

Методы распределенной визуализации тумана в трехмерной виртуальной среде с использованием GPU*

А. В. Мальцев

Федеральное государственное учреждение "Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук", г. Москва, Россия

Аннотация. В работе рассматриваются методы распределенной имитации в виртуальной среде тумана с изменяющейся плотностью и зависимостью видимости объектов сцены от высоты расположения камеры, а также расстояния между ней и этими объектами. Предлагается оригинальный подход для наблюдения тумана через полупрозрачные объекты. Разработанные решения используют параллельные вычисления на современных многоядерных GPU и обеспечивают визуализацию сложных трехмерных сцен в масштабе реального времени.

Ключевые слова: туман, атмосферное явление, визуализация, трехмерная сцена, реальное время, шейдеры.

DOI 10.14357/20718632200306

Введение

В настоящее время во многих областях профессиональной деятельности человека для подготовки специалистов применяются системы виртуального окружения и тренажерные комплексы, построенные с использованием моделируемой и визуализируемой на компьютере трехмерной виртуальной среды. Виртуальные сцены при этом могут содержать модели как закрытых (интерьер помещения, здания и т.д.), так и открытых пространств. Последние особо востребованы при обучении пилотов, водителей, операторов робототехнических устройств. Чтобы обеспечить высокий уровень подготовки, видимая учащимися виртуальная среда должна быть как можно более схожа с привычным нам реаль-

ным миром. Однако степень реалистичности визуализируемого пространства значительно ограничена необходимостью поддержки рендеринга в реальном времени, т.е. с частотой генерации кадров не менее 25 раз в секунду. Это требуется для реализации плавного движения виртуальных объектов в кадре, а также возможности интерактивного взаимодействия оператора со средой. Поэтому актуальной задачей является разработка новых распределенных методов и алгоритмов визуализации трехмерных виртуальных сцен в масштабе реального времени с использованием возможностей современных вычислительных устройств, таких как многоядерные графические процессоры (GPU).

Значимой составляющей моделирования открытых пространств в тренажерных системах

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-07-00950.

является имитация атмосферных явлений [1-3]. Среди них особо важен туман, поскольку именно от него в большей степени зависит дальность видимости объектов сквозь атмосферу. Можно выделить несколько различных подходов к моделированию тумана в виртуальной среде. Так в работе [4] рассматривается технология шейдерной постобработки изображений, полученных в результате основного рендеринга трехмерной сцены. Ее преимущество заключается в отсутствии необходимости изменять алгоритмы и программный код основной стадии визуализации. Автор рассматривает расчет на GPU «количества» тумана между наблюдателем и точкой виртуального объекта для каждого пиксела исходного кадра, выполняемый на основе данных буфера глубины видеокарты. Недостатком технологии постобработки является отсутствие тумана за полупрозрачными объектами, что ограничивает их применение в сцене.

В публикации [5] рассматривается метод, который также основан на вычислении «количества» тумана, но уже для каждой точки каждого объекта и непосредственно в процессе визуализации виртуальной среды, а не на дополнительном этапе. При расчете используются трехмерные шумовые текстуры, что позволяет добиться неоднородности имитируемого тумана. Однако изменение его плотности от высоты расположения наблюдателя и точек объектов не учитываются применяемой вычислительной моделью.

Еще один подход, представленный в работе [6], состоит в моделировании объемного тумана на базе технологии *ray-marching* [7], используемой для реализации целого ряда атмосферных эффектов. Имитация тумана в данном случае рассматривается, как одна из составных частей модели расчета освещения сцены, построенной на принципах атмосферного рассеяния. Решение основано на возможностях объемных 3D текстур и вычислительных шейдеров, выполняемых на GPU. Предлагаемый алгоритм включает 4 отдельных этапа, в ходе которых на графическом процессоре вычисляется объемная информация о рассеянии световых лучей для пространства внутри пирамиды видимости камеры. Приведенные авторами результаты рендеринга выглядят довольно реали-

стичными, однако, ввиду тесной взаимосвязи имитации тумана и расчета освещения по модели рассеяния, применение такого подхода в уже имеющихся программных продуктах с собственными моделями расчета освещения сцены представляется весьма затруднительным.

В данной работе рассматриваются оригинальные методы и алгоритмы распределенного рендеринга тумана на GPU в реальном времени с учетом высоты расположения наблюдателя и удаленности от него объектов виртуальной среды, представляющие собой существенную модификацию подходов из работы [4]. Новизной предлагаемых решений являются: перенос расчета и применения тумана из отдельного этапа в основной проход визуализации сцены; авторский подход к вычислению коэффициента «количества» тумана, проходимого зрительным лучом, обеспечивающим возможность наблюдения данного атмосферного явления через полупрозрачные объекты. В отличие от многих имеющихся подходов, предлагаемые методы могут быть довольно легко реализованы в уже имеющихся системах визуализации трехмерных виртуальных сцен. Далее в разделе 1 рассмотрим имитацию тумана на основе базовых принципов, изложенных в [4], а в разделе 2 – предлагаемые изменения структуры и метода вычислений.

1. Имитация тумана с помощью постобработки кадра

Для выполнения распределенного моделирования на GPU тумана с применением технологии постобработки кадра в качестве выхода основного этапа визуализации трехмерной виртуальной сцены необходимо иметь само сформированное изображение кадра, а также соответствующий ему буфер глубины. Изображение и данные буфера должны быть представлены в виде текстур T_{rgb} и T_{depth} , размер $W \times H$ (пикселей) которых эквивалентен текущему размеру области вывода. Чтобы подготовить T_{rgb} непосредственно в процессе визуализации сцены, эффективно использовать механизм рендеринга в текстуру с помощью внеэкранных буферов FBO (*frame buffer object*). Перенос данных из буфера глубины в текстуру T_{depth} можно осуществить средствами гра-

фической библиотеки OpenGL сразу после окончания основного рендеринга.

Постобработка и вывод итогового изображения выполняется с помощью собственного фрагментного шейдера тумана. Перед его применением требуется установить ортографическое проецирование и отключить тесты глубины и смешивания. Затем производится рендеринг прямоугольника размером $W \times H$ (пикселей) с наложением на него текстуры T_{rgb} . При вычислении цвета каждого пиксела (i, j) данного прямоугольника, где $i \in [0, W], j \in [0, H]$ (отсчет ведется от левого нижнего угла кадра), рассчитывается специальный коэффициент F . Он характеризует «количество» тумана, пройденное зрительным лучом R с началом в точке O расположения виртуальной камеры до точки P поверхности некоторого объекта виртуальной среды, которая отображается в пиксел (i, j) текстуры T_{rgb} . Благодаря работе теста глубины на этапе основного рендеринга сцены, поверхность, содержащая точку P , является ближайшей к виртуальной камере по направлению луча среди других поверхностей трехмерной сцены. Математически описанный луч выражается в виде параметрической формулы

$$R(t) = O + t \cdot D, \quad D = \frac{P - O}{|P - O|}, \quad (1)$$

где D – единичный направляющий вектор, $R(t)$ – точка, принадлежащая лучу и зависящая от значения параметра t . Координаты всех используемых точек должны быть выражены в мировой системе координат (СК) WCS. Для нахождения точки P используем алгоритм восстановления ее координат в СК WCS по экранным координатам $P_{scr} = (i/W, j/H, d, 1)$, где $d \in [0, 1]$ – значение глубины точки P , сохраненное в пикселе (i, j) текстуры T_{depth} . Как показано в [8]:

$$P' = (M_{pr} \cdot M_v)^{-1} \cdot (2 \cdot P_{scr} - (1, 1, 1, 1)), \quad (2)$$

$$P = \frac{P'}{P'_w},$$

где M_v и M_{pr} – текущие видовая и проекционная матрицы соответственно.

Вычисляя коэффициент F , будем считать, что плотность моделируемого тумана убывает с увеличением высоты (z -координаты в СК WCS)

по экспоненциальному закону. Пусть ρ_0 – плотность тумана в любой точке виртуального пространства, находящейся на нулевой высоте, т.е. в плоскости XY системы WCS; b – коэффициент падения плотности при увеличении высоты. Тогда зависимость плотности тумана в произвольной точке сцены от ее высоты z выразим следующей функцией:

$$\rho(z) = \rho_0 \cdot e^{-b \cdot z}. \quad (3)$$

При движении по упомянутому выше зрительному лучу R в общем случае меняются все три пространственные координаты, однако в нашем случае интерес представляет только изменение высоты, поскольку от нее зависит плотность тумана в конкретной точке луча. Параметрическая формула такого изменения получается выделением z -компоненты из равенства (1) и имеет вид:

$$z(t) = O_z + t \cdot D_z. \quad (4)$$

Значение искомого коэффициента F определим интегрированием функции (3) на отрезке луча R , ограниченном его началом O и рассматриваемой точкой P :

$$F = \int_0^S \rho(z(t)) \cdot dt, \quad (5)$$

где $S = |P - O|$ – расстояние между виртуальной камерой и точкой P . Подставив в данное равенство выражение $z(t)$ из (4), получим

$$F = \int_0^S \rho_0 \cdot e^{-b(O_z + t \cdot D_z)} \cdot dt = \rho_0 \cdot e^{-b \cdot O_z} \cdot \left(\frac{e^{-b \cdot D_z \cdot t}}{-b \cdot D_z} \Big|_0^S \right),$$

$$F = \rho_0 \cdot e^{-b \cdot O_z} \frac{1 - e^{-b \cdot D_z \cdot S}}{b \cdot D_z}. \quad (6)$$

Найденное значение F приведем к отрезку $[0.0, 1.0]$ и используем при расчете итогового цвета C обрабатываемого пиксела (i, j) , принадлежащего визуализируемому прямоугольнику, с учетом цвета C_{fog} моделируемого тумана и цвета соответствующего пиксела текстуры T_{rgb} :

$$k_{fog} = \max(0.0, \min(1.0, F)),$$

$$C = (1 - k_{fog}) \cdot T_{rgb}(i, j) + k_{fog} \cdot C_{fog}. \quad (7)$$

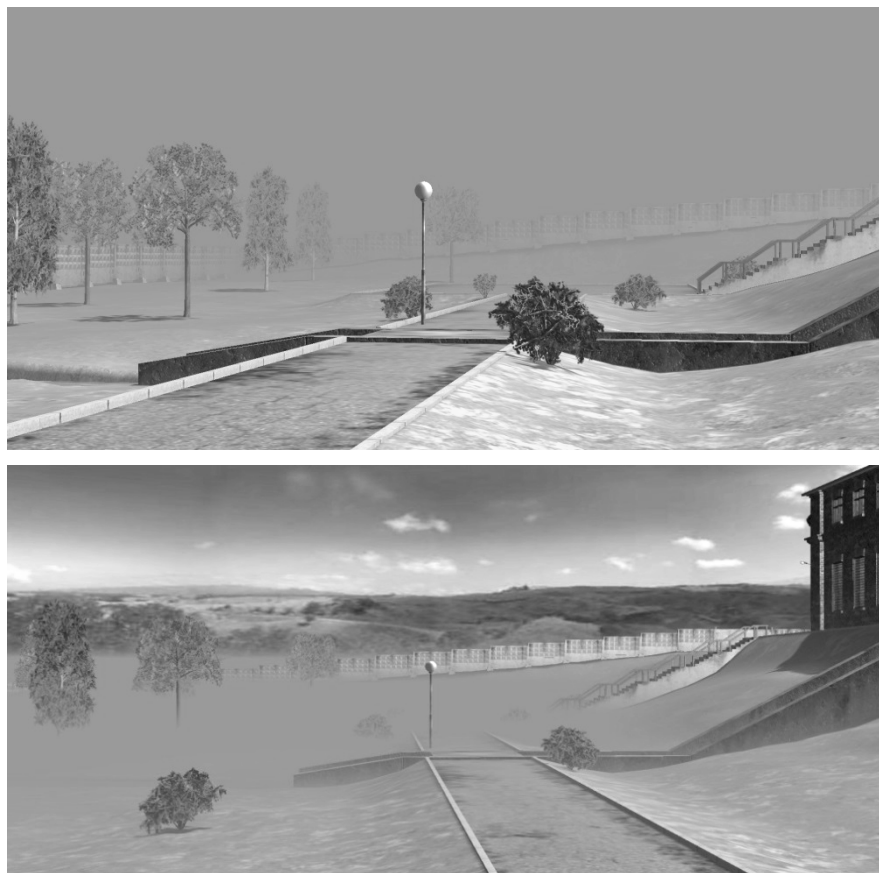


Рис. 1. Моделирование сплошного (сверху) и поземного (снизу) тумана в сцене трехмерного виртуального полигона

Преимуществом изложенного метода имитации тумана в виртуальной среде является учет зависимости видимости объектов сцены не только от расстояния между ними и наблюдателем, но и от высоты расположения последнего. Примеры реализации описанного решения с различными значениями параметров ρ_0 и b представлены на Рис. 1. Для моделирования сплошного тумана в сцене трехмерного виртуального полигона были использованы $\rho_0 = 0.0001$ и $b = 0.0001$, а для поземного – $\rho_0 = 0.001$ и $b = 0.0045$.

2. Расчет тумана в основном шейдере с учетом полупрозрачных объектов

Описанный выше метод распределенной реализации тумана с помощью постобработки обладает хорошей производительностью и легко встраивается в имеющиеся программные реше-

ния, поскольку не требует вмешательства в основные шейдеры визуализации виртуальной сцены. Однако область его использования ограничена сценами, которые не содержат полупрозрачных объектов и поверхностей. Проблема с полупрозрачными элементами возникает ввиду того, что туман рассчитывается и применяется только к ближайшим к камере точкам объектов, определяемым с помощью буфера глубины. А более далекие от наблюдателя поверхности, видимые через такие элементы, не могут быть учтены ввиду отсутствия на этапе постобработки каких-либо данных об их положении в пространстве. На Рис. 2 показан пример добавления модели стекла в уже знакомую сцену виртуального полигона. Поскольку стекло находится близко к камере, то само оно хорошо видно, как и другие близкие объекты. А вот в просматриваемой через него дальней части сцены туман просто отсутствует.

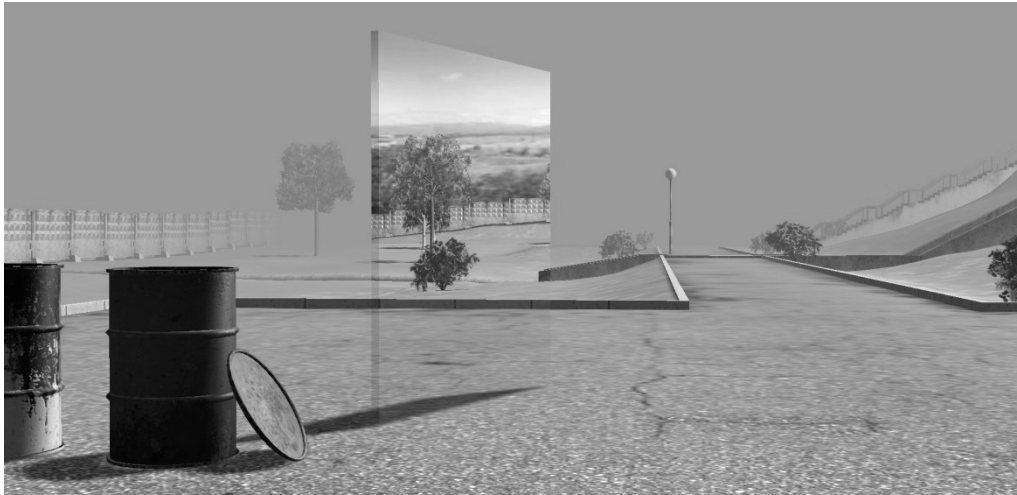


Рис. 2. Проблема отсутствия видимости тумана через полупрозрачные объекты

Для решения этой проблемы в данной работе предлагается новый более универсальный подход, который основан на использовании изложенных выше базовых принципов имитации тумана, но уже не на этапе постобработки, а непосредственно в процессе распределенного рендеринга сцены. Суть данного подхода заключается в разделении обрабатываемых шейдером фрагментов виртуальных объектов на две категории: фрагменты переднего и заднего планов, и вычислении для них значений коэффициентов F_n и F_f «количества» тумана на независимых отрезках зрительного луча R (Рис. 3). К фрагментам переднего плана отнесем те, которые принадлежат ближайшим к виртуальной камере поверхностям. Именно их глубина сохранена в буфере глубины сцены. Фрагменты всех остальных поверхностей будут представлять задний план.

Чтобы реализовать предлагаемый метод, нам потребуется передать в основной фрагментный шейдер содержимое буфера глубины сцены в виде текстуры T_{depth} размера $W \times H$ (пикселей), эквивалентного текущему размеру области вывода. Получить такую текстуру можно с помощью этапа предобработки, выполнив быструю визуализацию только геометрии объектов без расчета освещения, применения текстур и записи результата в цветовой буфер. Используя технологию внеэкранных буферов FBO, запись данных буфера глубины можно осуществлять непосредственно в T_{depth} .

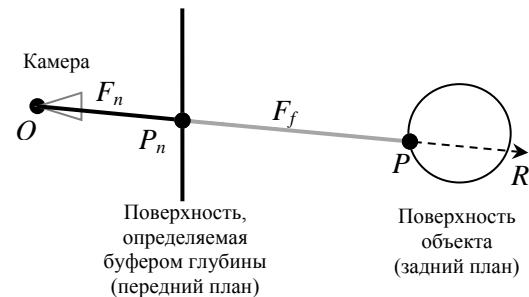


Рис. 3. Расчет «количества» тумана для фрагментов переднего и заднего планов

В процессе основного рендеринга для каждого объекта виртуальной сцены выполним вершинный GLSL-шейдер, который принимает на вход координаты V_{ocs} вершин полигонов этого объекта (представлены в локальной СК OCS), а также модельную M_m (переводит из СК OCS в мировую СК WCS), видовую M_v (переводит из СК WCS в СК VCS камеры) и проекционную M_{pr} матрицы. В данном шейдере произведем расчет координат поступающих вершин в мировой СК WCS и СК CCS отсечения по формулам:

$$V_{wcs} = M_m \cdot V_{ocs}, \quad V_{ccs} = M_{pr} \cdot M_v \cdot V_{wcs}.$$

Полученные координаты V_{wcs} передадим на выход шейдера в качестве пользовательской выходной переменной, а V_{ccs} – запишем в предопределенную выходную переменную положения $gl_Position$. Обработка вершин выполняется

на GPU в параллельных потоках. Далее графическая карта растеризирует каждый полигон на фрагменты и интерполирует переданные значения V_{ccs} , V_{wcs} трех его вершин, получая для каждого фрагмента его положение в СК области вывода и координаты P (в СК WCS) соответствующей ему точки объекта. Эти данные автоматически предаются на вход фрагментному шейдеру.

Во фрагментном шейдере воспользуемся предопределенной входной переменной $gl_FragCoord$, получая в качестве ее компонент $x \in [0, W]$ и $y \in [0, H]$ положение обрабатываемого фрагмента в системе координат области вывода (кадра) с размером $W \times H$ (пикселов). Координаты P доступны посредством пользовательской входной переменной. Далее необходимо определить, к какому из планов (переднему или заднему) относится рассматриваемый фрагмент и на основе этой информации рассчитать для него соответствующий коэффициент F «количества» тумана. Для этого вначале найдем точку некоторого виртуального объекта, которая является ближайшей к виртуальной камере по зрительному лучу R , проходящему через P (Рис. 3). Экранные координаты $P_{n,scr}$ этой точки равны $(x/W, y/H, d, 1)$, где $d \in [0, 1]$ – значение ее глубины, сохраненное в пикселе (x, y) текстуры T_{depth} . Используя формулу (2), восстановим ее координаты P_n в мировой СК WCS. В случае выполнения неравенства

$$\|P - O\| - \|P_n - O\| < \varepsilon,$$

где O – координаты камеры в СК WCS, ε – близкая к нулю константа, будем считать, что точки P и P_n совпадают, а рассматриваемый фрагмент относится к переднему плану. И, наоборот, при его несоблюдении фрагмент принадлежит заднему плану. Для фрагментов переднего плана коэффициент F_n «количества» тумана определим, как и в п.1, с помощью формулы (6) на отрезке зрительного луча от точки O до точки P_n (Рис. 3). Затем по формуле (7) произведем расчет их цветов с учетом моделируемого тумана.

Фрагменты заднего плана могут быть видны наблюдателю только сквозь полупрозрачные поверхности переднего плана. Чтобы получить правильный результат визуализации,

полупрозрачные объекты обычно сортируются по глубине относительно наблюдателя и выводятся уже после всех непрозрачных. Если вновь поступающий на обработку фрагмент полупрозрачного объекта проходит тест глубины (т.е. его глубина относительно камеры меньше той, которая уже записана в буфере глубины для соответствующего фрагменту пиксела), то пиксел буфера кадра окрашивается в смесь его текущего цвета и цвета рассматриваемого фрагмента. Смешивание производится графическим процессором на основе отвечающего за прозрачность значения α -канала приходящего фрагмента. Например, в ситуации на Рис. 3, точки P_n (передний план) и P (задний план) расположены на одном зрительном луче R , а значит попадают в один и тот же пиксел буфера кадра. Если P_n принадлежит полупрозрачной поверхности, то вычисленный шейдером цвет в P первым запишется в буфер кадра, после чего произойдет его смешивание с цветом в P_n .

Принимая во внимание такой механизм смешивания цветов, можно заметить, что вычисление коэффициента F_f «количества» тумана для точки P аналогично точке P_n привело бы к двойному включению отрезка от O до P_n в расчет тумана для результирующего цвета пиксела буфера кадра. Чтобы исключить такую ситуацию, найдем F_f для точек заднего плана на отрезке от P_n до P , т.е. вычислим интеграл (5) из п.1 в пределах S_n , $S = S_n + S_f$ где $S_n = |P_n - O|$ и $S_f = |P - P_n|$:

$$\begin{aligned} F_f &= \int_{S_n}^S \rho_0 \cdot e^{-b(O_z+t \cdot D_z)} \cdot dt = \\ &= \rho_0 \cdot e^{-b \cdot O_z} \cdot \left(\frac{e^{-b \cdot D_z \cdot t}}{-b \cdot D_z} \Big|_{S_n}^{S_n + S_f} \right) = \\ &= \rho_0 \cdot e^{-b \cdot (O_z + D_z \cdot S_n)} \frac{1 - e^{-b \cdot D_z \cdot S_f}}{b \cdot D_z}. \end{aligned}$$

Учитывая, что

$$O_z + D_z \cdot S_n = P_{n,z},$$

получим

$$F_f = \rho_0 \cdot e^{-b \cdot P_{n,z}} \frac{1 - e^{-b \cdot D_z \cdot S_f}}{b \cdot D_z}.$$

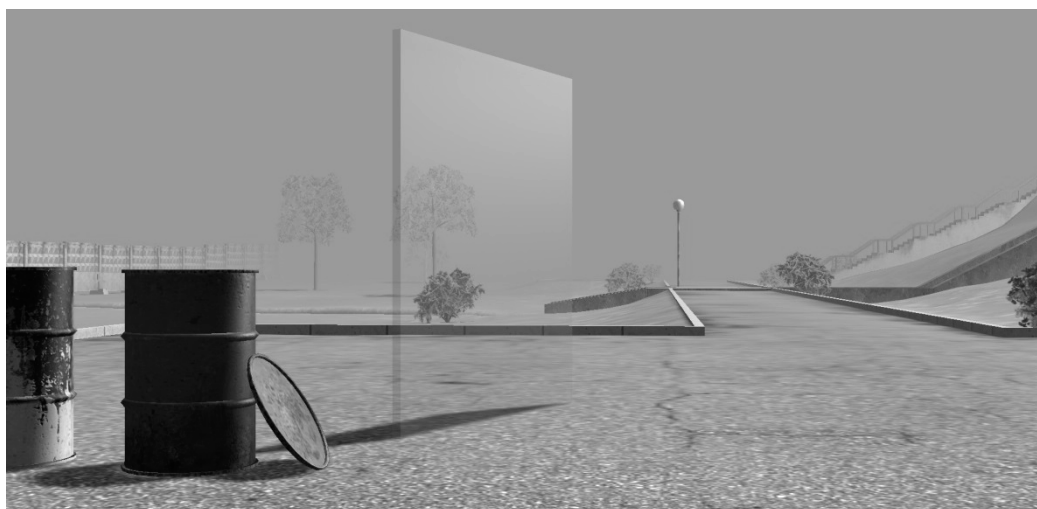


Рис. 4. Наблюдение тумана через стеклянную поверхность

Далее необходимо рассчитать цвет фрагмента заднего плана с учетом имитируемого тумана, воспользовавшись формулой (7). Пример реализации описанного подхода представлен на Рис. 4.

3. Результаты

В данной работе была проведена оценка производительности предложенных в разделе 2 решений. Она производилась путем измерения частоты генерации кадров (FPS) при визуализации виртуальной сцены трехмерного полигона (Рис. 1), содержащей порядка 600 тысяч треугольных полигонов. Эксперименты выполнялись на персональном компьютере с процессором Intel Core i7 (3,5 ГГц) и видеокартой NVIDIA GeForce GTX 970 (GPU содержит 1664 ядра) для различных разрешений получаемого изображения: HD 1280×720, Full HD 1920×1080 и UHD 4K 3840×2160. Результаты проведенных экспериментов занесены в Табл. 1. Они показывают, что рендеринг высоко полигональной сцены с моделированием тумана на основе разработанных методов и алгоритмов выполняется с высокими значениями FPS, обеспечивающими масштаб реального времени даже при синтезе изображений в формате 4K. Снижение FPS по сравнению с визуализацией той же сцены без имитации тумана находится в допустимых пределах. Поэтому предложенные решения могут использоваться при создании систем визуализации для имитационно-тренажерных комплексов и систем виртуального окружения.

Табл. 1. Замеры частоты генерации кадров для сцены виртуального полигона

Разрешение изображения, пиксели	FPS, кадров/с	
	Без тумана	С туманом
1280×720 (HD)	210	206
1920×1080 (Full HD)	174	169
3840×2160 (4K)	127	120

Заключение

В данной работе представлены новые методы и алгоритмы для моделирования тумана в трехмерных сценах на GPU в масштабе реального времени. Их преимуществом по сравнению со многими другими известными подходами является учет высоты расположения наблюдателя при вычислении видимости им виртуальной среды, а также простота встраивания в уже имеющиеся системы рендеринга. Практическим вкладом в рассматриваемую область являются программные модули для систем визуализации трехмерных виртуальных сцен, созданные на базе описанных в статье решений. Данные модули обеспечивают распределенный рендеринг тумана с изменяющейся плотностью и зависимостью видимости поверхностей объектов виртуальной среды от высоты расположения камеры. Для реализации программных компонентов использовались возможности параллельных вычислений на со-

временных многоядерных графических процессорах, а также шейдерный язык GLSL последнего поколения. Созданные модули прошли успешную апробацию в составе системы визуализации «GLView», разработанной в ФГУ ФНЦ НИИСИ РАН.

Значимость полученных результатов состоит в возможности их применения при реализации программного обеспечения имитационно-тренажерных комплексов и систем виртуального окружения, что позволит повысить реалистичность синтезируемых в них изображений с минимальными потерями в скорости визуализации.

Литература

1. Narasimhan S.G., Nayar S.K. Vision and the atmosphere // *International Journal of Computer Vision*. 2002. Vol.48, No.3. P.233-254.
2. Мальцев А.В. Моделирование атмосферных осадков в трехмерных сценах с использованием CUDA // Ин-

формационные технологии и вычислительные системы. 2015. №2. С.31-39.

3. Fearing P. Computer modelling of fallen snow // In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000. P.37-46.
4. Quilez I. Better fog // *Inigo Quilez :: fractals, computer graphics, mathematics, shaders, demoscene and more*. URL: <https://www.iquilezles.org/www/articles/fog/fog.htm> (дата обращения: 02.07.2020)
5. Zdrojewska D. Real time rendering of heterogenous fog based on the graphics hardware acceleration. 2004.
6. Wroński B., Volumetric Fog and Lighting // *GPU Pro 6: Advanced Rendering Techniques*. 2015. P.217-242.
7. Tóth B., Umenhoffer T. Real-Time Volumetric Lighting in Participating Media // *Eurographics 2009*. URL: <http://cg.iit.bme.hu/~szirmay/lightshaft.pdf> (дата обращения: 02.07.2020)
8. Мальцев А.В. Методы распределенного рендеринга виртуальных сцен на GPU с реализацией эффекта размытия в движении // *Программная инженерия*. 2020. Т.11, №4.

Мальцев Андрей Валерьевич. Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук» (ФГУ ФНЦ НИИСИ РАН), г. Москва, Россия. Ведущий научный сотрудник. Кандидат физико-математических наук. Количество печатных работ: 90. Область научных интересов: компьютерная графика, системы виртуальной реальности, информационные технологии. E-mail: avmaltcev@mail.ru

Methods for Distributed Visualization of Fog in 3D Virtual Environment Using GPU

A. V. Maltsev

Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences" (SRISA RAS), Moscow, Russia

Abstract. The paper presents methods for distributed simulation of fog with variable density and visibility dependence of virtual environment objects on a camera height, as well as on distance between it and these objects. The original approach is proposed for observing fog through semitransparent objects. Created solutions use parallel computing on modern multi-core GPU and provide real-time visualization of complex three-dimensional scenes.

Keywords: fog, atmospheric phenomenon, visualization, three-dimensional scene, real time, shaders.

DOI 10.14357/20718632200306

References

1. Narasimhan S.G., Nayar S.K. Vision and the atmosphere // *International Journal of Computer Vision*. 2002. Vol.48, No.3. P.233-254.
2. Maltsev A.V. Modelirovanie atmosferykh osadkov v trekhmerykh stsenakh s ispolzovaniem CUDA [Simulation of atmospheric condensation at 3D scenes by using CUDA] // *Informatsionnye tekhnologii i vychislitelnye*
3. Fearing P. Computer modelling of fallen snow // In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000. P.37-46.
4. Quilez I. Better fog // *Inigo Quilez :: fractals, computer graphics, mathematics, shaders, demoscene and more*. URL: <https://www.iquilezles.org/www/articles/fog/fog.htm> (accessed: July 2, 2020)

5. Zdrojewska D. Real time rendering of heterogenous fog based on the graphics hardware acceleration. 2004.
6. Wroński B., Volumetric Fog and Lightning // GPU Pro 6: Advanced Rendering Techniques. 2015. P.217-242.
7. Tóth B., Umenhoffer T. Real-Time Volumetric Lighting in Participating Media // Eurographics 2009. URL: <http://cg.iit.bme.hu/~szirmay/lightshaft.pdf> (accessed: July 2, 2020)
8. Maltsev A.V. Metody raspredelenного renderinga virtualnykh stsen na GPU s realizatsiei effekta razmytiia v dvizhenii [Methods for Distributed Rendering of Virtual Scenes on GPU with Motion Blur Implementation] // Programmnyaya Ingeneria [Software Engineering]. 2020. T.11, №4.

Maltsev A.V. PhD. Federal State Institution “Scientific Research Institute for System Analysis of the Russian Academy of Sciences”, 36/1 Nakhimovskiy Av., Moscow, 117218, Russia, e-mail: avmaltcev@mail.ru