

УДК 681.3

## Моделирование и визуализация сложных поверхностей на основе скалярных функций возмущения

Вяткин С.И.

Институт автоматизации и электрометрии СО РАН, Новосибирск, Россия  
sivser@mail.ru

### Abstract

*The problem of real-time photorealistic imaging of complex surfaces such as digital terrain is discussed. New technique for complex surfaces by scalar perturbation functions without its approximation by polygons is considered. A recursive algorithm for object space subdivision with masking of invisible surfaces and an effective technique of projective transformation for perspective imaging are proposed. The possibility to visualize photo-realistic terrain is shown. Examples of images obtained by modeling the work of the algorithm with elevation map of defining the digital terrain are presented.*

### Введение

В машинной графике огромная вычислительная сложность алгоритмов и потенциально бесконечная сложность изображаемых моделей требуют специализированной аппаратной поддержки. Большая вычислительная стоимость синтеза фотореалистичных изображений проистекает из сложных геометрических преобразований, использования сложных моделей освещения, отображения текстуры и окружающей среды, а также методов устранения искажений, возникающих из-за дискретной природы устройств вывода изображений (устранение лестничного эффекта).

В основе систем визуализации для тренажерных комплексов (авиационных, космических, морских, автомобильных и т.п.) лежит полигональное задание объектов трехмерных сцен с последующим преобразованием и сканированием в плоскости изображений. И хотя достигнут достаточно высокий реализм отображаемых сцен в реальном времени, существует ряд задач, например, отображение больших районов местности, где требуется хранить и отображать сцены, содержащие существенно большее число полигонов, чем реализовано в современных системах. К настоящему времени разработано довольно много алгоритмов, предназначенных для построения адаптивной триангуляции рельефа путем использования иерархических структур данных (бинарных деревьев, квадродеревьев) [1-3]. Основное назначение таких алгоритмов – сократить число выводимых треугольников без существенной потери качества изображения. Однако для сложных поверхностей, каким является, к примеру, горный рельеф местности, требуется огромное число треугольников. Ни одна

из существующих архитектур не позволяет линейно масштабировать производительность систем [4-7]. Проблемы возникают всякий раз, когда запросы достигают предела производительности одного графического конвейера, в большинстве случаев из-за ограниченной пропускной способности памяти. Тогда для получения желаемого ускорения необходимо поставить параллельно несколько конвейеров и подобрать подходящий метод распределения данных. В идеальном случае производительность системы растет линейно с числом параллельных конвейеров. Однако на практике графики производительности асимптотически приближаются к некоторой величине, а увеличение производительности вообще основывается на определенных предположениях, которые могут и не выполняться для всех приложений. Для системы, отображающей многоугольники, повышение качества изображения эквивалентно повышению производительности системы, измеряемой в количестве обрабатываемых многоугольников за время одного кадра, потому что реалистичность изображения непосредственно зависит от сложности сцены и точности приближения.

Поэтому производители графических акселераторов стремятся повышать реалистичность уже не только за счет количества полигонов. Такие технологии как Bump Mapping и Environment Mapped Bump Mapping позволяют на порядок увеличить реалистичность изображения, не требуя для этого повышения количества геометрических примитивов. В настоящий момент все эти эффекты моделируются за счет специфических приемов, типа Environment Map – Bump Mapping описанный в [8] или Elevation Map, недостатками которых являются сильные ограничения на условия их применения. Например, в случае Elevation Map реализм достигается только при углах зрения близких к

прямоу. Относительно недавно появилась технология Relief Texture Mapping [9], устраняющая недостаток ограничения на угол зрения. А также новая технология Displacement Mapping (Карты смещения), которая является трехмерной, в отличие от Bump Mapping, создающей лишь иллюзию трехмерных деталей поверхности.

В данной работе приводятся результаты исследований моделирования и визуализации сложных поверхностей, частным случаем которых является рельеф местности, с применением скалярных функций возмущения. Для визуализации использовался алгоритм многоуровневого отслеживания лучей с учетом перспективы. Известные методы визуализации фотореалистичного рельефа местности очень медленны. Попытки увеличить скорость с помощью различных методов акселерации (иерархический [10], параметрический [11], кластер параллельных компьютеров [12] или специальный параллельный рэй-кэстинг, реализованный аппаратно [13], гибридный рэй-кэстинга и проекционного метода [14]) улучшают ситуацию, но этого не достаточно для достижения реального времени при визуализации фотореалистичного рельефа местности.

### Постановка задачи

Анализ возможных путей развития системы синтеза визуальной обстановки реального времени показывает, что самый простой способ увеличения качества изображения - за счет увеличения количества отображаемых в кадре многоугольников - не самый эффективный. На этом пути трудно ожидать качественного скачка. Например, такого, какой дало, в свое время, введение текстуры. Даже при многократно большем количестве отображаемых многоугольников изображение без текстуры кажется беднее, чем более простое, но - с цветной текстурой.

Основные требования к моделируемой поверхности: многоуровневость для плавной смены уровней детальности (LOD), компактное описание, гладкость и непрерывность. Используем для описания поверхности, удовлетворяющей перечисленным требованиям, скалярные функции возмущения.

Для этого предлагается следующее. Модель сложной поверхности, типа рельефа местности, кодируется дифференциальной картой высот, то есть задается алгебраически несущая поверхность и в каждом узле хранится только отклонение от этой базовой поверхности. Такой способ кодирования позволяет легко создавать непрерывные уровни детальности и облегчает качественную фильтрацию. Предлагаемый способ задания поверхностей обладает главным достоинством сплайнового

представления - высокой степенью гладкости. При использовании бикубической интерполяции поверхность, смоделированная предложенным способом, описывается функцией 3-го порядка и имеет гладкость степени  $C^2$  в рамках одного узла карты высот. Таким образом, поверхность может иметь порядок, значительно превосходящий третий, и в то же время моделироваться одной картой высот. Это эквивалентно заданию такой же поверхности большим количеством сплайн патчей, что увеличивает время геометрической обработки поверхности на величину прямо пропорциональную их количеству.

В процессе визуализации по мере рекурсивного деления вокселей на каждом уровне деления необходимо проецировать центры вокселей на некоторую плоскость. Полученные координаты, так же как и в случае с обычной RGB текстурой, определяют адрес в так называемой "карте высот" или "текстуре формы". Считываем по этому адресу для соответствующего уровня детальности соответствующую высоту и используем ее в тесте пересечения при модификации коэффициентов уравнения поверхности. В результате получаем промодулированную значениями "карты высот" гладкую поверхность, в принципе, - произвольной формы. Для плавного перехода уровней детальности и сглаживания высот необходимы те же процедуры, что и для обычной текстуры: выборка из карты четырех точек для соседних уровней детальности с последующей билинейной интерполяцией внутри каждого уровня и линейной - между уровнями.

### Геометрическая модель

Определим основные понятия и термины, которые будут использоваться в дальнейшем. Более подробно общая теория поверхностей описана в [15]. Открытое односвязное множество точек на плоскости будем называть плоской областью. Пусть  $D$  плоская область, а  $\bar{D}$  ее замыкание. Введем на плоскости координатную систему  $(u, v)$ . Пусть  $x, y, z$  прямоугольные декартовы координаты точек в трехмерном Евклидовом пространстве  $E^3$ . Зададим на множестве  $\bar{D}$  три непрерывные функции:

$$x = \varphi(u, v), \quad y = \psi(u, v), \quad z = \chi(u, v), \quad (1)$$

Предположим, что функции (1) обладают следующими свойствами. Если  $(u_1, v_1)$  и  $(u_2, v_2)$  различные точки множества  $\bar{D}$ , то точки  $M_1(x_1, y_1, z_1)$  и  $M_2(x_2, y_2, z_2)$  пространства  $E^3$ , координаты которых вычислены по формулам (1), тоже различны:

$$\begin{aligned} x_1 = \varphi(u_1, v_1), \quad y_1 = \psi(u_1, v_1), \quad z_1 = \chi(u_1, v_1), \\ x_2 = \varphi(u_2, v_2), \quad y_2 = \psi(u_2, v_2), \quad z_2 = \chi(u_2, v_2), \end{aligned} \quad (2)$$

Множество  $S$  точек  $M(x,y,z)$ , координаты  $x,y,z$  которых определяются соотношениями  $x = \varphi(u,v)$ ,  $y = \psi(u,v)$ ,  $z = \chi(u,v)$ , где функции  $\varphi, \psi, \chi$  в замыкании  $\bar{D}$  области  $D$ , обладают вышеописанным свойством, называется *простой поверхностью*.

Далее, *сложной поверхностью*  $F$  будем называть *простую поверхность*, являющуюся графиком функции, определенной в 3-х мерном пространстве  $z = f(x, y)$ . Задание сложной поверхности на основе скалярного поля, представляет собой совокупность некой *базовой поверхности*  $P$ , находящейся в той же системе координат, что и  $F$ , и *связанной* с базовой поверхностью  $P$  *карты высот*. Карта высот представляет собой двумерный *прямоугольник*, который далее будет именоваться *областью возмущения*  $D_p$  базовой поверхности  $P$ , внутри которого задана функция возмущения  $h(u,v)$ . Карта высот, в свою очередь, определяет само возмущение. Область определения функции  $h(u,v)$   $D_{h(u,v)} = \{U, V\}$ , где  $U$  и  $V$  размеры прямоугольника. Карта высот *связана* с базовой поверхностью следующим образом: существует преобразование  $G(\mathbb{R}^3 \Rightarrow \mathbb{R}^2)$  из системы координат, в которой находятся  $F$  и  $P$ , в систему координат карты высот. Как правило, таким преобразованием является параллельная проекция.

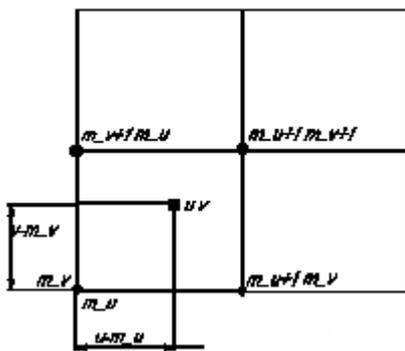


Рисунок 1 – Таблица чисел карты высот

Значение функции  $h(G(d_F))$  характеризует отклонение точки  $d_F$ , лежащей на поверхности  $F$ , от точки  $d_P$ , являющейся проекцией этой точки на поверхность  $P$ . Другими словами, значение функции  $h(G(d_F))$  равно модулю вектора

$$\vec{v} = (\vec{d}_F - \vec{d}_P) \quad (3)$$

Следовательно, область сложной поверхности можно определить как множество точек в  $\mathbb{R}^3$ , определяемых векторным уравнением.

$$\vec{F} = G(\vec{v}) + \vec{n} * h(G(\vec{v})); \forall \vec{v} \in \mathbb{R}^3, \quad (4)$$

где  $\vec{n}$  - нормаль к базовой поверхности.

Если  $\vec{v}$  находится за пределами области возмущения, то вектор  $\vec{n} * h(G(\vec{v})) = 0$ , а вектор  $\vec{F}$  является вектором на базовой поверхности.

Таким образом, для задания формы возмущающей поверхности можно использовать таблицу чисел, а в качестве функции  $h$  можно использовать функцию интерполяции по узловым значениям, которые берутся из таблицы (рис 1).

В таком случае, можно сказать, что в области возмущения  $D_p$  будет задано *скалярное поле*.

Итак, задание сложной поверхности  $F$  на основе скалярного поля (рис.2) есть: базовая поверхность, преобразование из системы координат базовой поверхности в систему координат карты высот, функция возмущения и таблица чисел, характеризующих отклонения поверхности  $F$  относительно базовой, в контрольных точках.

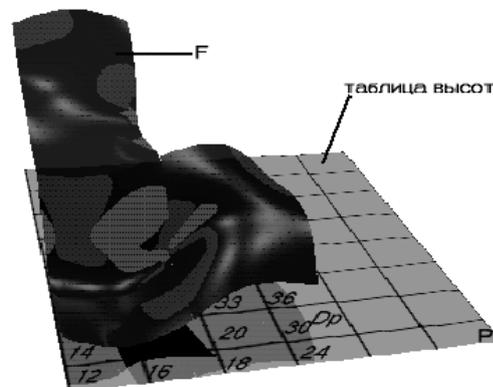


Рисунок 2 – Функция возмущения и базовая плоскость

Таким образом, мы имеем композицию базовой плоскости и возмущения  $F(x,y,z) = F(x,y,z) + R(x,y,z)$ , (5)

где  $R(x,y,z)$  - функция возмущения (карта высот). Значения функции возмущения задаются в поперечном сечении параллелепипеда двумерной таблицей высот. Направление нормали несущей (базовой) плоскости должно совпадать с продольным направлением параллелепипеда - области определения функции возмущения (рис. 3).

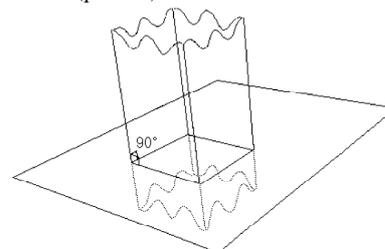


Рисунок 3 – Область определения функции возмущения

Поскольку во время растеризации необходимо оценивать максимум функции на трехмерном или одномерном интервале, то для эффективности вычислений предварительно составляются карты *уровней детальности* (рис. 4). Исходные данные составляют уровень  $n$ , если размерность сетки равна  $2^n \times 2^n$ . Данные для уровня  $n-1$  получаются выбором максимума из четырех смежных значений уровня  $n$ , остальные три далее не учитываются, т.е. получаем размерность сетки  $2^{n-1} \times 2^{n-1}$ . Уровень 0 состоит из единственного значения - максимума по всей карте высот. Таким образом, можно рассматривать карту высот как *текстуру формы*.

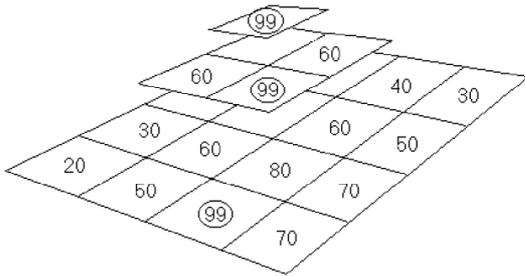


Рисунок 4 – Карты уровней детальности

При определении максимума возмущения вычисляется характерный размер проекции текущего интервала, в зависимости от которого, выбирается уровень детальности. Для большего интервала соответственно выбирается более грубое приближение исходной функции. Если требуется более точное представление, чем имеется в наличии, то производится билинейная или бикубическая интерполяция значений высот, составляющих последний уровень детальности.

### Визуализация

Для визуализации применяется метод многоуровневого отслеживания лучей [16, 17]. То есть, через каждый пиксель плоскости изображения пускается луч от взгляда наблюдателя сквозь пирамиду видимости (объектное пространство). Эти лучи проецируются на базовую плоскость. Выделение формы поверхности происходит на этапе деления луча (воксела, который ограничен передней и дальней гранями пирамиды видимости) по  $z$  координате. Одномерный брусок - воксел  $V_0$ , задается парой векторов  $P_0 = (x_0, y_0, z_0)$  и  $P_1 = (x_1, y_1, z_1)$ ,  $V_0 = \{P_0, P_1\}$ . Далее, координаты воксела  $V_0$  с помощью преобразования  $G$  преобразовываются в систему координат карты высот

$$\{(x_0, y_0, z_0), (x_1, y_1, z_1)\} \Rightarrow \{(u_0, v_0, h_0), (u_1, v_1, h_1)\} \quad (6)$$

Для этого вводится матрица  $T$  (7) преобразования в систему координат карты высот, которая, будучи умноженной, на матрицу геометрических преобразований  $M$ , дает результирующую матрицу преобразования  $G$ .  $G=T*M$ .

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

Тогда преобразованные координаты воксела  $(u, v, h, a)$  в системе координат карты высот вычисляются из  $(x, y, z)$  координат воксела в объектном пространстве умножением вектора точки в объектном пространстве на матрицу  $G$ .

$$G^* \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ h \\ a \end{bmatrix} \quad (8)$$

Далее осуществляется деление воксела по  $Z$  координате или бинарное деление воксела (рис. 5).

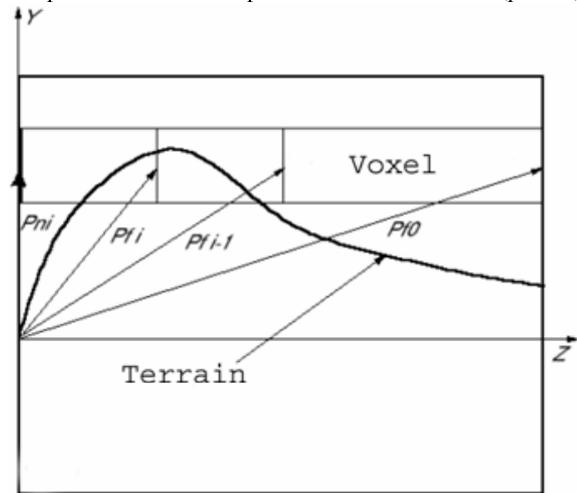


Рисунок 5 – Бинарное деление воксела

На этом этапе, для текущего уровня рекурсии вектор конца воксела являющегося ближайшим по отношению к наблюдателю, полагается равным вектору ближайшего конца воксела предыдущего уровня деления. Вектор дальнего конца воксела вычисляется как полусумма векторов ближнего и дальнего концов воксела предыдущего уровня деления.

$$P_{ni} = P_{ni-1}, P_{fi} = (P_{ni-1} + P_{fi-1})/2, V_i = \{P_{ni}, P_{fi}\} \quad (9)$$

где  $V_i$  - воксел  $i$ -го уровня рекурсии, а  $P_{ni}$  и  $P_{fi}$  координаты ближнего, по отношению к наблюдателю, и дальнего концов воксела на  $i$ -м уровне рекурсии. Таким образом, геометрические преобразования для воксела делаются всего лишь один раз, на первом уровне бинарного деления, а далее идет просто процесс деления воксела на две части и выбор ближайшего пересеченного воксела.

По размерам проекции воксела, соответствующего уровня рекурсии, вычисляется уровень детальности, который, по сути, является тоже картой высот, но меньшего разрешения и по  $u$  и  $v$  координатам точек  $P_{ni}$  и  $P_{fi}$  осуществляется выборка максимального значения из таблицы чисел, представляющих данный уровень

детальности. Полученное таким образом число и является значением функции возмущения (5) базовой поверхности.

На каждом этапе бинарного деления воксела по его размерам вычисляется уровень детальности - 'level'. Если 'level' не последний уровень детальности, то полученная высота 'h', сравнивается со значением высоты данного уровня 'Hmax', и если 'h>Hmax', то деление воксела прекращается. Таким образом, вычисления происходят в два этапа.

1. Определяем габариты прямоугольника, являющегося проекцией воксела на карту высот, как максимум расстояния от точки {u0,v0} до точки {u1,v1} -  $L_p$  (рис. 6).

2. Из неравенства  $\frac{1}{2^{level}} < L_p < \frac{1}{2^{level+1}}$  определяем уровень детальности 'level'.

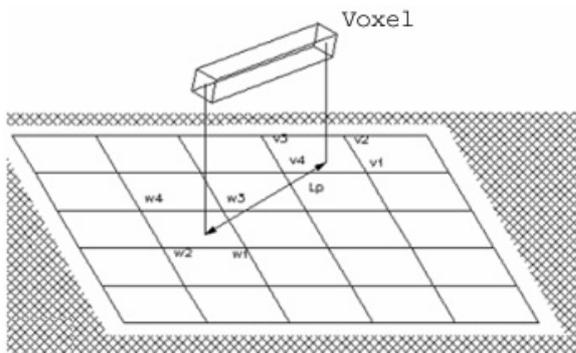


Рисунок 6 – Определение уровня детальности по проекции воксела текущего уровня

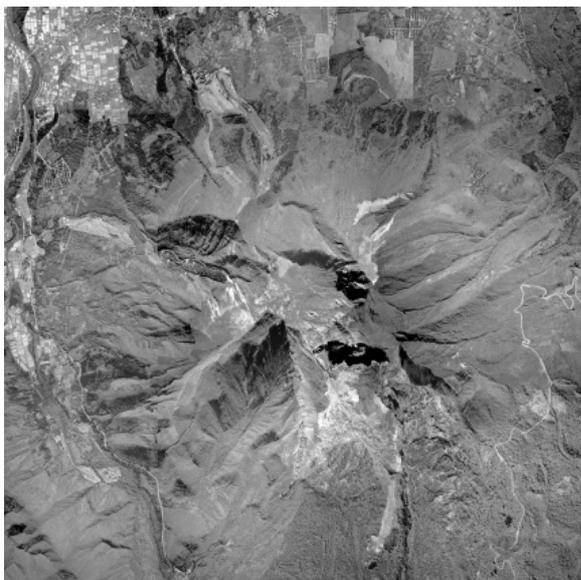


Рисунок 7 – Вид рельефа сверху

На рисунках 7 и 8 показаны результаты моделирования фотореалистичного горного рельефа местности без предварительной триангуляции с разрешением карты высот

200x200.

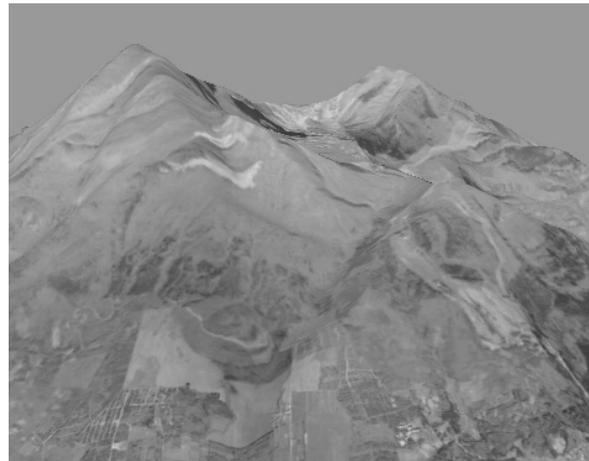


Рисунок 8 – Фронтальный вид рельефа

Проблемы, которые предлагаемый метод позволяет решить, обычным способом решаются гораздо сложнее. Действительно, обычный способ представления рельефа - треугольниками требует огромного количества граней. Кроме того, сразу возникают проблемы высокой глубинной сложности, отбрасывания задних граней, проблема приоритетов, определения и смены уровней детальности, клипирования многоугольников пирамидой видимости и пр.

В предлагаемом методе этих проблем нет. В геометрическом процессоре обрабатывается всего одна базовая плоскость. Правильный приоритетный порядок обеспечивается соответствующим обходом дерева деления и системой масок. Обратная поверхность рельефа отбрасывается автоматически. Клипирование рельефа пирамидой видимости становится ненужным, так как в процессе растривования автоматически обеспечивается выборка только нужных высот из карты высот. Для смены уровней детальности используется тот же механизм, что и для обычной текстуры.

### Заключение

Проведенные исследования визуализации сложных поверхностей, в частности фотореалистичного рельефа местности, позволили выявить ряд преимуществ, как в способе задания таких поверхностей, так и в алгоритме растривования, ориентированном на реализацию в реальном времени. К основным положительным особенностям предлагаемого подхода следует отнести:

1. Эффективность метода растривования с маскированием, сочетающего простоту вычисления с быстрым поиском и отбраковкой областей, не занятых объектами сцены.
2. Уменьшение количества данных для описания рельефа местности.

3. Снижение нагрузки на геометрический процессор и уменьшение потока данных от него в видеопроцессор.
4. Существенное упрощение построения рельефа местности обусловленное тем, что не требуется предварительная триангуляция поверхностей и клипирование пирамидой видимости. В алгоритме отображения рельефа и смены уровней детальности используется тот же механизм, что и для обычной текстуры, для фильтрации изображения при динамике используется интерполяция высот по аналогии с текстурой цвета. Время вычислений при генерации рельефа местности практически не зависит от разрешения карты высот, а зависит только от разрешения экрана (квадратичная зависимость) и от разрешения по глубине Z (логарифмическая зависимость).
5. Простота анимации поверхности (анимация текстуры формы).

В данной технологии главным, на чем она основана, является иерархический механизм управления уровнями детальности, который напоминает *MipMapping* обычной текстуры цвета. Также как и в случае текстуры цвета, время вычислений не зависит от разрешения карты высот. Если проводить полную аналогию с текстурой, то все ее положительные свойства наследуются данной технологией, к которым можно отнести, в том числе, и простоту анимации такой поверхности без вычислительных затрат на геометрические преобразования вершин треугольников, в отличие от поверхности аппроксимированной треугольниками.

В дальнейшем планируется реализация данного метода на шейдерах стандартных графических ускорителей (GPU). После его реализации станет возможным отображать сложные сцены в реальном времени.

### Литература

- [1] Markus H. Gross, Roger Gatti, and Oliver G. Staadt. *Fast multiresolution surface meshing*. In Proc. of 14th International Conf. on Data Engineering, ICDE'98, IEEE, 1998. - p.550-557.
- [2] Gross M.H., Staadt O.G., Gatti R. *Efficient Triangular Surface Approximations Using Wavelets and Quadtree Data Structures*. IEEE Trans. on Visualization and Computer Graphics. Vol. 2, N 2, June 1996. - p.130-143.
- [3] Renato Pajarola. *QuadTIN: Quadtree based Triangulated Irregular Networks*. In proceedings IEEE Visualization 2002, IEEE Computer Society Press, 2002. - p.395-402.
- [4] W. Straber, A. Schilling, G. Knittel. *High Performance Graphics Architectures*, // *Graphicon'95*

*Proceedings*, S. Klimenko et al. (Eds). St-Petersburg 1995. - p.114-121.

[5] K. Akeley. *Reality Engine Graphics*, SIGGRAPH'93 Conference Proceedings, *Computer Graphics*, Vol. 27, August 1993. - p.109-116.

[6] Evans & Sutherland. Technical Report 517902-904 AA, Evans & Sutherland Computer Corporation, USA, September 1992.

[7] Fuchs, H., J.Poulton, J.Eyles, T.Greer, J.Goldfeather, D.Ellsworth, S.Molnar, G.Turc, B.Tebbs, and L. Israel, "Pixel-Planes 5": "A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories", *Computer Graphics*, Vol. 23, N 3, 1989. - p.79-88.

[8] J.F. Blinn. *Simulation of Wrinkled Surfaces*, *Computer Graphics. Proc. SIGGRAPH 78*, Vol. 12, N 3, 1978. - p.263.

[9] Manuel M. Oliveira, Gary Bishop, David McAllister. *Relief Texture Mapping*. *Proc. SIGGRAPH 2000* (New Orleans, Louisiana, July 23-28, 2000). - p.324.

[10] D. Cohen and A. Shaked. *Photo-realistic imaging of digital terrain*. *Proceedings of Eurographics'93*, Barcelona-Spain, 6-10 September 1993, Volume 12 N 3, 6-10 September 1993. - p.363-373.

[11] D. W. Paglieroni and S. M. Petersen. *Parametric heights field ray-tracing*. *Proceedings of Graphics Interface'92*, 1992. - p.192-200.

[12] G. Vezina and P. K. Robertson. *Terrain perspectives on a massively parallel SIMD Computer*. *Proceedings of CG International'91*, Springer-Verlag, 1991. - p.163-188.

[13] P. Pitot, Y. Duthen, and R. Caubet. *A parallel architecture for ray-casting*. *Computer Graphics'89*, Blenheim, 1989. - p.463-472.

[14] G. Agranov and C. Gotsman. *Algorithms for Rendering Realistic Terrain Image Sequences and Their Parallel Implementation*. *Proceedings of Graphicon'95*, St. Petersburg, 1995. - p.153-161.

[15] Э. Позняк, Е. Шикин. *Дифференциальная геометрия*// Из-во МГУ 1990.

[16] Vyatkin S.I., Dolgovesov B.S., Yesin A.V. et al. *Voxel Volumes volume-oriented visualization system* // International Conference on Shape Modeling and Applications (March 1-4, 1999, Aizu-Wakamatsu, Japan) IEEE Computer Society, Los Alamitos, California, 1999. - p.234-241.

[17] Vyatkin S.I., Dolgovesov B.S., Guimaoutdinov O.Y. *Synthesis of virtual environment using perturbation functions* // volume III (Emergent Computing and Virtual Engineering), World Multiconference on Systemics, Cybernetics and Informatics Proceedings, Orlando, FL, USA, July 22-25, 2001. - p.350-355.

Поступила в редколлегию 03.03.2009