

АНАЛИЗ И СИНТЕЗ СИГНАЛОВ И ИЗОБРАЖЕНИЙ

УДК 681.3.19

КОМБИНИРОВАННЫЙ МЕТОД ВИЗУАЛИЗАЦИИ ФУНКЦИОНАЛЬНО ЗАДАННЫХ ПОВЕРХНОСТЕЙ И ТРЁХМЕРНЫХ ТЕКСТУР

© С. И. Вяткин, Б. С. Долговесов

*Институт автоматизи и электрометрии СО РАН,
630090, г. Новосибирск, просп. Академика Коптюга, 1
E-mail: sivser@mail.ru
bsd@iae.nsk.su*

Предложен комбинированный метод визуализации объектов на основе аналитических и скалярных функций возмущения и трёхмерных текстур с применением графических процессоров. Для отображения рельефа и смены уровней детальности используется тот же механизм, что и для текстуры цвета, а для рассеянного света — вершинные шейдеры. Описывается метод визуализации объёмных облаков в реальном времени. Для этого предлагается формировать трёхмерные текстуры с использованием предварительной обработки структуры облака и объёмно-ориентированной визуализации.

Ключевые слова: функции возмущения, трёхмерная текстура, объёмные облака, рассеянный свет, объёмно-ориентированная визуализация, графические процессоры.

DOI: 10.15372/AUT20190209

Введение. В компьютерной графике огромная вычислительная сложность алгоритмов и потенциально бесконечная структурированная сложность отображаемых сцен требуют специализированной аппаратной поддержки. Высокая вычислительная стоимость синтеза реалистичных изображений вытекает из геометрических преобразований, использования сложных моделей освещения, отображения текстуры и окружающей среды. Многие функциональные возможности современных графических процессоров берут своё начало из алгоритмов рендеринга, созданных для высокопроизводительных графических рабочих станций и систем визуализации реального времени для различных видов тренажёров. Наиболее распространённая модель для визуализации проекционных изображений пространственных объектов — полигональное приближение. Наряду со множеством преимуществ такая модель имеет и недостатки. Моделирование реальных объектов формирует приближённую полигональную модель. Для хорошего качества изображения чаще всего необходимо увеличивать количество полигонов, что влечёт за собой повышение времени визуализации и объёма используемой памяти. К тому же дополнительные проблемы вносит изменение масштаба объекта, поскольку нельзя быстро и эффективно изменить количество полигонов для модели объекта. При сканировании двумерного пространства не получится полноценное трёхмерное изображение. Полигональная графика со сканированием полигонов в плоскости изображения не является трёхмерной. Информация, которая предоставляется пользователю в такой технологии, неполная. Один из главных недостатков — это отсутствие информации о глубине объекта (т. е. информации о луче, проходящем сквозь объект). Кроме этого, для визуализации сложных объектов очень актуальны геометрические преобразования: унарные, бинарные и другие геометрические операции, которые наиболее оптимально решаются в рамках твердотельного моделирования и конструктивной геометрии для функциональных моделей, например теоретико-множественные операции, с помощью которых можно конструировать объекты и их композиции неограниченной

сложности. Достигается это в первую очередь применением булевых операций объединения и пересечения [1]. Процесс моделирования становится значительно эффективнее, если использовать функционально заданные примитивы и объёмно-ориентированную визуализацию.

Предлагается альтернативная технология визуализации, основанная на методе отслеживания лучей и функциональном задании примитивов, в отличие от традиционной технологии отображения полигональных сцен, построенной на методе рендеринга треугольников [2–5]. Принципиальное отличие предлагаемого метода от остальных заключается в неполигональном представлении поверхностей и использовании алгоритма отслеживания лучей для растеризации сцены. Механизм рендеринга отличается от реализованных технологий в современных ускорителях компьютерной графики. Объекты представляются функционально, что обеспечивает компактность задания баз данных. Использование алгоритма отслеживания лучей устраняет недостатки, характерные для систем с растрированием на плоскости, и проблемы больших объёмов данных.

В ряде работ [6–10] показаны функциональные способы описания моделей, позволяющие в сравнении с полигональным заданием существенно сократить объём баз данных для определённого класса объектов. Однако отображение в реальном времени объектов, заданных таким способом, связано с существенным ростом требуемых вычислений из-за высокого порядка описываемых функций. Отметим следующие функциональные способы задания примитивов. Поверхности свёртки [7–12] — это интегральное представление неявно заданных поверхностей, известных в компьютерной графике как капельные модели [10], метасферы [11], мягкие объекты [12]. Данные поверхности сочетают в себе гибкость капельных моделей и компактность скелетных моделей [13]. В то же время они имеют недостатки, к которым можно отнести сложность вычисления точек поверхности.

Цель данного исследования заключается в разработке комбинированного метода визуализации объектов на основе аналитических [2] и скалярных функций возмущения [2, 14], вычисления рассеянного света в атмосфере с использованием вершинных шейдеров графических процессоров [15] и рендеринга объёмных облаков на основе трёхмерных текстур.

Объекты на основе аналитических функций возмущения. Предлагается описывать сложные геометрические объекты, задавая функцию отклонения (второго порядка) от базовой квадратики в виде

$$F(x, y, z) = A_{11}x^2 + A_{22}y^2 + A_{33}z^2 + A_{12}xy + A_{13}xz + A_{23}yz + A_{14}x + A_{24}y + A_{34}z + A_{44} \geq 0. \quad (1)$$

На базе квадратик строятся свободные формы, которые представляют собой композицию базовой квадратики и возмущений [2]:

$$F'(x, y, z) = F(x, y, z) + \sum_{i=1}^N f_i R_i(x, y, z), \quad (2)$$

где f_i — формфактор, $R(x, y, z)$ — возмущение:

$$R_i(x, y, z) = \begin{cases} Q_i^3(x, y, z), & \text{если } Q_i(x, y, z) \geq 0, \\ 0, & \text{если } Q_i(x, y, z) < 0, \end{cases} \quad (3)$$

($Q(x, y, z)$ — возмущающая квадратика).

Через i и N (2) обозначаются пределы суммирования (нижний и верхний соответственно). В качестве нижнего предела суммирования применяется единица (одна функция возмущения), а в качестве верхнего предела — несколько функций возмущения (зависит от сложности формы).

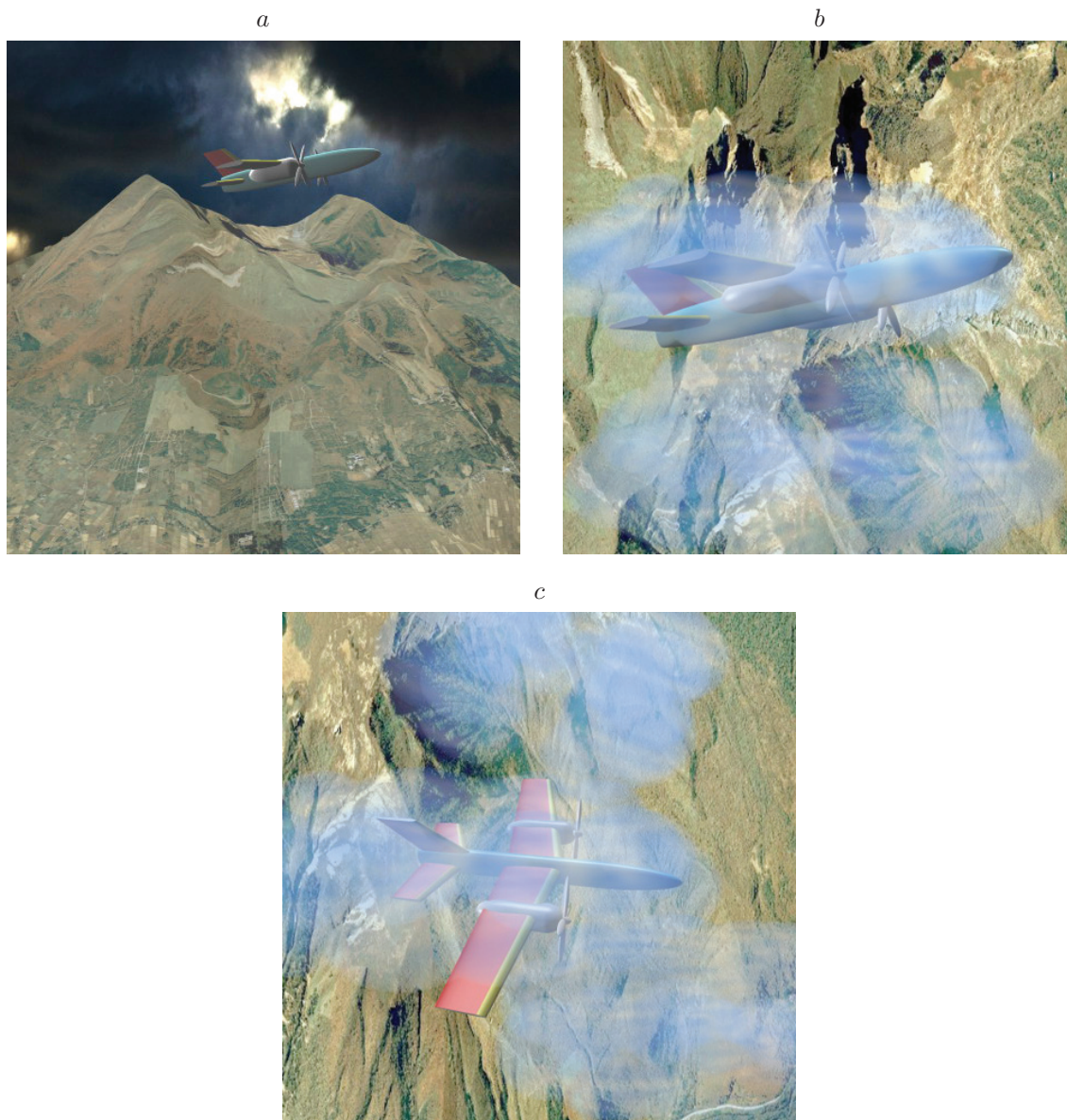


Рис. 1. Изображение объекта, заданного аналитическими функциями возмущения (турбовинтовой самолёт), и воксельно-базируемого рельефа местности: *a* — в рассеянном свете в атмосфере, *b* — в объёмных облаках (вид сбоку), *c* — в объёмных облаках (вид сверху)

Полученная поверхность будет гладкой, и потребуется небольшое количество функций возмущения для создания сложных форм поверхностей, количество данных сокращается более чем в 100 раз в сравнении с полигональным описанием (рис. 1) [16]. На рис. 1 модель самолёта создана на базе функций возмущения и геометрических операций объединения [1].

Рельеф местности на основе скалярных функций возмущения. Предлагается описывать рельеф местности, задавая функцию отклонения от базовой плоскости [14]. Значения функции возмущения задаются в поперечном сечении параллелепипеда двумерной таблицей высот. Направление нормали базовой плоскости должно совпадать с продольным направлением параллелепипеда — областью определения функции возмущения. Для эффективности вычислений предварительно составляются карты уровней детальности.

Исходные данные составляют уровень n , если разрешение сетки равно $2^n \times 2^n$. Данные для уровня $n - 1$ получаются выбором максимума из четырёх смежных значений уровня n , остальные три далее не учитываются, т. е. получаем разрешение 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.

При определении максимума возмущения вычисляется характерный размер проекции текущего интервала, в зависимости от которого выбирается уровень детальности. Для большего интервала соответственно берётся 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.

Для визуализации применяется метод отслеживания лучей [14], т. е. 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 задаётся 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 переходят в систему координат карты высот:

$$\{(x_0, y_0, z_0), (x_1, y_1, z_1)\} \Rightarrow \{(u_0, v_0, h_0), (u_1, v_1, h_1)\}. \quad (4)$$

Для этого вводится матрица T преобразования в систему координат карты высот, которая, будучи умноженной на матрицу геометрических преобразований M , даёт результирующую матрицу преобразования $G = TM$:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

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:

$$G \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ h \\ a \end{bmatrix}. \quad (6)$$

Далее осуществляется деление воксела по z -координате или бинарное деление воксела. На этом этапе для текущего уровня рекурсии вектор конца воксела, являющегося ближайшим по отношению к наблюдателю, равен вектору ближайшего конца воксела предыдущего уровня деления. Вектор дальнего конца воксела вычисляется как полусумма векторов ближнего и дальнего концов воксела предыдущего уровня деления:

$$P_{ni} = (P_{ni-1} + P_{fi-1})/2; \quad V_i = \{P_{ni}, P_{fi}\}, \quad (7)$$

где V_i — воксел i -го уровня рекурсии, а P_{ni}, P_{fi} — координаты ближнего по отношению к наблюдателю и дальнего концов воксела на i -м уровне рекурсии. Таким образом, геометрические преобразования для воксела выполняются всего лишь один раз на первом

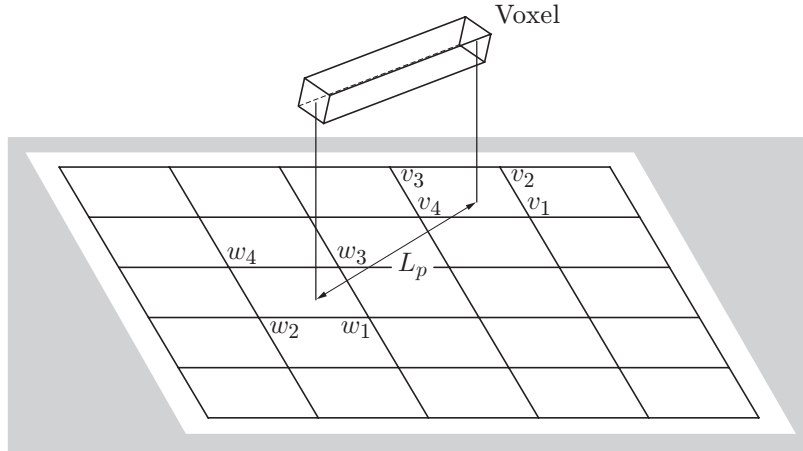


Рис. 2. Определение уровня детальности по проекции вокселя текущего уровня

уровне бинарного деления, а далее идёт процесс деления вокселя на две части и выбор ближайшего пересечённого вокселя.

Воксел — это параллелепипед (рис. 2), у которого выбираются габаритные вершины при проецировании на плоскость. Проекция будет представлять собой многоугольник. По максимальным размерам проекции берётся уровень детальности, что является важным условием для определения правильного уровня детальности карты высот.

По размерам проекции вокселя и соответствующего уровня рекурсии вычисляется уровень детальности, который, по сути, является тоже картой высот, но меньшего разрешения. По координатам u и v точек P_{ni} и P_{fi} осуществляется выборка максимального значения из таблицы чисел, представляющих данный уровень детальности. Полученное таким образом число является значением функции возмущения базовой поверхности.

На каждом этапе бинарного деления вокселя по его размерам вычисляется уровень детальности $level$. Если это не последний уровень, то полученная высота h сравнивается со значением высоты уровня H_{\max} , и если $h > H_{\max}$, 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 .

2. Из неравенства $2^{-level} < L_p < 2^{-(level+1)}$ определяется $level$.

Открытое односвязное множество точек на плоскости будем называть плоской областью. Пусть D — плоская область, а \bar{D} — её замыкание. Введём на плоскости координатную систему (u, v) . Пусть x, y, z — прямоугольные декартовы координаты точек в трёхмерном евклидовом пространстве E^3 . Зададим на множестве \bar{D} три непрерывные функции:

$$x = \varphi(u, v); \quad y = \psi(u, v); \quad z = \chi(u, v). \quad (8)$$

Предположим, что функции (8) 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 (8), тоже различны:

$$\begin{aligned} x_1 &= \varphi(u_1, v_1); & y_1 &= \psi(u_1, v_1); & z_1 &= \chi(u_1, v_1); \\ x_2 &= \varphi(u_2, v_2); & y_2 &= \psi(u_2, v_2); & z_2 &= \chi(u_2, v_2). \end{aligned} \quad (9)$$

Множество K 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 φ, ψ, χ в замыкании \bar{D} области D обладают вышеописанным свойством, называется простой поверхностью.

Далее строится сложная поверхность F , являющаяся графиком функции, определённой в трёхмерном пространстве $z = f(x, y)$. Задание сложной поверхности на основе скалярного поля представляет собой совокупность некоей 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 $G (R^3 \Rightarrow R^2)$ 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

$$\mathbf{v} = (\mathbf{d}_F - \mathbf{d}_P). \quad (10)$$

Следовательно, область сложной поверхности можно задать как множество точек в R^3 , определяемых векторным уравнением

$$\mathbf{F} = G(\mathbf{v}) + \mathbf{n}h(G(\mathbf{v})); \quad \forall \mathbf{v} \in R^3, \quad (11)$$

где \mathbf{n} — нормаль к базовой поверхности.

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

Таким образом, для задания формы возмущающей поверхности можно применять таблицу чисел, а в качестве функции h можно использовать функцию интерполяции по узловым значениям, которые берутся из таблицы. В таком случае можно утверждать, что в области возмущения D_P будет задано скалярное поле.

Функция h имеет следующий вид:

$$h(u, v) = f_0 + (f_1 - f_0)(v - m_v), \quad (12)$$

где $f_0 = (1 - (u - m_u) \text{table}[m_u, m_v]) + (u - m_u) \text{table}[m_u + 1, m_v]$; $f_1 = (1 - (u - m_u) \text{table}[m_u, m_v + 1]) + (u - m_u) \text{table}[m_u + 1, m_v + 1]$; m_u — целая часть u , m_v — целая часть v , $\text{table}[m_u, m_v]$ — (m_u, m_v) -й элемент таблицы.

Итак, задание сложной поверхности F на основе скалярного поля 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 поверхности в контрольных точках.

В данной работе рассматривается задание поверхностей свободных форм на основе плоскостей. В этом случае преобразованием G является параллельная проекция в направлении, противоположном вектору нормали базовой плоскости. Понятие поверхности свободной формы F используется как комбинация плоскости и функции возмущения, которая может быть поверхностью с прямоугольным контуром или поверхностью, определяемой векторным уравнением (11).

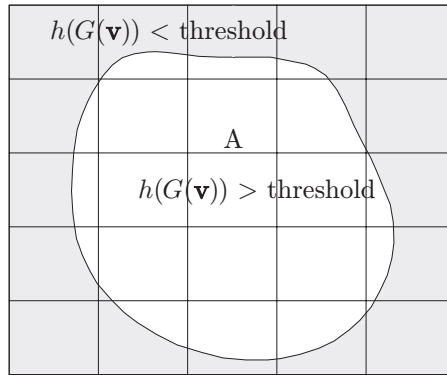


Рис. 3. Область возмущения

В этом случае способ построения поверхностей был реализован таким образом, что стало возможным отображать объекты с нелинейной границей (рис. 3). Для этого введены понятие порогового значения, находящегося в узле таблицы, и критерий принадлежности точки пространства [14]:

$$\mathbf{F} = G(\mathbf{v}) + \mathbf{n}h(G(\mathbf{v})), \quad h(G(\mathbf{v})) \geq \text{threshold}, \quad G(\mathbf{v}) \in D_p, \quad \forall \mathbf{v} \in R^3, \quad (13)$$

где $\text{threshold} = \text{const} \in R$ — пороговое значение.

Введение требования принадлежности образа вектора \mathbf{v} к области возмущения D_p позволяет отсекалть части пространства, которые отображаются в зону, находящуюся вне области возмущения. По критерию $h(G(\mathbf{v})) \geq \text{threshold}$ отсекаются элементы пространства, образ которых лежит в области возмущения, но которые не принадлежат моделируемому объекту. Область в карте высот, удовлетворяющая этим требованиям, определяет в пространстве возмущения многосвязную область, используемую при моделировании объекта (см. рис. 3).

На рис. 1 модель рельефа местности создана на базе скалярных функций возмущения.

Вычисление рассеянного света и тумана. В работе [15] был предложен метод вычисления рассеянного света и тумана на этапе аппаратных трансформации и освещения в графическом процессоре.

Туман в графических процессорах вычисляется с помощью двух методов: пиксельного (табличного) и вершинного.

В пиксельном методе цвета смешиваются с цветом тумана. Коэффициенты смешивания берутся отдельно для каждого пиксела по первым битам Z -буфера из предварительно вычисленной таблицы. Вершинный метод более прогрессивный. Он работает на этапе Transform & Lighting, определяя коэффициенты для каждой вершины, а затем интерполируя их, как и в обычном освещении. В данной работе применён вершинный метод.

Вычисляются две составляющие света: компонента освещённости и компонента тумана. Интенсивность света первого источника зависит от высоты верхнего слоя тумана (рис. 4) и составляющей окружающей среды A . Компонента освещённости объекта вычисляется по следующей формуле [17]:

$$I = A e^{-h \operatorname{cosec}(\alpha B)} + I_c e^{-h \operatorname{cosec}(\alpha B) - BH \operatorname{cosec} \beta} R, \quad (14)$$

где I_c — интенсивность, которая вычисляется 1 раз по достижении светом границы тумана; R — угловой коэффициент зеркальности; A — коэффициент окружающей среды; B — коэффициент затухания (зависит от угла рассеяния света μ), который несёт информацию

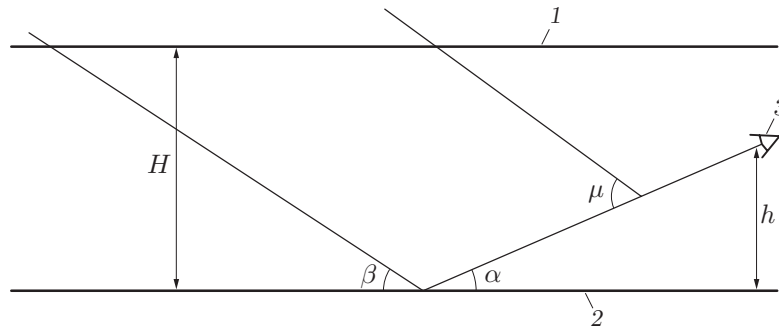


Рис. 4. Световая модель: 1 — верхняя граница тумана; 2 — поверхность земли; 3 — наблюдатель; H — высота верхнего слоя тумана; h — высота наблюдателя над землёй; μ — угол рассеяния света; α — угол зрения между взглядом наблюдателя и землёй; β — угол солнечного света относительно Земли

о потере света луча на единицу расстояния и зависит также от рассеяния частиц и их плотности вдоль луча (см. рис. 2).

Второй источник света для объекта — это рассеяние света туманом на некоторый угол. Компонента тумана вычисляется по формуле [17]:

$$I_t = I_c e^{-BH \operatorname{cosec} \beta} S \frac{1 - e^{-Bh \operatorname{cosec} \alpha}}{B}, \quad (15)$$

где S — угловая функция рассеяния, зависящая от размера и формы частиц, а также от длины волны света.

Угловая функция рассеяния аппроксимируется следующим образом [18]:

$$\frac{(1 - l)^2}{(1 + l^2 - 2l \cos \mu)^{3/2}} \quad (16)$$

(здесь l — параметр, изменяющийся от направления света).

Если компонента тумана (15) увеличивается, то это означает, что объект становится постепенно невидимым.

Объёмные облака на основе трёхмерной текстуры. Рендеринг объёмных облаков является ресурсоёмким процессом, проблематичным для использования в приложениях реального времени [5]. В то же время потребность в визуализации объёмных облаков существует. Так, разработчики ищут новые средства для увеличения реализма игр. Ограничивающие объёмы (skyboxes) и плоские текстуры (flat textures) хорошо работают в 3D-сценах, где камера находится далеко от облаков и не смещается на большие расстояния. Но в открытом мире игры, где положение камеры нельзя предугадать, скайбоксы дают статическое изображение сцены, а плоская текстура — артефакты. Это означает, что изображения неба, облаков и пейзажей размещаются на далёком расстоянии вокруг зрителя, создавая иллюзию реальности. Также отсутствует возможность рендеринга сцены вблизи или внутри облаков при использовании скайбоксов. Чтобы исправить это, облака должны быть размещены в 3D-сцене. Это можно сделать с помощью 2D-текстуры, но тогда изображения будут выглядеть плоскими и не очень реалистичными. Для хороших результатов необходимы объёмные облака. Образование и структура облаков очень динамичны, и главным образом это результат вертикальных движений от конвекции. Облака состоят из мелких капель жидкости или кристаллов, которые образуются, когда влажный воздух поднимается и расширяется по мере достижения более низкого атмосферного давления.

Это происходит вследствие расширения воздуха, понижения температуры и конденсации водяного пара во влажном воздухе.

Чтобы визуализировать облака реалистично, применяется метод отслеживания лучей, т. е. трассировка лучей от камеры в сцену. Для лучей выполняются эквидистантные выборки визуализации облаков. Облака процедурно генерируются с использованием функции шума и хранятся в 3D-текстурах. Улучшенная версия метода шума Перлина реализуется в нашем методе [19]. Трассировка вторичных лучей выполняется в каждой точке внутри облака. Эти лучи трассируются в направлении Солнца для выполнения вычисления освещения. Чтобы получить формы облаков из текстур шума, необходимо несколько шагов для достижения различных характеристик, например возвышающихся вершин, которые являются результатом конвекции. В нашей реализации имеется функция распределения высоты, которая управляет формой облака. Трёхмерные текстуры ограничены свободными формами [5]. Последний этап формирования облака — это использование пороговой функции — простого порогового значения. С его помощью отсекаются «тонкие» части 3D-текстуры, и можно показывать более отчётливые облака.

На рис. 1, *b*, *c* показаны объекты (самолёты) в объёмных облаках.

Визуализация. Процесс рендеринга разбивается на два этапа и распределяется между центральным и графическим процессорами.

Центральный процессор выполняет деление объектного пространства по четверичному дереву. При поиске фрагментов объекта куб делится на четыре части в плоскости XY , для которых проверяется тест на пересечение с объектом. Если пересечение объекта с данным объёмом не имеет места, то из дальнейшего рассмотрения эта часть объёма (куба) исключается. А с частями, в которых может быть пересечение, проводится аналогичная процедура деления. В общем случае этот процесс заканчивается, когда рассматриваемой части соответствует клетка определённого размера [4]. Это функция центрального процессора. Преимущество такого подхода в том, что можно отбросить на ранней стадии большие части куба, в которых нет заданного объекта.

На вход графического процессора поступают фрагменты объекта. Далее фрагмент тестируется на пересечение с лучом, направленным вдоль оси Z , и производится бинарный поиск ближайшей точки пересечения с объектом. Задача состоит в нахождении первой точки, в которой функция обращается в ноль. Найдя такую точку, для каждого луча можно вычислить координату Z . Далее в вычисленных точках определяется нормаль. Имея все координаты и нормали в точках, можно использовать модель освещения.

Реализация метода и результаты работы. При реализации метода применён пакет программы DirectX версии 12 для вершинных и пиксельных шейдеров. Вершинные шейдеры делают вычисления в соответствии с формулами (14) и (15). На вход шейдеров поступают координаты вершин, матрицы преобразования, интенсивность и направление солнечного света, а также коэффициенты его затухания и рассеяния.

Изображения на рис. 1 визуализированы с применением вычислительной машины, содержащей процессор Intel Core i7-2700K и графический акселератор (GPU) NVIDIA GeForce GTX 950 Ti (1024 МГц) с 768-ю унифицированными шейдерными процессорами в реальном масштабе времени (30 кадров в секунду).

Таким образом, привлечено большое количество процессоров для одновременного вычисления множества лучей. При их реализации учитывалось влияние скорости работы с памятью. Максимально задействованы регистры, потому что это самый быстрый вид доступной памяти. Для эффективного использования GPU было взято максимально возможное количество потоков, которые выполняют полезную работу.

В алгоритме отображения рельефа и смены уровней детальности используется тот же механизм, что и для обычной текстуры. Для фильтрации изображения при динамике применяется интерполяция высот по аналогии с текстурой цвета. Время вычислений при

генерации рельефа местности практически не зависит от разрешения карты высот, а обусловлено только разрешением экрана (квадратичная зависимость) и глубиной Z (логарифмическая зависимость). Для ландшафта местности не требуется триангуляция рельефа. В данной технологии главным является иерархический механизм управления уровнями детальности, который напоминает *MipMapping* обычной текстуры цвета. Если проводить полную аналогию с текстурой, то все её положительные свойства наследуются данной технологией. К ним можно отнести простоту анимации поверхности без вычислительных затрат на геометрические преобразования вершин.

Заключение. В представленной работе предложен комбинированный метод визуализации трёх видов данных: аналитического описания объекта на базе функций возмущения; воксельно-базируемого рельефа местности и объёмных облаков на основе трёхмерных текстур. Переход от растривания в плоскости изображения к растриванию в объёме в сочетании с предлагаемыми способами задания поверхностей и объёмов приводит к ряду положительных моментов, повышающих реализм отображаемых сцен. К основным положительным особенностям предлагаемого подхода следует отнести: эффективность метода растривания; уменьшение количества данных, снижение нагрузки на центральный процессор и уменьшение потока данных от него к графическому процессору.

СПИСОК ЛИТЕРАТУРЫ

1. **Вяткин С. И.** Преобразования функционально заданных форм // Программные системы и вычислительные методы. 2014. **9**, № 4. С. 484–499.
2. **Вяткин С. И.** Моделирование сложных поверхностей с применением функций возмущения // Автометрия. 2007. **43**, № 3. С. 40–47.
3. **Вяткин С. И.** Метод бинарного поиска элементов изображения функционально заданных объектов с применением графических акселераторов // Автометрия. 2014. **50**, № 6. С. 89–96.
4. **Вяткин С. И.** Метод рекурсивного поиска элементов изображения функционально заданных поверхностей // Автометрия. 2017. **53**, № 3. С. 53–57.
5. **Вяткин С. И.** Моделирование неоднородностей при визуализации атмосферных эффектов // Вестн. компьютерных и информационных технологий. 2016. **146**, № 7. С. 9–14.
6. **Blinn J. F.** A generation of algebraic surface drawing // ACM Trans. Graph. 1982. **1**, N 3. P. 235–256.
7. **Bloomenthal J., Shoemake K.** Convolution surfaces // Comput. Graph. 1991. **25**, N 4. P. 251–256.
8. **Sealy G., Wyvill G.** Smoothing of three dimensional models by convolution // Proc. of the Computer Graphics Intern. Conf. Pohang, South Korea, 24–28 June, 1996. P. 184–190.
9. **McCormack J., Sherstyuk A.** Creating and rendering convolution surfaces // Comput. Graph. Forum. 1998. **17**, N 2. P. 113–120.
10. **Muraki S.** Volumetric shape description of range data using “blobby model” // Comput. Graph. 1991. **25**, N 4. P. 227–235.
11. **Nishimura H., Hirai M., Kawai T. et al.** Object modelling by distribution function and a method of image generation // Trans. Institute Electron. Commun. Eng. Japan. 1985. **J68-D**, N 4. P. 718–725.
12. **Wyvill G., McPheeters C., Wyvill B.** Data structure for soft objects // Vis. Comput. 1986. **2**, N 4. P. 227–234.
13. **Shirley P., Ashikhmin M., Marschner S.** Fundamentals of Computer Graphics. Boca Raton: CRC Press, 2009. 804 p.

14. **Вяткин С. И.** Визуализация фотореалистичного рельефа местности на основе текстуры-формы с использованием графических ускорителей // Программные системы и вычислительные методы. 2015. **10**, № 1. С. 89–107.
15. **Вяткин С. И.** Метод вычисления интенсивности рассеянного света и тумана с использованием графических акселераторов // Вестн. компьютерных и информационных технологий. 2017. **155**, № 5. С. 35–38.
16. **Вяткин С. И., Долговесов Б. С.** Метод сжатия геометрических данных с применением функций возмущения // Автометрия. 2018. **54**, № 4. С. 18–25.
17. **Klassen R. V.** Modelling the effect of the atmosphere on light // ACM Trans. Graph. 1987. **6**, N 3. P. 215–237.
18. **Blinn J. F.** Light reflection techniques for simulation of clouds and dusty surfaces // ACM SIGGRAPH Comp. Graph. 1982. **16**, N 3. P. 21–29.
19. **Perlin K.** Improved noise // ACM Trans. Graph. 2002. **21**, N 3. P. 681–682.

Поступила в редакцию 19.12.2018

После доработки 30.01.2019

Принята к публикации 07.02.2019
