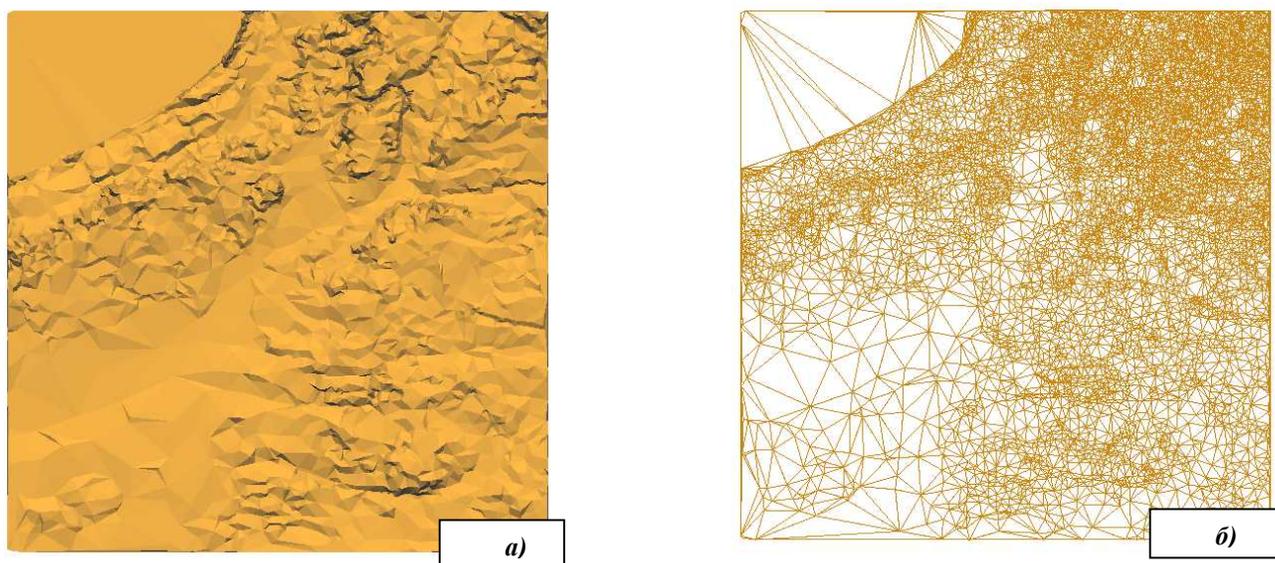


Рис 9. Модель Terrain 1. 9500 треугольников. View-dependent аппроксимация. Точка наблюдения – правый верхний угол.

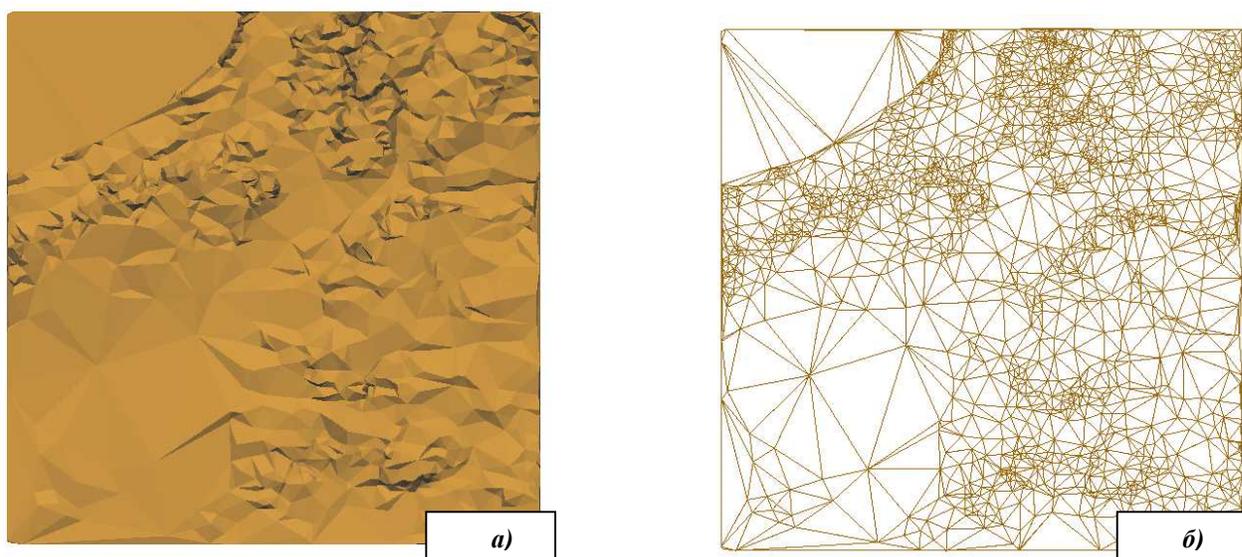


Рис 10. Модель Terrain 1. 3500 треугольников. Аппроксимация постоянного разрешения.