

УДК 004.921+004.925.83

Моделирование реалистичных изображений облаков

Сафонова А.В., студент

*кафедра «Теоретическая информатика и компьютерные технологии»
Россия, 1050005, г. Москва, МГТУ им. Н.Э. Баумана*

*Научный руководитель: Вишняков И.Э.,
Россия, 1050005, г. Москва, МГТУ им. Н.Э. Баумана
bauman@bmstu.ru*

Введение. В наше время, когда вычислительная мощность компьютеров стала довольно высокой, визуализация различных природных явлений считается весьма востребованной задачей, поскольку такие объекты как ландшафт, водная поверхность, растительность, небо и облака являются неотъемлемой частью практически любой естественной сцены. Все эти детали оказывают значительное влияние на реалистичность, и их реализация находит себя в разных сферах (разработка компьютерных игр, моделирование спецэффектов, создание анимации). Особым образом в этом ряду выделяется моделирование неба и облачности. Во-первых, раскраска неба в соответствии со временем суток является очень сложной и практически неразрешимой в реальном времени задачей [1]. Во-вторых, облака представляют собой динамический объект, структура которого сложна и неоднородна: плотность, с которой распределены по облаку капли воды неравномерна как на границах, так и внутри облака. Она меняется постоянно, в особенности при изменении погоды и при появлении осадков. Но при визуализации внимание уделяется в основном частицам на поверхности облака, тогда как плотность частиц, находящихся внутри может считаться постоянной, поскольку она влияет лишь на преломление и рассеивание света, и при визуализации это можно отразить, придавая облакам белый цвет [2].

Основной целью данной работы является изображение трехмерных облаков различных типов (перистые, кучевые) и сопутствующих им природных явлений (дождь, снег). При этом нужно обеспечить возможность интерактивно управлять изменением этих типов и придать облакам динамичность: реализовать изменение их структуры, связанное с

хаотичным передвижением частиц, образующих облако, и передвижение первых в соответствии с физическими законами.

Алгоритмы двумерного моделирования облачности. Облака имеют сложную физическую структуру, которая визуально воспринимается разрозненной, но не полностью хаотичной. Говоря конкретнее, в одних местах мы можем наблюдать скопление их «массы», а в других – разреженность, и все эти области плавно переходят друг в друга по всему объему облака. Одним из способов описать такую структуру при моделировании облачности является использование шумовой функции, например, шума Перлина (Perlin Noise). Принцип его создания подробно описан в [3] и заключается в построении функции, являющейся комбинацией шумовой и интерполяционной.

Чтобы реализовать шум Перлина, нужно построить ряд функций, основными величинами для которых будут частота и амплитуда. Амплитудой будем называть значения локальных экстремумов, а частотой – величину обратную длине волны (т.е. разнице между двумя ближайшими нулями). Сначала строится функция с минимальной выбранной частотой и максимальной амплитудой. Для построения каждой следующей функции нужно увеличивать частоту и уменьшать амплитуду. Тогда последняя функция будет иметь максимальную выбранную частоту и минимальную амплитуду. Каждая из функций называется октавой. Их количество влияет на реалистичность конечного изображения. Сложив все эти функции, мы получим результирующую - гладкую и в достаточной степени хаотичную (Рис. 1).



Рис. 1. Получение результирующей функции для шума Перлина

Аналогично, можно реализовать алгоритм, который используя эти функции, будет создавать изображения, соответствующие промежуточным октавам, а затем получить финальное изображение, соответствующее конечной функции. Оно выглядит довольно реалистично (Рис. 2) и может служить текстурой небесной поверхности при создании двумерных облаков [4].



Рис. 2. Сравнение изображения, полученного при помощи шума Перлина, и фотографии

Алгоритмы трехмерного моделирования облачности. Трехмерные модели облаков с наибольшей точностью и реалистичностью отражают их физическую структуру и позволяют визуализировать большее число происходящих в них процессов. В связи с этим существует большое количество алгоритмов, позволяющих создавать трехмерные облака.

В [1] описан алгоритм трехмерного моделирования облачности, основанный на использовании карты высот, представляющей из себя текстуру шума Перлина. На каждый пиксель карты приходится одна из градаций серого, представленная, например, числом от 0 до 255. На основе карты строится трехмерная сетка, длина и ширина которой совпадают с размерами текстуры, а высота – с некоторым заранее определенным числом. В узлах решетки находятся воксели – минимальные графические единицы, операции над которыми выполняются как над самостоятельными элементами. Они определяются координатами внутри решетки и имеют общий метод отрисовки. Для каждого вокселя предлагается использовать параметр, отвечающий за присутствие облака в соответствующей точке решетки. Это значение должно проставляться для всех элементов трехмерной сетки в соответствии с картой высот. Воздействие ветра на такую структуру, ее перемещение, изменение, можно реализовать за счет сдвига заполненных вокселей относительно трехмерной сетки.

Данная реализация основывается на описанном алгоритме, но отличается тем, что в ней не требуется хранить в памяти компьютера трехмерную решетку – для моделирования трехмерного облака используется только карта высот. Ее создание является наиболее важной задачей, поскольку именно она будет отвечать за форму и структуру облачности, а следовательно, влиять на ее реалистичность. При этом, все эффекты (изменение формы

облака, ветер) реализуются за счет перемещения по карте высот, которая, соответственно, должна иметь достаточную площадь.

Моделирование сопутствующих облачности эффектов. Что же касается эффектов, сопутствующих облачности, то одним из возможных решений этой задачи можно считать использование систем частиц. Этот способ применяется, если нужно визуализировать большое количество маленьких объектов, не имеющих четких границ [5]. И принцип его заключается в том, что все части системы имеют схожий вид, определенное время жизни, и подчиняются одному и тому же закону изменения размера, цвета, скорости и т.д. Данный способ часто используется при моделировании таких эффектов, как дым, туман, взрыв. В нашем случае это подходящий способ для моделирования осадков.

Разработка алгоритма создания карты высот. С точки зрения реализации карта высот должна представлять из себя массив значений, каждое из которых соответствует высоте облака в некоторой точке его основания. Для простоты будем говорить о ней как о двумерной текстуре. По сути же это эквивалентные понятия: числовое значение цвета каждого пикселя текстуры можно рассматривать в качестве значения высоты облака.

Чтобы создать двумерную текстуру, нужно выбрать параметры картинки, эквивалентные амплитуде и длине волны – основным параметрам, отвечающим за свойства шума Перлина. Вместо длины волны целесообразно взять площадь той части изображения, которая имеет постоянный фиксированный цвет, а аналогом амплитуды – диапазон цветов, или максимальную разность между цветами участков данного изображения.

Таким образом, первое изображение (октава), будет состоять из самых крупных по площади участков постоянного цвета, разность цветов также будет наибольшей. Для каждого следующего величина площади и разность между всеми цветами будет уменьшаться, пока не достигнет заданного значения. Цвет каждого пикселя col_{ij} результирующего изображения рассчитывается по формуле (1), где map_k – k -ое промежуточное изображение, а n – количество промежуточных изображений.

$$col_{ij} = \frac{1}{n} \sum_{k=0}^n map_k(i, j) \quad (1)$$

В данном случае для каждого участка в качестве значения $map_k(i, j)$ случайным образом выбирается значение в диапазоне от 0 до 1.

Для того, чтобы получить карту высот для перистой облачности, каждому участку октавы присваивается случайный цвет из доступного диапазона, без проверки цветов соседних участков (Рис. 3, а). Чтобы получить более плотную карту, подходящей для кучевых облаков, необходимо всякий раз, выбирая цвет для текущего участка, корректировать его так, чтоб он отличался от соседних не более чем на выбранную константу (Рис. 3, б). Для повышения реалистичности, финальная текстура дополнительно сглаживается: в качестве результирующего значения каждого пикселя изображения берется среднее арифметическое между всеми соседними с ним пикселями.

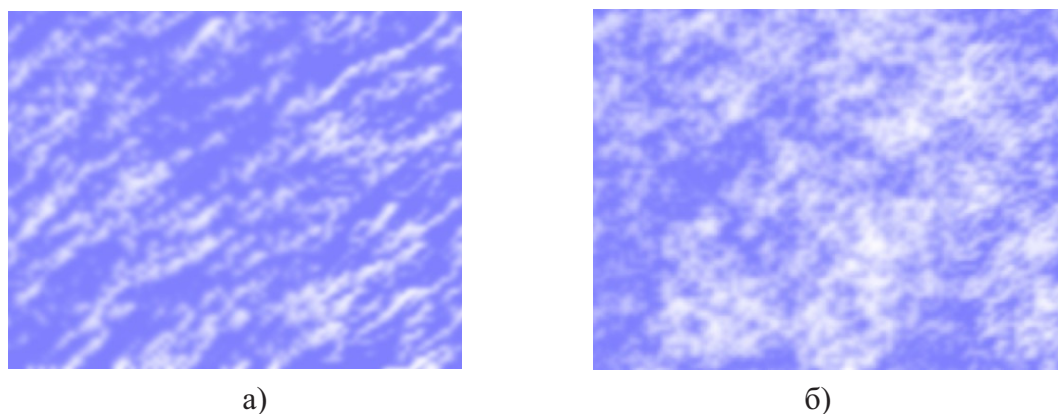


Рис. 3. Текстура для карты высот: а) для перистых облаков, б) для кучевых облаков

Разработка системы частиц для визуализации осадков. Для моделирования таких эффектов, как дождь и снег, каждую частицу удобно представить следующими параметрами: координатой зарождения, цветом, скоростью в текущий момент времени, скоростью в предыдущий момент времени и ускорением. Сама система будет представлять собой массив таких элементов. Местом зарождения всех частиц будет являться плоскость, в которой находится моделируемое облако. Соответственно, z -координата точки зарождения частицы будет совпадать с z -координатой плоскости неба. Координаты x и y зарождения частицы будут определяться случайным образом.

В общем случае изменение положения частиц в каждый момент времени можно описать следующим законом: $v_{i+1} = v_i + a \cdot t$, $z = z + v_{i+1} \cdot t$, где v – текущая скорость, v_i – скорость в предыдущий момент времени, a – ускорение, t – изменение времени. При достижении частицей поверхности земли, она будет удаляться из системы. Такое

поведение характерно для частиц дождя. Для создания снега можно действовать таким же образом, добавив при этом случайное смещение каждой частицы на небольшую константу в горизонтальной плоскости, а с помощью перемещения в заданном направлении можно смоделировать воздействие ветра.

Реализация карты высот. Функция, строящая карту, получает на вход пустую карту высот и ее размеры. В ходе выполнения она генерирует несколько промежуточных «слоев», у которых различны два параметра: размер части октавы, имеющий один цвет (частота) и максимальная возможная разница значений цвета внутри октавы (амплитуда). Количество октав задается внутри функции и влияет на детализацию. Для создания карт высот кучевых и перистых облаков реализовано две различные функции. И у каждой из них отличаются параметры частоты, амплитуды, количества октав. В частности, для функции, отвечающей за кучевые облака, наилучший с точки зрения реалистичности начальный параметр, соответствующий частоте (размер квадрата, получающего одно и то же значение внутри одной октавы), равен приблизительно 140 элементам; амплитуда – 1, а количество октав – 3. Для второй функции эти параметры будут равны соответственно: 500, 0,9 и 4.

Поскольку для корректной работы алгоритму Перлина в качестве размера карты высот требуется значение как минимум в четыре раза большее начального параметра, соответствующего частоте, можно сделать вывод, что размер карты должен быть не меньше 2000 элементов. Это также согласуется с потребностью генерировать карту высот большой площади.

Предусмотренное алгоритмом сглаживание, при указанных параметрах работы функций создания карт высот, должно применяться к каждой карте 10 – 15 раз. Этого достаточно для получения нужного визуального эффекта.

Визуализация облачности. На данный момент имеется две карты высот (для перистых и кучевых облаков), по которым можно полностью восстановить форму облака. Функция, которая выполняет это действие, получает на вход готовую карту высот, точку, начиная с которой будет выводиться облако, и размер окна, через которое будет просматриваться карта высот. По этим параметрам функция найдет нужную часть карты и для каждого ее элемента выведет в соответствующей точке экрана требуемое количество элементов. Таким образом, чтобы смоделировать эффект ветра, нужно при каждой смене кадра подавать функции на вход новые значения точки, с которой

начинается вывод. Тогда окно вывода может перемещаться по карте, например, как показано на Рис. 4.

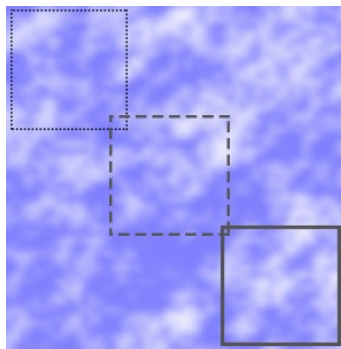


Рис. 4. Перемещение окна по карте высот

Каждый элемент облака, при этом, представляет собой три взаимно перпендикулярных четырехугольника. Наиболее реалистичного эффекта можно добиться, если каждый из них будет иметь белый полупрозрачный цвет. Чтобы это обеспечить, нужно создавать текстуры с альфа-каналом, отвечающим за прозрачность, а при загрузке текстуры в программе использовать цветовую модель RGBA.

Визуализация осадков. При реализации наиболее важным вопросом является выбор количества частиц, одновременно присутствующих в сцене и способа их отрисовки. Для изображения дождя наилучшим образом подходят небольшие треугольники, близкие к форме капель, с натянутыми на них полупрозрачными текстурами, а для изображения снега – четырехугольники, более мелкие по сравнению с частицами дождя. Все это позволяет достичь неплохого визуального эффекта.

Стоит также заметить, что при поворотах сцены, частицы представляющие собой обычные полигоны могут вырождаться в отрезки, поэтому необходимо использование панелей. Скорректировать ориентацию многоугольника можно следующим образом. Находится векторное произведение вектора направления взгляда и любого другого вектора, неколлинеарного с первым. Полученный вектор лежит в искомой плоскости. Поворачивая его нужное количество раз, получаем вершины панели, отвечающей за данную частицу.

Остальные детали реализации (такие как движение частиц в направлении к поверхности земли и воздействие на них ветра) вполне соответствуют плану визуализации, описанному при разработке, и не представляют особого интереса.

Моделирование сцены. Для оценки результатов было предложено два режима визуализации. Основной вид формируется посредством отрисовки одной из имеющихся карт высот в верхней части экрана. В таком случае есть возможность наблюдения осадков (Рис. 5, а). При этом программа использует три различных текстуры - для частиц облака, для дождя и снега.

Исходя из того, что основной целью создания данного приложения является повышение реалистичности натуральных сцен, формируется второй режим, посредством наложения получившейся облачности на геоснимок (Рис. 5, б)

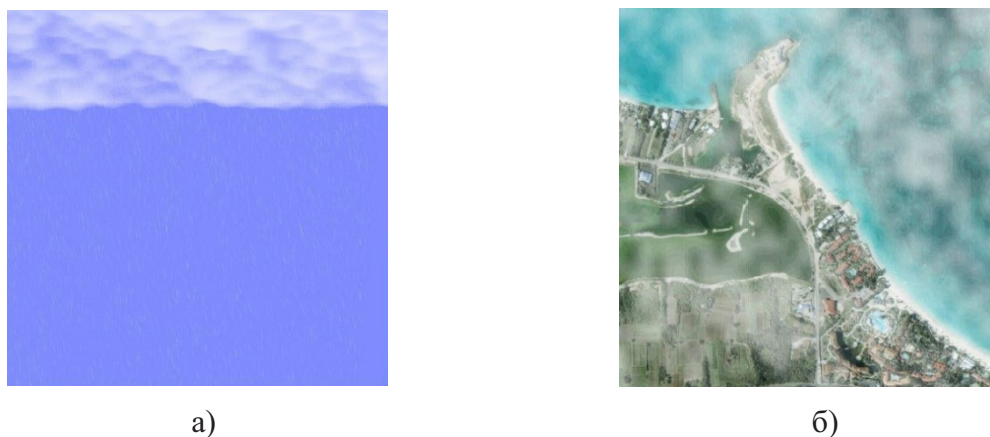


Рис. 5. Варианты сцены: а) вид с поверхности земли, б) вид сверху

Оценка производительности. Тестирование производилось на оборудовании, оснащенном процессором Intel Core i3, с тактовой частотой 1.60 ГГц, и оперативной памятью объемом 4 ГБ.

Создание карты высот – наиболее важный и наиболее требовательный к вычислительным ресурсам этап выполнения данного приложения. Возникает вопрос, возможно ли генерировать карту высот в реальном времени, то есть поддерживая скорость смены кадров равной хотя бы 25 кадрам в секунду. В таком случае на формирование каждого кадра должно тратиться не больше 0,04 секунды. Проведем несколько измерений с различными параметрами для функции построения карты высот. На Рис. 8 представлен график зависимости производительности алгоритма от количества октав, требующихся для создания карты.

Единственный раз, когда алгоритм уложился в оговоренное время приходится на количество октав, равное 1. При этом совершенно не учитывался тот факт, что для получения лучшего эффекта карту необходимо сглаживать. Оценка производительности функции сглаживания (График на Рис. 9) показывает, что при генерировании карты высот в реальном времени функция сглаживания неприменима.

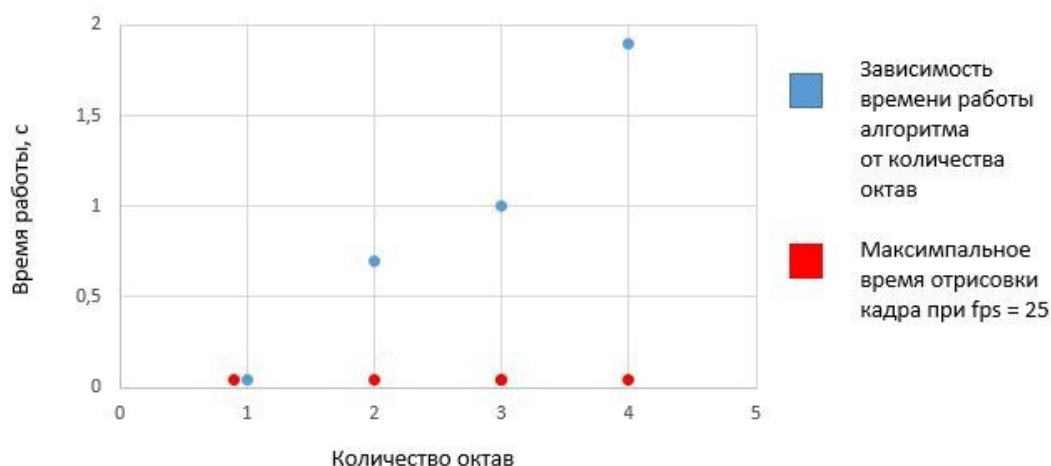


Рис. 9. Производительность алгоритма сглаживания карты высот

В случае с системой частиц интересно знать, какое количество частиц одновременно может присутствовать на сцене, не нарушая плавности анимации. Будем также пользоваться тем, что на создание каждого кадра не должно уходить больше 0,04 секунды. Измерив время, затраченное на вычисление всех параметров, связанных с системой частиц и выводом ее на экран, при различном количестве частиц, можно убедиться в том, что даже в тот момент, когда в кадре находится около 2000 частиц, измеренное время не превышает 0,009 секунды, при этом для наиболее реалистичного эффекта достаточно присутствия 500 частиц.

Заключение. В ходе данной работы было создано приложение для моделирования эффекта облачности. В его возможности входит создание карты высот для таких типов облачности как кучевая и перистая, визуализация облаков в соответствии с картой высот, создание двух видов осадков – снега и дождя, формирование двух сцен для демонстрации реалистичности полученных изображений. Оценка производительности используемых алгоритмов показывает, что они не пригодны для использования в реальном времени, либо требуют более мощных вычислительных ресурсов.

Список литературы

1. Визуализация неба и облаков. Режим доступа: http://www.graphicon.ru/oldgr/courses/cg02b/assigns/hw-5/hw5_cld.htm (дата обращения 02.03.2014).

2. Interactive multiple anisotropic scattering in clouds. Режим доступа: <http://www-evasion.imag.fr/Publications/2008/BNMBC08/clouds.pdf> (дата обращения 02.03.2014)
3. Perlin Noise. Режим доступа: http://freespace.virgin.net/hugo.elias/models/m_perlin.htm (дата обращения 02.03.2014).
4. Боресков А.В. Графика трехмерной компьютерной игры на основе OPENGL. М.: ДИАЛОГ-МИФИ, 2004. 379 с.
5. OpenGL - реализация системы частиц. Режим доступа: http://esate.ru/page/sistemy_chastits_opengl_2 (дата обращения 19.02.2014).