

УДК 004.94:551.576

МЕТОДЫ ВИЗУАЛИЗАЦИИ ПОГОДНЫХ ЯВЛЕНИЙ В ИМИТАЦИОННЫХ СИСТЕМАХ

Н. А. Агафонов, А. М. Гиацингов, А. В. Родителев

*Федеральный научный центр Научно-исследовательский институт
системных исследований РАН,
agafonovkolja@gmail.com, algts@inbox.ru, avrod_94@mail.ru*

При моделировании внекабинного пространства в авиационных тренажерных комплексах, зачастую, необходимо воспроизводить эффекты, возникающие при полетах в облаках. При этом нужно обеспечить работу подсистемы визуализации в режиме реального времени с частотой смены кадров не менее 25 в секунду. В работе проанализированы подходы к визуализации облачности: использование skybox, процедурное моделирование, моделирование на основе клеточного автомата, использование шума Перлина совместно с билбордами и вокселями. Выявлены основные недостатки, такие как невозможность пролета сквозь облака, низкая реалистичность и высокая ресурсоемкость. Показано, что среди всех, подход с использованием шума Перлина совместно с билбордами дает наиболее реалистичные результаты и обеспечивает работу подсистемы визуализации в масштабе реального времени с заданной частотой кадров.

Ключевые слова: визуализация, имитационные системы, облачность, шум Перлина, процедурное моделирование.

VISUALIZATION METHODS OF WEATHER EVENTS IN SIMULATION SYSTEMS

N. A. Agafonov, A. M. Giatsintov, A. V. Roditelev

*System Research Institute, Russian Academy of Sciences,
agafonovkolja@gmail.com, algts@inbox.ru, avrod_94@mail.ru*

It is necessary to simulate the events that occur during cloud flying when modeling outside world in flight simulators. In addition to that, visualization system should work in real-time with no less than 25 frames per second. This article presents an analysis of different types of cloud visualization: procedural modelling, modelling with the use of cellular automaton, skyboxes, the Perlin noise used in conjunction with billboard and voxel rendering. Major disadvantages of these methods are presented, such as computational complexity, lack of realism and inability to fly through the clouds. It is demonstrated that the use of the Perlin noise in conjunction with billboards provides better performance of the visualization system and more realistic results.

Keywords: visualization, simulation systems, clouded sky, Perlin noise, procedural modeling.

Введение

В настоящее время, в связи со значительной сложностью технических систем, при эксплуатации возникает риск возникновения нештатных и опасных для жизни ситуаций, появляется потребность в тренажерно-обучающих системах. Тренажерно-обучающая система оператора сложной технической системы (далее – ТОС) – техническое средство для подготовки операторов сложных технических систем, отвечающее требованиям методик подготовки, обеспечивающее получение знаний, навыков и умений, реализующее модель таких систем и осуществляющее контроль над действиями обучаемого, а также для исследований

[1]. ТОС могут применяться для формирования индивидуальных профессиональных навыков и умений, а также отработки групповых операций. Подсистема визуализации ТОС обеспечивает отображение результатов моделирования внешней среды и объекта управления с помощью устройств отображения информации. Подсистема должна обеспечивать воспроизведение визуальной картины с достаточно подробным содержанием, позволяющим операторам ТОС успешно выполнять поставленные задачи. Визуализация погодных явлений – неотъемлемая часть при создании авиационных ТОС [2]. При моделировании внекабинного пространства зачастую необходимо воспроизводить эффекты, возникающие при полетах в облаках. В руководстве ИКАО 9625, описаны требования, предъявляемые к визуализации внекабинного пространства. Должны воспроизводиться следующие эффекты: переменная плотность облаков, многослойная облачность, включая отдельные, рассеянные и разорванные облака, а также сплошная облачность, которая частично или полностью препятствует обзору поверхности земли, постепенный выход из облаков и переход к видимости окружающей обстановки. Так же немаловажным условием является дальность прорисовки облаков. В реальных условиях, видимость может достигать десятков километров. При этом необходимо обеспечить работу подсистемы визуализации в режиме реального времени с частотой смены кадров не менее 25 в секунду [3]. Для решения данной задачи, вначале необходимо проанализировать типы облаков и способы их визуализации.

Виды облаков. Метеорологи классифицировали облака в зависимости от их внешнего вида. Каждый из этих типов имеет латинское название, такое как «stratocumulus (слоисто-кучевые)». Эти названия формируются из следующих корней: кучевые облака (cumulus) – плотные, днем ярко-белые облака со значительным вертикальным развитием. Высота нижней границы обычно от 800 до 1500 м, иногда – 2–3 км и более. Толщина составляет 1–2 км, иногда – 3–5 км. Верхние части кучевых облаков имеют вид куполов или башен с округлыми очертаниями. Слоистые облака (stratus) образуют однородный слой, сходный с туманом, но расположенный на некоторой высоте (чаще всего от 100 до 400 м, иногда 30–90 м). Обычно они закрывают все небо, но иногда могут наблюдаться в виде разорванных облачных масс. Перистые (cirrus) – отдельные, тонкие, нитеобразные облака в виде белых тонких волокон или чуть сероватых вытянутых гряд и клочьев, часто имеющие вид бородки пера, обыкновенно белого цвета; иногда располагаются полосами, пересекающими небесный свод подобно меридианам [4].

Основным источником света в дневное время служит солнце, как результат – цвет облаков в большей степени зависит от солнца. Вот почему мы видим белые облака в полдень и красные на закате. Однако не только солнце влияет на цвета облаков. Они так же зависят от цвета неба и земли. Ночью основным источником света становится луна, когда она видима.

Skybox. Существует несколько подходов к визуализации облаков, каждый из которых имеет свои преимущества и недостатки. Один из подходов основан на использовании Skybox – панорамная текстурой, которая находится за всеми объектами в сцене, чтобы визуализировать небо. Это шесть текстур, представляющие шесть видимых направлений вдоль основных осей. Если Skybox создан правильно, изображения создают непрерывный окружающий образ без швов, который можно рассматривать «изнутри» в любом направлении. Таким образом, Skybox – это простой способ повысить реалистичность сцены с минимальной нагрузкой на видеокарту. С помощью данного подхода можно реализовать реалистичное облачное небо, но он не приемлем для авиационных тренажеров, поскольку невозможно выполнение пролета сквозь облака.

Визуализация облачного неба с использованием моделей. В данном подходе облачность формируется с помощью шаблонов: для облаков различного типа дизайнеры подготавливают соответствующее количество моделей, составленных из заданного числа частиц. Затем из этих моделей либо на основе случайного распределения, либо вручную формируется облачный слой. Существенным недостатком такого метода является ограниченное число моделируемых типов облачной поверхности.

Процедурное моделирование для создания кучевых облаков использует пирокластические сферические примитивы. Такие примитивы можно получить путем фрактального смещения вершин сферы либо на случайную величину (2D смещение), либо на случайный вектор (3D смещение). Обычно в качестве функции фрактального смещения используется так называемое фрактальное броуновское движение. Важной особенностью процедурного моделирования является отсутствие ограничений в степени детализации. Таким образом, доступно любое желаемое разрешение моделирования. Процедурное моделирование позволяет представить объект в аналитической форме. Отсюда основной недостаток – вычислительные затраты возрастают с увеличением сложности аналитической записи модели. В то же время не существует и ограничений, в количестве вариаций получаемых моделей.

Моделирование на основе клеточного автомата. Метод основывается на двух процессах: моделирование и рендеринг. Пространство моделирования разделено на воксели. Воксели – аналоги двумерных пикселей для трехмерного пространства и соответствуют ячейкам, используемым в клеточном автомате. Каждая ячейка имеет три логические переменные: пар/влажность (hum) определяет, достаточно ли воды для формирования облака; облачность (cld) показывает, содержит ли ячейка часть облака; активация (act) указывает, что фазовый переход готов к осуществлению. Состояние каждой переменной будет равно 0 или 1. Эволюция облаков моделируется путем применения простых правил перехода на каждом временном шаге. Правила перехода описывают образование, угасание и реакцию на ветер. Так как состояние равно 0 или 1, правила могут быть выражены булевыми операциями. Изображения генерируются в процессе рендеринга с использованием результатов моделирования. Для каждого вокселя, исходя из правил, определяется наличие или отсутствие облака. Затем облака визуализируются с использованием метода объемного рендеринга. Процесс рендеринга состоит из двух шагов. Первый шаг вычисляет интенсивность света, достигающего центра каждого вокселя. Тени от облаков также вычисляются на этом этапе [5]. На следующем шаге генерируется изображение.

Данный подход обеспечивает визуализацию в режиме реального времени. Однако при таком подходе наблюдается средняя реалистичность выходных изображений.

Подход к визуализации облаков с помощью шума Перлина. Это градиентный шум, состоящий из набора псевдослучайных единичных векторов (направлений градиента), расположенных в определенных точках пространства и интерполированных функцией сглаживания между этими точками. Для генерации шума Перлина в одномерном пространстве необходимо для каждой точки этого пространства вычислить значение шумовой функции, используя направление градиента (или наклон) в указанной точке. Алгоритм шума Перлина можно масштабировать для одно-, двух- и трехмерного вида. Более того, в алгоритм можно ввести четвертое временное измерение, позволяя алгоритму динамически изменять текстуры во времени [6]. Используя шум Перлина, можно процедурно генерировать трехмерные карты облачности, тем самым мы имеем значительное число вариаций облаков.

Использование билбордов. Билборд – это полигон, все время направленный на наблюдателя. В данном подходе облачная поверхность представляет собой трехмерную решетку, в узлах которой находятся билборды. С помощью шума Перлина, путем складывания октав шума, генерируется трехмерная карта облачности. На основе этой карты выставляются позиции билбордов. После чего к каждому билборду применяется небольшая текстура, соответствующая части облака. Для корректного отображения облачной поверхности, необходимо производить сортировку билбордов от дальнего к ближнему относительно позиции камеры. Так же возможно применение следующих оптимизаций: облачная поверхность делится на сектора, и выполняется проверка, какие из секторов попадают в область видимости камеры. Видимые сектора сортируются от дальнего к ближнему, а не входящие в область видимости отсекаются. Также необходимо отсекал сектора полностью скрытые за другими секторами. Так же для всех билбордов в видимых секторах необходимо рассчитать цвет с уче-

том источников света, плотности и рассеяния. Данный подход дает хорошие показатели реалистичности и обеспечивает работу в режиме реального времени.

Визуализация с помощью вокселей. Отличием данного подхода от предыдущего является использованием вокселей вместо билбордов. Таким образом, небо представляется трехмерной решеткой, в узлах которой находятся воксели. С помощью шума Перлина генерируется трехмерная карта облачности, далее на основе этой карты идет выставление позиций вокселей. После чего, при помощи трассировки лучей света, рассчитывается цвет вокселей с учетом источников света, плотности облаков и рассеяния света. Данный подход дает более детализированный результат в сравнении с билбордами, но является более ресурсозатратным и сложен в оптимизации.

Повышение производительности прорисовки облаков. Важную роль в достижении реалистичности отображаемых облаков играет освещение. При традиционном (forward rendering) подходе расчет освещения производится следующим образом: для каждой вершины в трехмерной сцене и каждого пикселя конечного изображения, выводимого на экран, последовательно производится расчет влияния каждого источника света в сцене, т. е. этапы расчета освещения и геометрической обработки выполняются для каждого пикселя. Кроме того, расчеты будут произведены даже для скрытых и перекрывающихся поверхностей, а также для тех пикселей, которые могут не попасть в итоговое изображение. Соответственно, при большом количестве источников света производительность подсистемы визуализации может быть существенно снижена.

Для повышения производительности разрабатываемой подсистемы применяют метод так называемой отложенной визуализации, в частности, отложенного освещения. Основной идеей является отделение этапа геометрической обработки от этапа освещения трехмерной сцены. Прорисовка изображения осуществляется в несколько этапов. Вся геометрия сцены прорисовывается один раз, при этом, сохраняется информация о цвете, нормалях и глубине прорисовки для каждого пикселя в промежуточный G-Buffer, который используется на последующих этапах прорисовки [7]. Далее, используя информацию о цвете, глубине и нормалях, сохраненную в G-Buffer, производится расчет освещения. Затем итоговое изображение копируется в кадровый буфер. Всего этапов четыре: этап геометрической обработки, этап освещения, этап постобработки и заключительный этап [8].

Однако прорисовка трехмерных сцен при помощи отложенной визуализации также имеет множество ограничений. Во-первых, существуют значительные проблемы с отображением полупрозрачных объектов из-за того, что в G-Buffer сохраняется информация только о ближайшем к виртуальной камере трехмерной объекте. Во-вторых, ограничено применение аппаратного сглаживания изображения (Multisample Anti-aliasing – MSAA) из-за сложности реализации. В-третьих, отложенная визуализация характеризуется значительным расходом видеопамяти, который увеличивается при повышении разрешения итогового изображения, а также значительным объемом данных, передаваемых по шине данных при прорисовке каждого кадра.

В настоящее время существует несколько методов, устраняющих некоторые недостатки отложенной визуализации с сохранением ее преимуществ. Одним из таких методов является тайловая (плиточная) визуализация, которая может применяться как с традиционным подходом к освещению трехмерной сцены, так и с отложенным [9]. Отложенная визуализация с помощью тайлов позволяет уменьшить объем передаваемых данных по шине, но увеличивает общий объем вычислений, связанных с освещением. С учетом мощностей современных графических адаптеров, применение отложенной визуализации с помощью тайлов обеспечивает значительное увеличение производительности системы визуализации, по сравнению с «чистой» отложенной визуализацией.

Принцип тайловой визуализации состоит в группировании пикселей итогового изображения, которое будет отображено на экране, в прямоугольные участки (тайлы), а также использовании значения глубины каждого пикселя для определения дополнительных зон види-

мости – усеченных пирамид (View Frustum). Соответственно, тайлы, в которых находятся пиксели со схожими значениями глубины, например, в данных пикселях отображается один и тот же трехмерный объект, определяют достаточно небольшой объем трехмерного пространства. В то же время, для тайлов, пиксели, которых существенно отличаются по значениям глубины, дальняя граница ограничивающего объема тайла должна совпадать с максимальным значением глубины его пикселей. Это снижает эффективность отсечения обрабатываемых источников света, и приводит к прямой зависимости производительности от выводимого изображения [10]. Данный аспект особенно нежелателен в приложениях, работающих в режиме реального времени (не менее 25 кадров в секунду), так как становится практически невозможно гарантировать стабильность частоты смены кадров системы визуализации.

Заключение

С помощью описанных подходов возможна визуализация облачного неба, однако не все из них подходят для авиационных ТОС. В подходе с использованием skybox не реализуется пролет сквозь облака. Использование моделей дает слишком малое количество вариаций облаков. Моделирование на основе клеточного автомата дает низкие показания реалистичности. Среди всех подходов, методы с использованием шума Перлина позволяют добиться требуемых результатов, при этом обеспечивается визуализация в режиме реального времени.

Работа выполнена при поддержке РФФИ, грант № 17-07-00169 А.

Литература

1. Гиацинтов А. М., Мамросенко К. А., Решетников В. Н. Инструментальные средства предтренажерной и тренажерной подготовки операторов сложных технических систем [Электронный ресурс] // Программные продукты, системы и алгоритмы. 2014. № 1. С. 1–3. URL: <http://swsys-ru/simulator-training-operators.html> (дата обращения: 08.07.2015).
2. Мамросенко К. А., Решетников В. Н. Моделирование подстилающей поверхности в имитационных системах // Программные продукты и системы. 2015. № 4. С. 70–74.
3. Мамросенко К. А. Имитационно-тренажерные и обучающие распределенные системы // Программные продукты и системы. 2008. № 3. С. 32–35.
4. Решетников В. Н., Мамросенко К. А. Основы построения тренажерно-обучающих систем сложных технических комплексов // Программные продукты и системы. 2011. № 3. С. 86–90.
5. Андреев А. О., Дукальская М. В., Головина Е. Г. Облака: происхождение, классификация, распознавание. СПб. : РГГМУ, 2007. 228 с.
6. Dobashi Y., Kaneda K., Yamashita H., Okita T., Nishita T. A Simple, Efficient Method for Realistic Animation of Clouds. N.Y. : ACM Press/Addison-Wesley Publishing Co, 2000. P. 19–28.
7. Боресков А. Steps3D – Tutorials – Deferred Shading. [Электронный ресурс]. URL: <http://steps3d.narod.ru/tutorials/ds-tutorial.html> (дата обращения: 08.07.2015).
8. Alamia M. Coding Labs: Simple OpenGL Deferred Rendering Tutorial [Электронный ресурс]. URL: http://www.codinglabs.net/tutorial_simple_def_rendering.aspx (дата обращения: 08.07.2015).
9. Olsson O., Assarsson U. Tiled Shading // J. Graph. GPU Game Tools. 2011. Т. 15. № 4. С. 235–251.
10. Гиацинтов А. М., Мамросенко К. А. Отображение трехмерных объектов с использованием кластерной визуализации // Программные продукты и системы. 2016. № 4. С. 69–72.