

Вяткин С. И.

## ВИЗУАЛИЗАЦИЯ ФОТО-РЕАЛИСТИЧНОГО РЕЛЬЕФА МЕСТНОСТИ НА ОСНОВЕ ТЕКСТУРЫ-ФОРМЫ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ УСКОРИТЕЛЕЙ

**Аннотация:** Объектом исследования являются новый способ задания и визуализация фото-реалистичного рельефа местности. Предлагается метод рендеринга террейна не сложнее отображения текстуры. Модель 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. По мере рекурсивного деления вокселей на каждом уровне деления необходимо проецировать центры вычисленных вокселей на базовую 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” поверхность. Визуализация в реальном масштабе времени реализована на графических ускорителях. Метод исследования базируется на аналитической геометрии в пространстве, дифференциальной геометрии и векторной алгебре, теории интерполяции и теории матриц, опирается на математическое моделирование и теорию вычислительных систем. Основными выводами проведенного исследования являются: возможность генерации рельефа местности, в котором для отображения рельефа и смены уровней детальности используется тот же механизм, что и для текстуры цвета. Для фильтрации изображения при динамике используется интерполяция высот по аналогии с текстурой цвета. В сравнении с известными алгоритмами в предложенном алгоритме время вычислений при генерации рельефа практически не зависит от разрешения карты высот.

**Ключевые слова:** Высотно-базируемый террейн, Воксельно-базируемый террейн, Текстура формы, Уровни детальности, Скалярные функции возмущения, Графические ускорители, Регулярная сетка высот, Нерегулярная сетка высот, Метод отслеживания лучей, Неполигональное представление рельефа

## 1. Введение

В компьютерной 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).

Визуализация террейна - сложная задача для приложений, требующих точных изображений больших массивов данных при высокой частоте смены кадров. Многие системы используют регулярные высоты местности - сетки с квадратными ячейками. Это приводит к алгоритмической простоте вычислений, однородности базы данных и строгого определения отношений между соседними уровнями детальности (LODs), что приводит к упрощению генерации базы данных. С другой стороны, регулярная сетка несет в себе избыточную информацию. Регулярная сетка может быть квадратной, шестиугольной или треугольной, и так далее. Для прямоугольной системы координат наиболее естественным является квадратная сетка со сторонами коллинеарными осям системы координат. Такая сетка называется регулярной. Большинство нерегулярных сеток имеют структуру со случайно расположенными узлами. Другие нерегулярные сетки могут иметь иные структуры. Например, генерируется неравномерность, когда квадратная сетка вращается, или смещается относительно системы координат, или имеет независимо сдвинутые узлы относительно системы координат. Некоторые свойства регулярных сеток можно использовать для вычислений нерегулярных сеток. Как правило, для достижения более высокого сжатия информации требуется дополнительная обработка и ресурсы (время, память), в том числе, и для декомпрессии. Регулярная сетка имеет фиксированную частоту дискретизации для каждого LOD. В этом случае текущий уровень детальности представляет собой набор треугольников, которыми аппроксимируется ландшафт местности, и максимальная ошибка не превышает соответствующую константу. Нерегулярная сетка имеет фиксированную полосу пропускания, и LOD с той же максимальной ошибкой может содержать меньше узлов. Здесь текущий уровень детальности представлен множеством треугольников, организованных в группы, для каждой из которых максимальная погрешность аппроксимации находится между текущим максимумом детализации и максимумом детализации следующего уровня. Каждый LOD может иметь "встроенную" (неявно) шероховатость поверхности [1] и, следовательно, объем базы данных может быть снижен. Визуализация полигонального террейна в реальном времени, и иерархическая триангуляция описаны в ряде работ [2-11].

Были разработаны многочисленные способы визуализации рельефа на основе высот [12]. В качестве баз данных для земного ландшафта можно использовать DEM (digital elevation model) модели. Этот стандарт разработан U.S. Geological Survey и, по сути, является таблицей высот земной поверхности с отсчетами через 7.5 или 15 минут. DEM

модель состоит из двух файлов: бинарного файла данных, в котором записаны высоты в виде 16-разрядных целых чисел, и файла заголовка который описывает формат записи чисел используемый в файле данных (BigEndian или SmallEndian).

В методе [13] рельеф делится на блоки, в каждом из которых выбирается уровень детальности. Для режима реального времени оптимально адаптирован алгоритм, основанный на методе [13] путем организации местности сетки высот в структуру бинарного дерева [14]. Геоморфинг непрерывных уровней детальности описан в [15]. Методика прогрессивной сетки была расширена для террейна на основе высот [16]. Алгоритм, описанный в [17] представляет еще один способ, в котором рельеф делится на предварительно обработанные блоки, и каждый блок независимо триангулирован в зависимости от точки наблюдения. Понятие текстурных клип-карт (clipmaps) [18] легли в основу создания террейна на основе карт [19, 20]. Работа [21] расширяет геометрию clipmaps так, что полосы конечной ширины вставляются между блоками для различных разрешений. Возможность использования метода отслеживания лучей на GPU для визуализации местности в режиме реального времени было описано в работе [22].

Известные методы визуализации фотореалистичного рельефа местности на базе вокселей без триангуляции очень медленные. Попытки увеличить скорость с помощью различных методов акселерации (иерархический [23], параметрический [24], кластер параллельных компьютеров [25] или специальный параллельный рэй-кэстинг, реализованный аппаратно [26], гибридный рэй-кэстинга и проекционного метода [27]) улучшают ситуацию, но этого не достаточно для достижения реального времени при визуализации фотореалистичного рельефа местности.

Для того чтобы визуализировать воксельно-базируемый террейн в реальном времени, предлагаемый метод должен быть в состоянии преобразовывать 3D-скалярное поле в набор вершин и треугольников, затем производить рендеринг с помощью графического оборудования. Способ построения сетки треугольников, вершины которых совпадают с нулевым уровнем изоповерхности - это алгоритм марширующих кубов [28]. Хотя воксельно-базируемый террейн предоставляет много больше возможностей, использование его в режиме реального времени для виртуального моделирования приводит к нескольким новым трудностям. Алгоритмы, используемые для генерации поверхности из карты вокселей, производят гораздо большее число вершин и треугольников по сравнению с террейном на основе двумерных высот. Разработка бесшовных уровней детальности для воксельно-базируемого террейна гораздо более сложная, чем аналогичная задача для местности на основе высот. Текстурирование и затенение воксельно-базируемого террейна более затратное, чем местности на основе высот. В случаях, когда сетки треугольников создаются для различных разрешений, возникает проблемы крекинга (разрывы и дырки). Способ устранения крекинга на границе между клетками, триангулированными в разных разрешениях, был описан в [29]. В работе [30] используют более простой способ, в котором вершины граней с высоким разрешением клетки перемещаются таким образом, что они лежат на одной прямой с концами соответствующими кромке на общей поверхности с низким разрешением клетки. В работе [31] подошли к проблеме устранения

крекинга с другой стороны. Они подразделяют ребра, принадлежащие клеткам с низким разрешением, и вставляют новые вершины, соответствующие местоположению вершин, принадлежащих к соседним клеткам с высоким разрешением. GPU-базируемый рэй-кастинг был предложен в [32]. Эта работа представляет собой трехмерный аналог той же идеи в двумерной области, и он обладает тем же набором преимуществ и недостатков. В частности, разрывы между уровнями детализации не проблема, но низкая производительность становится проблемой, и качество изображения снижается.

В данной работе приводятся результаты исследований визуализации фотореалистичного рельефа местности с применением скалярных функций возмущения. Для визуализации использовался алгоритм многоуровневого отслеживания лучей с применением графических ускорителей.

## 2. Методы повышения реализма изображений

Микрорельеф может быть неплохим дополнением для увеличения детализации поверхности. Основные 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 [33] 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 прямому. Повышение степени детализации ведет к сильной загрузке транспортных магистралей, и хотя вычислительных мощностей современных процессоров вполне хватает для обработки такого количества полигонов, пропускной способности шины не хватает для передачи данных. Применение текстуры решает проблему реалистичности лишь отчасти, так как вблизи объект все так же представляется как плоскость с наклеенной картинкой. Эффекты типа Bump Mapping немного улучшают картину, но и они обладают существенными ограничениями. В самой простой форме, Bump Mapping добавляет реализм к текстурам и объектам, создавая иллюзию рельефности, другими словами, изменений в поверхностной глубине, на плоской поверхности. В преобразовании выдавливания, для убедительности, глаз должен чувствовать изменения в поверхностной глубине даже притом, что поверхность действительно плоская. На визуальное восприятие глубины влияет количество света, отраженного рассматриваемой поверхностью, а количество света, отраженного в любом данном направлении, зависит от отражающей поверхности. Иными словами, гладкая поверхность всегда отражает больше чем рельефная. В трехмерной графике иллюзия рельефа основана исключительно на цвете освещения/затенения. Эффект освещения/затенения легко достигается модификацией цвета каждого из элементов текстуры, или текстурируемых пикселей, при выводе изображения на экран. Bump Mapping осуществляется следующим образом. Имеется карта рельефа

поверхности, как правило, это графический файл, и после вычисления нормали к грани, нормаль отклоняется на некий "возмущающий" вектор, значение которого вычисляется по карте рельефа. Соответственно изменяется количество отражаемого света, значит и цвет пикселя. Очевидны недостатки такой методики. Хотя поверхность трехмерной модели и кажется рельефной, на краю рельеф отсутствует. Так же, при Bump Mapping не учитывается Z координата объекта. В итоге, неровности на объекте, расположенном вблизи, и неровности на удаленном объекте имеют одинаковый размер. Технология Elevation maps, предложенная компанией Nvidia, по сути, является комбинацией Bump Mapping и геометрического моделирования рельефа, учитывающим Z- глубину. Суть метода заключается в визуализации объекта, представленного как набор параллельных друг другу слоев треугольников, и использовании альфа канала текстуры как высоты данного текселя относительно некой базовой поверхности, и канала RGB для вычисления нормали к поверхности (здесь и проявляется общность Elevation Map с Bump Mapping). Соответственно, если высота пикселя меньше высоты соответствующего пикселя в карте высот то, по значениям RGB канала данного текселя вычисляется нормаль, и затем цвет пикселя. Так как при таком подходе мы имеем текстуру "слитую" с геометрией объекта в одном примитиве, то они согласованы по глубине (Z координате) и, соответственно, лишены вышеописанных недостатков Bump Mapping. Недостатки же данного подхода - очевидны. Использование технологии Elevation Map имеет сильное ограничение на угол зрения. При угле зрения близком к нулевому, будет наблюдаться набор параллельных друг другу плоскостей. Следовательно, отобразить с высокой степенью детализации можно только те поверхности, на которые мы практически всегда будем смотреть под углами, близкими к прямому, например траву на земле. Реалистичное изображение эффектов типа волн на поверхности воды, облаков тумана, можно получить, используя технологию Environment map- Bump mapping, являющуюся еще одной модификацией bump mapping. В этом случае, помимо базовой текстуры объекта, применяются еще две текстуры. Текстура, являющаяся отрастрированным вариантом трехмерной сцены вокруг объекта (*environment map*), и текстура - карта рельефа (*bump map*). Самостоятельно и совместно с процедурными текстурами, данная технология позволяет получить такие натуральные эффекты как отражение, отражение в кривом зеркале, дрожание поверхностей, искажение изображения, вызываемое водой и теплым воздухом, трансформация искажений по шумовым алгоритмам, имитация туч на небе и другие. Технология Relief Texture Mapping [34] устраняет недостаток ограничения на угол зрения. Еще одна технология, реализованная в современных графических ускорителях это Displacement Mapping (Карты смещения), которая является трехмерной, в отличие от Bump Mapping, которая лишь иллюзия трехмерных деталей поверхности.

Резюмировать вышесказанное можно следующим образом. Повышение реализма идет уже не только за счет количества примитивов, которыми представлена сцена, а за счет моделирования специальных эффектов окружающей среды и мелких деталей поверхностей объектов сцены, причем все выше описанные технологии, применяемые в современных графических акселераторах, не решают эту задачу в полном объеме.

### 3. Постановка задачи

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. Так задача отображения горных районов местности требует для исходного описания сотни тысяч полигонов. Приведем перечень основных проблем полигональной технологии:

1. На границе силуэта проявляется многоугольная структура объекта, т. е. граница имеет форму ломаной. Следовательно полностью скрыть “граненность” тела не удастся, а в силу психологических свойств зрения, такой эффект оказывает неблагоприятное воздействие на процесс восприятия синтезированной сцены. Обычно этот дефект частично устраняется более точной аппроксимацией поверхности, что требует увеличения количества многоугольников.
2. Большое количество “мелкого мусора” – вырожденных граней, которые удалить на раннем этапе довольно трудно, необходима нетривиальная оценка малого углового размера грани. С другой стороны, при приближении наблюдателя к объекту при смене уровня детальности, необходимо добавлять новые грани, чтобы силуэт объекта не выглядел граненым.
3. Проблемы высокой глубинной сложности, при большом количестве треугольников, особенно для горного ландшафта местности.
4. Для симуляторов полёта необходим контроль высоты (HAT-height above terrain), иначе объекты, стоящие на террейне будут «зависать» над ним при смене высот и уровней детальности.
5. Проблемы крекинга при стыковке полигонов разных уровней детальности.
6. Проблемы приоритетов, определения и смены уровней детальности, клиппирования треугольников пирамидой видимости и т.д.
7. Поскольку при увеличении количества треугольников в сцене, последние имеют устойчивую тенденцию “мельчать”. Распараллеливание вычислений на уровне конкатенации практически невозможно, поскольку размер треугольников соизмерим с размером пикселя, а распараллеливание со слиянием имеет большие проблемы. В результате необходимо иметь геометрические процессоры чуть ли не на каждый пиксель.
8. Сложное распараллеливание и синхронизация процессоров, несмотря на простоту обработки многоугольников.

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.

Рельеф местности достаточной реалистичности обычно состоит из сотен тысяч примитивов (треугольников). При изменении положения наблюдателя все примитивы подвергаются геометрическим преобразованиям в режиме реального времени. Поэтому примитивы должны обладать хорошими изобразительными возможностями и отображаться за минимальное время; база данных должна быть компактной.

Основные требования к примитивам, пригодным для реалистического отображения террейна:

- многоуровневость (MIP-MAPPED TERRAIN), для плавной смены уровней детальности (LOD);
  - для компактного описания – скалярные функции возмущения;
  - поверхность и ее производные - гладкие;
  - простые алгоритмы расчета точек поверхности и ее производных.
1. Используем для описания поверхности, удовлетворяющей перечисленным требованиям, скалярные функции возмущения.
  2. Доказываем, что функция описывает поверхность без разрывов.
  3. Гладкость стыковки участков поверхности в точке определяем значениями производных функции в этой точке.
  4. Переводим рельеф из объектной системы координат (СК) в СК наблюдателя с учетом динамики наблюдателя и перспективы.
  5. Моделируем освещенность в каждой текущей точке.
  6. Моделируем текстуру в каждой текущей точке. Текстура также многоуровневая пирамидальная (texture MIP-mapping) [35].

#### 4. Геометрическая модель рельефа

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:

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

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:

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

Множество  $S$  точек  $M(x, y, z)$ , координаты  $x, y, z$  которых определяются соотношениями,  $x = \phi(u, v)$ ,  $y = \psi(u, v)$ ,  $z = \chi(u, v)$ , где функции  $\phi, \psi, \chi$  в замыкании  $\bar{D}$  области  $D$ ,

обладают вышеописанным свойством, называется *простой поверхностью*.

Далее строится *сложная поверхность*  $F$ , являющаяся 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 называется *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.

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

$$\vec{v} = (\vec{d}_F - \vec{d}_p) \tag{3}$$

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

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

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.

### 5. Функции возмущения в скалярном виде

Предлагается описывать рельеф местности, задавая функцию отклонения от базовой плоскости. То есть рельеф есть композиция базовой плоскости и возмущения

$$F'(x,y,z) = F(x,y,z) + R(x,y,z), \tag{5}$$

где  $R(x,y,z)$  - функция возмущения (карта высот) [36]. 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. 1).



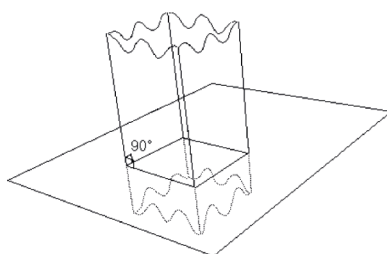


Рис. 1. Область определения функции возмущения

Поскольку во время растеризации необходимо оценивать максимум функции на трехмерном или одномерном интервале, то для эффективности вычислений предварительно составляются карты *уровней детальности* (рис. 2). 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*.

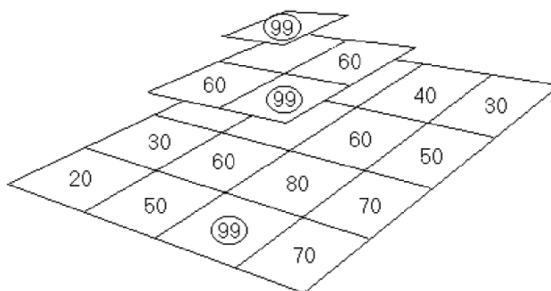


Рис. 2. Карты уровней детальности

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 интерполяция значений высот, составляющих последний уровень детальности.

## 6. Алгоритм многоуровневого отслеживания лучей

В случае скалярных функций возмущения значением возмущающей функции  $R$  является число, вычисленное по проекции данного вокселя на карту высот. Процесс растеризации такой поверхности выглядит следующим образом. Через каждый пиксель плоскости изображения пускается луч от взгляда наблюдателя сквозь пирамиду видимости (объ-

ектное пространство). Эти лучи проецируются на базовую плоскость. Выделение формы поверхности происходит на этапе деления луча (вокселя, который ограничен передней и дальней гранями пирамиды видимости) по  $z$  координате. Одномерный брусок - воксель  $V_0$  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 вокселя  $V_0$  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  $G$ .  $G=TM$ .

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

Тогда преобразованные координаты вокселя  $(u, v, h, a)$  в системе координат карты высот вычисляются из  $(x, y, z)$  координат вокселя 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 вокселя по  $Z$  координате или бинарное деление вокселя (рис. 3). На этом этапе, для текущего уровня рекурсии вектор конца вокселя являющегося ближайшим по отношению к наблюдателю, полагается равным вектору ближайшего конца вокселя предыдущего уровня деления. Вектор дальнего конца вокселя вычисляется как полусумма векторов ближнего и дальнего концов вокселя 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  $V_i$  - воксель  $i$ -го уровня рекурсии, а  $P_{ni}, P_{fi}$  координаты ближнего, по отношению к наблюдателю, и дальнего концов вокселя на  $i$ -м уровне рекурсии. Таким образом, геометрические преобразования для вокселя делаются всего лишь один раз, на первом уровне бинарного деления, а далее идет просто процесс деления вокселя на две части и выбор ближайшего пересеченного вокселя.

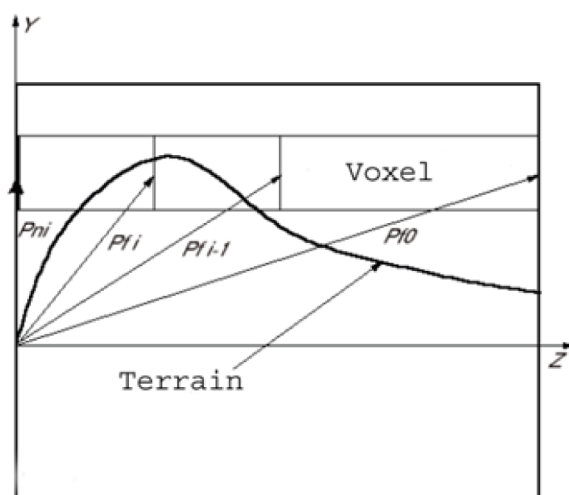


Рис. 3. Бинарное деление вокселя

По размерам проекции вокселя, 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 вокселя 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 вокселя прекращается. Таким образом, вычисления происходят в два этапа.

1. Определяются габариты прямоугольника, являющегося проекцией вокселя на карту высот, как максимум расстояния от точки  $\{u_0, v_0\}$  до точки  $\{u_1, v_1\}$  -  $L_p$  (рис. 4).
2. Из неравенства  $\frac{1}{2^{level}} < L_p < \frac{1}{2^{level+1}}$  определяется уровень детальности 'level'.

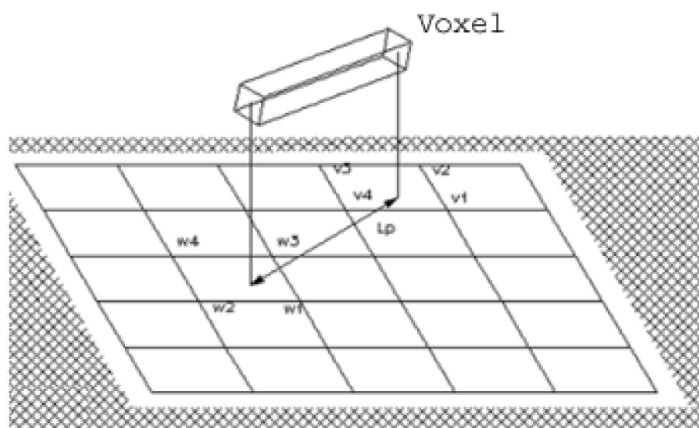


Рис. 4. Определение уровня детальности по проекции вокселя текущего уровня

## 7. Анализ метода и результаты работы

Таким образом, метод представляет собой гибрид террейна на основе высот и воксельно-базируемых методов, с наследованием положительных свойств и тех и других. Проблемы, которые предлагаемый метод позволяет решить, обычным способом решаются гораздо сложнее. Действительно, обычный способ представления рельефа - гранями требует огромного количества граней. Кроме того, сразу возникают проблемы высокой глубинной сложности, отбрасывания задних граней, проблема приоритетов, определения и смены уровней детальности, клиппирования 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. Клиппирование рельефа пирамидой видимости становится ненужным, так как в процессе растривания автоматически обеспечивается выборка только нужных высот из карты высот. Для смены уровней детальности используется тот же механизм, что и для обычной текстуры.

В сравнении с полигональным способом, предложенный метод обладает рядом достоинств. Использование билинейной интерполяции позволяет снизить объем данных, представляющих террейн. Но даже в случае описания поверхности, имеющей  $n \cdot n$  вершин, ее представление на базе полигонов требует  $3n^2$  вещественных чисел необходимых для описания вершин и  $6(n-1)^2$  целых чисел, необходимых для описания треугольников. В то время как задание такой же поверхности на базе текстуры формы требует  $n^2$  вещественных чисел необходимых для описания возмущения. Если учесть, что размер одного вещественного числа равен размеру 2-х целых чисел, то коэффициент сжатия размера исходных данных  $k = (3n^2 + 3(n-1)^2) / n^2 \approx 6$ . Если же террейн не имеет резких перепадов высот, то в предположении, что использование билинейной интерполяции позволяет использовать карту высот размерности меньшей исходной в  $l$  раз, то коэффициент сжатия  $(n = l \cdot m) \quad k = (3(l \cdot m)^2 + 3(1(m-1))^2) / m^2 \approx 6l^2$ .

Поскольку пропускная способность (количество инструкций/операций, выполняемых за единицу времени) для данной задачи важнее, чем задержка (latency - время, затрачиваемое на выполнение одной инструкции/операции), применение GPU является оптимальным решением.

Для реализации была использована Compute Unified Device Architecture (CUDA) от компании NVIDIA. CUDA - это модель параллельного программирования вместе с набором программных средств, которая позволяет реализовывать программы на языке C для исполнения на графическом акселераторе. Используется большое количество вычислительных процессоров, для одновременного вычисления множества лучей, как это было сделано в работе [37]. При реализации учитывалось влияние скорости работы с памятью. Максимально использованы регистры, потому что это самый быстрый вид доступной памяти. Для эффективного использования GPU было использовано максимально возможное количество потоков, которые выполняют полезную работу.

Для отображения изображения использовалась DirectX. Тестирование производилось

на процессоре Intel Core i7-2700K, GTX 550Ti и GTX 750 Ti в разрешении 1920x1080 пикселей. В таблице приведены результаты тестирования рельефа размером 50x80 км, с пространственной частотой DEM - 1 м и текстуры – 10 см. Как видно из тестов, время вычислений при генерации рельефа местности практически не зависит от разрешения карты высот.

На рисунках 5 и 6 показан результат моделирования фотореалистичного рельефа местности без предварительной триангуляции с разрешением карты высот 1024x1024. На рисунке 7 показан фронтальный вид горной местности с разрешением карты высот 256x256.

Разрешение карты высот	i7-2700K	GTX 550Ti	GTX 750 Ti
256x256	802,65 мс	67,03 мс	31,07 мс
512x512	850,81 мс	71,05 мс	32,93 мс
1024x1024	856,52 мс	71,53 мс	33,15 мс



Рис. 5. Фрагмент 1 рельефа на основе текстуры формы: карта высот 1024x1024



Рис. 6. Фрагмент 2 рельефа на основе текстуры формы: карта высот 1024x1024

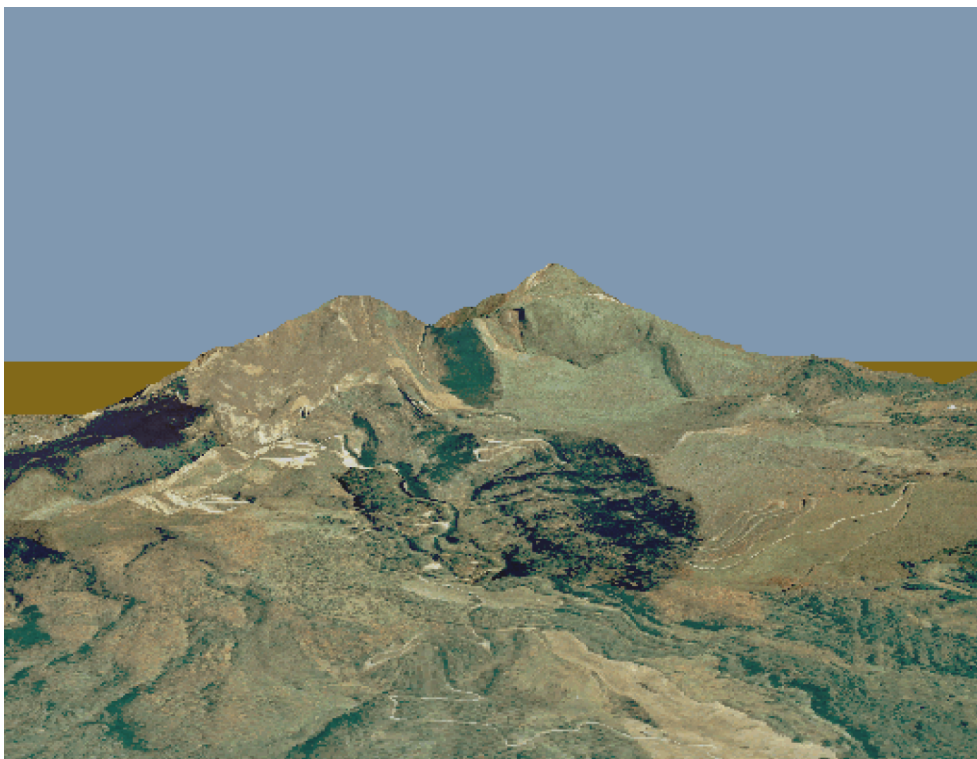


Рис. 7. Горная местность на основе текстуры формы: карта высот 256x256

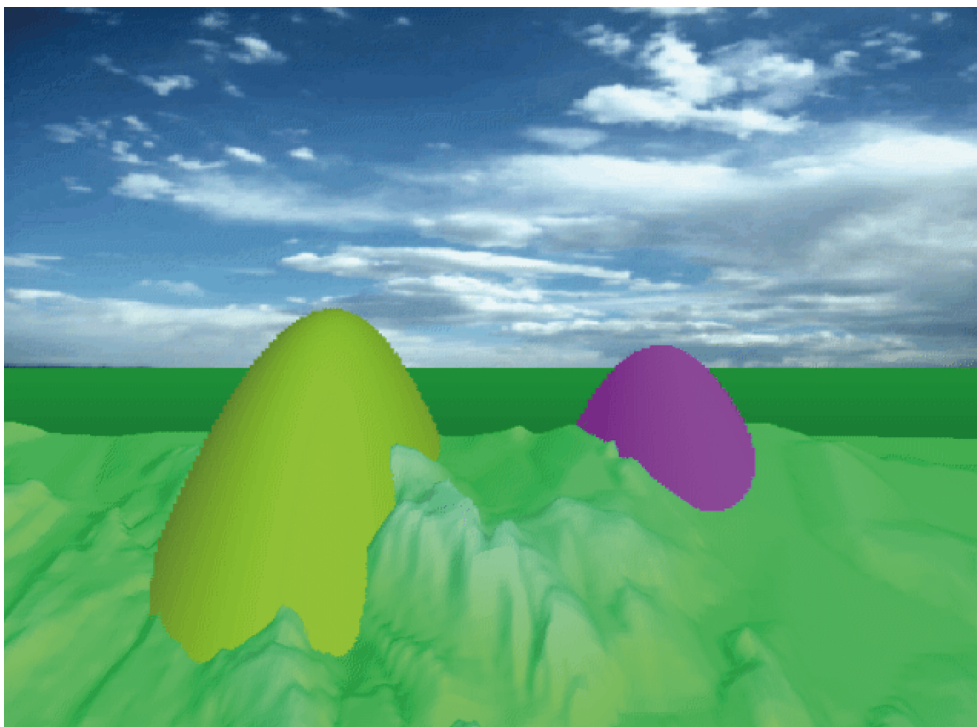


Рис. 8. Комбинированная сцена: террейн на основе текстуры формы и квадрики (карта высот 64x64)

Анализ возможных путей развития систем визуализации 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.

### 8. Заключение

Проведенные исследования визуализации фотореалистичного рельефа местности позволили выявить ряд преимуществ, как в способе задания рельефа, так и в алгоритме растривания, ориентированных на реализацию в реальном времени. Переход от растривания в плоскости изображения к растриванию в объеме, в сочетании с предлагаемыми способами задания фотореалистичного рельефа местности, хотя и в целом повышает объем вычислений, выполняемых в реальном времени, тем не менее, приводит к ряду положительных моментов, повышающих реализм отображаемых сцен. К основным положительным особенностям предлагаемого подхода следует отнести: эффективность метода растривания с маскированием; уменьшение количества данных для описания рельефа местности; снижение нагрузки на CPU и уменьшение потока данных от него в GPU; существенное упрощение построения рельефа местности обусловленное тем, что не требуется предварительная триангуляция поверхностей и клиппирование 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). Предлагаемый метод позволяет визуализировать, расположенные на нем трехмерные объекты, без необходимости контроля высоты полёта (рис. 8).

В данной технологии главным, на чем она основана, является иерархический механизм управления уровнями детальности, который напоминает *MipMapping* обычной текстуры цвета. Если проводить полную аналогию с текстурой, то все ее положительные свойства наследуются предлагаемой 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.

**Библиография :**

1. R. Ferguson, R. Economy, W. Kelly, and P. Ramos. "Continuous Terrain Level of Detail for Visual Simulation", Proceedings of Image V Conference, June, 1990. pp. 144-151.
2. L.L. Scarlatos "A Refined Triangulation Hierarchy for Multiple Levels of Terrain Detail", Proceedings of Image V Conference, June, 1990. pp. 115-122.
3. L.L. Scarlatos. Adaptive Terrain Models for Real-Time Simulation, Proceedings of the Digital Electronic Terrain Board Symposium, Wichita, KS, October 1989, pp. 219-229.
4. L.L. Scarlatos. A Compact Terrain Model Based On Critical Topographic Features, Proceedings of Auto Carto 9, Baltimore, MD, April 1989, pp. 146-155.
5. L. Scarlatos and T. Pavlidis. Adaptive Hierarchical Triangulation, Proceedings of Auto-Carto 10, Baltimore, MD, March 1991, pp. 234-246.
6. L. Scarlatos and T. Pavlidis. Hierarchical Triangulation Using Terrain Features, Proceedings of Visualization '90, San Francisco, CA, October 1990, pp. 168-175.
7. L.L. Scarlatos. An Automated Critical Line Detector for Digital Elevation Matrices, Technical Papers of 1990 ACSM-ASPRS Annual Convention, Volume 2, Denver, CO, March 1990, pp. 43-52.
8. L. Scarlatos and T. Pavlidis. Adaptive Hierarchical Triangulation, Proceedings of Auto-Carto 10, Baltimore, MD, March 1991, pp. 234-246.
9. L. Scarlatos and T. Pavlidis. Hierarchical Triangulation Using Cartographic Coherence, CVGIP: Graphical Models and Image Processing, 54 (2), March 1992, pp. 147-161.
10. L.L. Scarlatos and T. Pavlidis. "Real Time Manipulation of 3D Terrain Models", 1993 ACSM/ASPRS Annual Convention & Exposition Technical Papers, Volume 3, New Orleans, LA, February 1993, pp. 331-339.
11. L.L. Scarlatos and T. Pavlidis. Techniques for Merging Raster and Vector Features with 3D Terrain Models in Real Time, 1993ACSM/ASPRS Annual Convention & Exposition Technical Papers, Volume 1, New Orleans, LA, February 1993, pp. 372-381.
12. R. Pajarola, and E. Gobbetti. Survey on Semi-Regular Multiresolution Models for Interactive Terrain Rendering. The Visual Computer: International Journal of Computer Graphics, Volume 23, Number 8, 2007, pp. 583–605.
13. P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hodges, F. Nick, and G. A. Turner. Real-Time, Continuous Level of Detail Rendering of Height Fields. Proceedings of SIGGRAPH 1996, pp. 109–118.
14. M. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Alrich, and M. B. Mineev-Weinstein. ROAMing Terrain: Real-time Optimally Adapting Meshes. Proceedings of the 8th Conference on Visualization '97, pp. 81–88.
15. S. Rottger, W. Heidrich, P. Slusallek, and H-P. Seidel. Real-Time Generation of Continuous Levels of Detail for Height Fields. Proceedings of WSCG '98, pp. 315–322.
16. H. Hoppe. Smooth View-Dependent Level-of-Detail Control and its Application to Terrain Rendering. Proceedings of the Conference on Visualization '98, pp. 35–42.
17. Thatcher Ulrich. Super-size it! Scaling up to Massive Virtual Worlds". SIGGRAPH 2002, Course Notes. New-York: ACM press, 2002.
18. C. C. Tanner, C. J. Migdal, and M. T. Jones. The Clipmap: A Virtual Mipmap. Proceedings of SIGGRAPH 1998, pp. 151–158.



19. F. Losasso, and H. Hoppe. Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids. ACM Transactions on Graphics Proceedings of SIGGRAPH 2004, Volume 23, Issue 3, 769–776.
20. A. Asirvatham, and H. Hoppe. Terrain Rendering Using GPU-Based Geometry Clipmaps. GPU Gems 2, Addison-Wesley, Chapter 2, 2005, pp. 27–45.
21. Y. Livny, Z. Kogan, and J. El-Sana. Seamless Patches for GPU-Based Terrain Rendering. The Visual Computer: International Journal of Computer Graphics, Volume 25, Number 3, 2009, pp. 197–208.
22. C. Dick, J. Kruger, and R. Westermann. GPU Ray-Casting for Scalable Terrain Rendering. Proceedings of Eurographics 2009, pp. 43–50.
23. D. Cohen and A. Shaked. Photo-realistic imaging of digital terrain. Proceedings of Eurographics'93, Barcelona-Spain, 6-10 September 1993, Volume 12 Number 3, pages 363-373.
24. D. W. Paglioni and S. M. Petersen. Parametric heights field ray-tracing. Proceedings of Graphics Interface'92, 1992, pages 192-200.
25. G. Vezina and P. K. Robertson. Terrain perspectives on a massively parallel SIMD Computer. Proceedings of CG International'91, Springer-Verlag, 1991, pages 163-188.
26. P. Pitot, Y. Duthen, and R. Caubet. A parallel architecture for ray-casting. Computer Graphics'89, pages 463-472. Blenheim, 1989.
27. G. Agranov and C. Gotsman. Algorithms for Rendering Realistic Terrain Image Sequences and Their Parallel Implementation. Proceedings of Graphicon'95, pages 153-161, St. Petersburg, 1995.
28. W. E. Lorensen, and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm". Computer Graphics, Proceedings of SIGGRAPH 87, Volume 21, Issue 4, pp. 163–169.
29. R. Shu, C. Zhou, and M. S. Kankanhalli. Adaptive Marching Cubes. The Visual Computer, Volume 11, pp. 202–217.
30. R. Shekhar, E. Fayyad, R. Yagel, and J. F. Cornhill. Adaptive Marching Cubes". The Visual Computer, Volume 11, 1996, pp. 202–217.
31. M. Kazhdan, A. Klein, K. Dalal, and H. Hoppe. Unconstrained Isosurface Extraction on Arbitrary Octrees. Proceedings of the 5th Eurographics Symposium on Geometry Processing, pp. 125–133.
32. J.F. Blinn. Simulation of Wrinkled Surfaces, Computer Graphics. Proc. SIGGRAPH 78, Vol. 12, No. 3, P.263.
33. C. Crassin, F. Neyret, S. Lefebvre, and E. Eisemann. GigaVoxels: Ray-guided Streaming for Efficient and Detailed Voxel Rendering. Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games, pp. 15–22.
34. Manuel M. Oliveira, Gary Bishop, David McAllister. Relief Texture Mapping. Proc. SIGGRAPH 2000 (New Orleans, Louisiana, July 23-28, 2000). P. 324.
35. Lance Williams. "Pyramidal Parametrics, Computer Graphics", 1983-12, N 4, P. 270.
36. Вяткин С. И. Моделирование сложных поверхностей с применением функций возмущения // Автометрия, 2007, т. 43, № 3. С. 40–47.
37. Вяткин С. И. Метод бинарного поиска элементов изображения функционально заданных объектов с применением графических акселераторов // Автометрия, Том 50, Номер 6, 2014, С. 89-96.

**References:**

1. R. Ferguson, R. Economy, W. Kelly, and P. Ramos. "Continuous Terrain Level of Detail for Visual Simulation", Proceedings of Image V Conference, June, 1990. pp. 144-151.
2. L.L. Scarlatos "A Refined Triangulation Hierarchy for Multiple Levels of Terrain Detail", Proceedings of Image V Conference, June, 1990. pp. 115-122.
3. L.L. Scarlatos. Adaptive Terrain Models for Real-Time Simulation, Proceedings of the Digital Electronic Terrain Board Symposium, Wichita, KS, October 1989, pp. 219-229.
4. L.L. Scarlatos. A Compact Terrain Model Based On Critical Topographic Features, Proceedings of Auto Carto 9, Baltimore, MD, April 1989, pp. 146-155.
5. L. Scarlatos and T. Pavlidis. Adaptive Hierarchical Triangulation, Proceedings of Auto-Carto 10, Baltimore, MD, March 1991, pp. 234-246.
6. L. Scarlatos and T. Pavlidis. Hierarchical Triangulation Using Terrain Features, Proceedings of Visualization '90, San Francisco, CA, October 1990, pp. 168-175.
7. L.L. Scarlatos. An Automated Critical Line Detector for Digital Elevation Matrices, Technical Papers of 1990 ACSM-ASPRS Annual Convention, Volume 2, Denver, CO, March 1990, pp. 43-52.
8. L. Scarlatos and T. Pavlidis. Adaptive Hierarchical Triangulation, Proceedings of Auto-Carto 10, Baltimore, MD, March 1991, pp. 234-246.
9. L. Scarlatos and T. Pavlidis. Hierarchical Triangulation Using Cartographic Coherence, CVGIP: Graphical Models and Image Processing, 54 (2), March 1992, pp. 147-161.
10. L.L. Scarlatos and T. Pavlidis. "Real Time Manipulation of 3D Terrain Models", 1993 ACSM/ASPRS Annual Convention & Exposition Technical Papers, Volume 3, New Orleans, LA, February 1993, pp. 331-339.
11. L.L. Scarlatos and T. Pavlidis. Techniques for Merging Raster and Vector Features with 3D Terrain Models in Real Time, 1993ACSM/ASPRS Annual Convention & Exposition Technical Papers, Volume 1, New Orleans, LA, February 1993, pp. 372-381.
12. R. Pajarola, and E. Gobbetti. Survey on Semi-Regular Multiresolution Models for Interactive Terrain Rendering. The Visual Computer: International Journal of Computer Graphics, Volume 23, Number 8, 2007, pp. 583–605.
13. P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hodges, F. Nick, and G. A. Turner. Real-Time, Continuous Level of Detail Rendering of Height Fields. Proceedings of SIGGRAPH 1996, pp. 109–118.
14. M. Duchaineau, M. Wolinsky, D. E. Sigi, M. C. Miller, C. Alrich, and M. B. Mineev-Weinstein. ROAMing Terrain: Real-time Optimally Adapting Meshes. Proceedings of the 8th Conference on Visualization '97, pp. 81–88.
15. S. Rottger, W. Heidrich, P. Slusallek, and H-P. Seidel. Real-Time Generation of Continuous Levels of Detail for Height Fields. Proceedings of WSCG '98, pp. 315–322.
16. H. Hoppe. Smooth View-Dependent Level-of-Detail Control and its Application to Terrain Rendering. Proceedings of the Conference on Visualization '98, pp. 35–42.
17. Thatcher Ulrich. Super-size it! Scaling up to Massive Virtual Worlds'. SIGGRAPH 2002, Course Notes. New-York: ACM press, 2002.
18. C. C. Tanner, C. J. Migdal, and M. T. Jones. The Clipmap: A Virtual Mipmap. Proceedings of SIGGRAPH 1998, pp. 151–158.

19. F. Losasso, and H. Hoppe. Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids. ACM Transactions on Graphics Proceedings of SIGGRAPH 2004, Volume 23, Issue 3, 769–776.
20. A. Asirvatham, and H. Hoppe. Terrain Rendering Using GPU-Based Geometry Clipmaps. GPU Gems 2, Addison-Wesley, Chapter 2, 2005, pp. 27–45.
21. Y. Livny, Z. Kogan, and J. El-Sana. Seamless Patches for GPU-Based Terrain Rendering. The Visual Computer: International Journal of Computer Graphics, Volume 25, Number 3, 2009, pp. 197–208.
22. C. Dick, J. Kruger, and R. Westermann. GPU Ray-Casting for Scalable Terrain Rendering. Proceedings of Eurographics 2009, pp. 43–50.
23. D. Cohen and A. Shaked. Photo-realistic imaging of digital terrain. Proceedings of Eurographics'93, Barcelona-Spain, 6-10 September 1993, Volume 12 Number 3, pages 363-373.
24. D. W. Paglioni and S. M. Petersen. Parametric heights field ray-tracing. Proceedings of Graphics Interface'92, 1992, pages 192-200.
25. G. Vezina and P. K. Robertson. Terrain perspectives on a massively parallel SIMD Computer. Proceedings of CG International'91, Springer-Verlag, 1991, pages 163-188.
26. P. Pitot, Y. Duthen, and R. Caubet. A parallel architecture for ray-casting. Computer Graphics'89, pages 463-472. Blenheim, 1989.
27. G. Agranov and C. Gotsman. Algorithms for Rendering Realistic Terrain Image Sequences and Their Parallel Implementation. Proceedings of Graphicon'95, pages 153-161, St. Petersburg, 1995.
28. W. E. Lorensen, and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm". Computer Graphics, Proceedings of SIGGRAPH 87, Volume 21, Issue 4, pp. 163–169.
29. R. Shu, C. Zhou, and M. S. Kankanhalli. Adaptive Marching Cubes. The Visual Computer, Volume 11, pp. 202–217.
30. R. Shekhar, E. Fayyad, R. Yagel, and J. F. Cornhill. Adaptive Marching Cubes". The Visual Computer, Volume 11, 1996, pp. 202–217.
31. M. Kazhdan, A. Klein, K. Dalal, and H. Hoppe. Unconstrained Isosurface Extraction on Arbitrary Octrees. Proceedings of the 5th Eurographics Symposium on Geometry Processing, pp. 125–133.
32. J.F. Blinn. Simulation of Wrinkled Surfaces, Computer Graphics. Proc. SIGGRAPH 78, Vol. 12, No. 3, P.263.
33. C. Crassin, F. Neyret, S. Lefebvre, and E. Eisemann. GigaVoxels: Ray-guided Streaming for Efficient and Detailed Voxel Rendering. Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games, pp. 15–22.
34. Manuel M. Oliveira, Gary Bishop, David McAllister. Relief Texture Mapping. Proc. SIGGRAPH 2000 (New Orleans, Louisiana, July 23-28, 2000). P. 324.
35. Lance Williams. "Pyramidal Parametrics, Computer Graphics", 1983-12, N 4, P. 270.
36. Vyatkin S. I. Modelirovanie slozhnykh poverkhnostei s primeneniem funktsii vozmushcheniya // Avtometriya, 2007, t. 43, № 3. С. 40–47. .
37. Vyatkin S. I. Metod binarnogo poiska elementov izobrazheniya funktsional'no zadannykh ob"ektov s primeneniem graficheskikh akseleratorov // Avtometriya, Tom 50, Nomer 6, 2014, S. 89-96.