

Санкт-Петербургский государственный университет информационных
технологий,
точной механики и оптики

Кафедра компьютерных технологий

А.В. Тихомиров

Адаптивная модификация текстурной раскладки для
карт освещенности

Бакалаврская работа

Научный руководитель: С.Э. Володарский

Санкт-Петербург
2009

Оглавление

Введение.....	3
Глава 1. Стандартная реализация.....	6
1.1. Базовые понятия и определения.....	6
1.2. Стандартная реализация.....	9
1.3. Достоинства и недостатки стандартной реализации.....	13
1.4. Фильтрация полученных текстур.....	14
Выводы по главе 1.....	15
Глава 2. Получение градиентной информации исходных карт освещенности.....	16
2.1. <i>Wavelet</i> -преобразования.....	16
2.2. Получение градиентной информации.....	17
2.3. Реализация фильтра Добеши.....	18
2.4. Модификация фильтров Добеши и Коэн Добеши.....	21
2.5. Сравнения фильтров D4, D12, D20.....	25
Выводы по главе 2.....	26
Глава 3. Алгоритм адаптивной модификации текстурной раскладки.....	27
3.1. Основная идея алгоритма.....	27
3.2. Клеточный автомат, его определение и свойства.....	29
3.3. Адаптивная модификация текстурной раскладки.....	29
3.3.1. Реализация клеточного автомата с выделенным направлением диффузии энергии между ячейками.....	39
3.3.2. Сравнение фильтраций исходной текстуры и модифицированной текстуры.....	40
3.4. Расчет новой текстурной раскладки.....	41
Выводы по главе 3.....	46
Заключение.....	47
Список литературы.....	49

Введение

В последнее время оптимизация расчетов графических вычислений становится все более актуальной задачей. Несмотря на увеличение мощностей рабочих станций, увеличиваются требования к качеству визуальных эффектов, количеству полигонов трехмерных моделей, а также к разрешению текстур. Поэтому разрабатываются специальные алгоритмы для расчета данных на этапе их подготовки с целью уменьшения количества геометрических расчетов в режиме реального времени. Отметим также, что при этом наилучшие результаты по соотношению производительности к качеству дает комбинирование, как алгоритмов реального времени, так и предварительных расчетов.

В современных графических приложениях одной из основных задач является быстрое и качественное отображение статических теней, то есть теней, отбрасываемых неподвижными источниками света на неподвижные объекты. Для решения этой проблемы используется технология под названием *Lightmap*, что в переводе с английского означает «карта освещенности». Этот метод освещения пространства в 3D-приложениях заключается в том, что создается специальная текстура, содержащая информацию об освещенности трехмерных моделей. Такое решение значительно экономит ресурсы системы, так как графическому приложению не приходится рассчитывать освещение в режиме реального времени, но при этом оно проигрывает динамическому освещению в реалистичности изображения. Также хочется отметить, что при помощи *Lightmap* текстур можно подготавливать данные не только об освещенности трехмерных моделей сцены, но и рассчитывать такие эффекты как *Ambient occlusion*, *Global Illumination* и т. д. (*Global Illumination* – симуляция диффузного распространения света от поверхностей, что позволяет достигать эффектов «мягкости» освещения)

Примером использования комбинации динамического освещения и карт освещенности является графический движок «*Source Engine*» компании *Valve* [1].

В работе рассмотрены существующие методы расчета карт освещенности, их достоинства и недостатки. В результате анализа предметной области была поставлена и обоснована задача улучшения базовой реализации алгоритма расчета *Lightmap* текстур. Целью этой задачи является обеспечение большей детализации переходов свет-тень, что выражается в возросшем числе пикселей в зоне перехода. Улучшение достигается за счет повторного расчета карт освещенности на основе данных, полученных после обработки первоначальных *Lightmap* текстур. То есть расчет карт освещенности происходит два раза, причем после первого расчета происходит «корректировка» входных данных. Так как все эти действия производятся на этапе подготовки данных, то на скорость работы графического приложения для конечной 3D-сцены в режиме реального времени они влияния не оказывают.

Основой для решения поставленной задачи был выбран подход модификации имеющейся текстурной раскладки на основе уже рассчитанных карт освещенности. При помощи *Wavelet*-преобразований первоначальных текстур получается информация об интенсивности перехода цвета. Далее строится сеть узлов так, что каждая ячейка сети соответствует пикселю на исходном изображении. Путем эмуляции физических процессов сеть изменяется таким образом, чтобы физическая система находилась в состоянии равновесия. При этом каждая ячейка фактически представляет собой клеточный автомат.

После получения градиентных данных (данных о скорости изменения цвета пикселей) была решена задача усовершенствования алгоритма таким образом, чтобы учитывались данные о фронте тень-свет для более качественного сжатия малоинформативных частей текстуры (рис. 1).

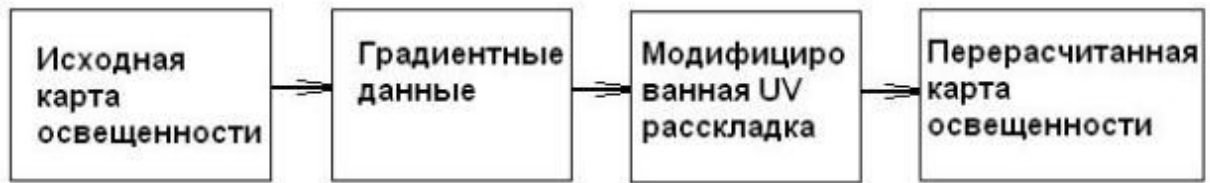


Рис. 1. Схема работы оптимизатора карт освещенности

В главе 1 выполнен обзор существующих методов расчета карт освещенности, а также рассмотрены их достоинства и недостатки.

В главе 2 выполнен разбор метода получения градиентной информации по исходной карте освещенности. Рассмотрены стандартные дискретные вейвлет преобразования и их модификации для решения поставленной задачи.

В главе 3 выполнен разбор метода модификации текстурной раскладки исходных карт освещенности. Дано определение и свойства клеточного автомата. Рассмотрена его реализация для решения поставленной задачи.

Глава 1. Стандартная реализация

Первая глава является обзором базовой реализации создания карт освещенности. Сначала даны определения сопутствующих терминов и рассмотрен базовый алгоритм. Далее приведены его достоинства и недостатки.

1.1. Базовые понятия и определения

Картой освещенности или *Lightmap* текстурой называется специальное изображение, чаще всего хранящееся в формате *LDR (Low Dynamic Range)*, но возможно хранение и в *HDR (High Dynamic Range)* форматах, которое содержит информацию об интенсивности и цвете света, падающего на модели трехмерной сцены.

В том или ином виде карты освещенности используются во всех современных *3D*-приложениях реального времени. Чаще всего этот метод применяется для создания всего статического освещения *3D*-сцены. Освещение генерируется для статической геометрии до начала цикла отрисовки изображения (рендеринга), и во время рендеринга в основном не изменяется. Фактически карта освещения содержит информацию о статических тенях, и накладывается вместе с основной текстурой, изменяя ее яркость. За счет этого и происходит имитация затененности. На современном оборудовании реализация полностью динамического освещения с использованием карт освещенности невозможна из-за большой ресурсоемкости процесса их создания. Отметим также, что данный метод освещения поддерживается такими *3D* библиотеками реального времени, как *id Tech, Source* [1], *Unreal Engine* [2], *Lithtech* и т. д.

Текстурные координаты модели (также «текстурная раскладка», «текстурная развертка») – это координаты вершины 3D-модели на текстуре (U и V соответственно), то есть отображение треугольников модели в соответствующие треугольники на текстурах. Координаты нормализованы, то есть находятся в промежутке $[0 \dots 1]$. На рис. 2 приведен пример текстурной раскладки для полигона.

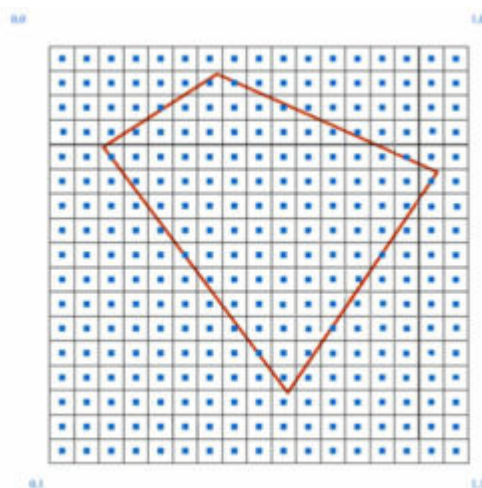


Рис. 2. Пример UV-раскладки

Ниже (рис. 3) представлена текстурная раскладка одной из карт освещенности проекта студии.

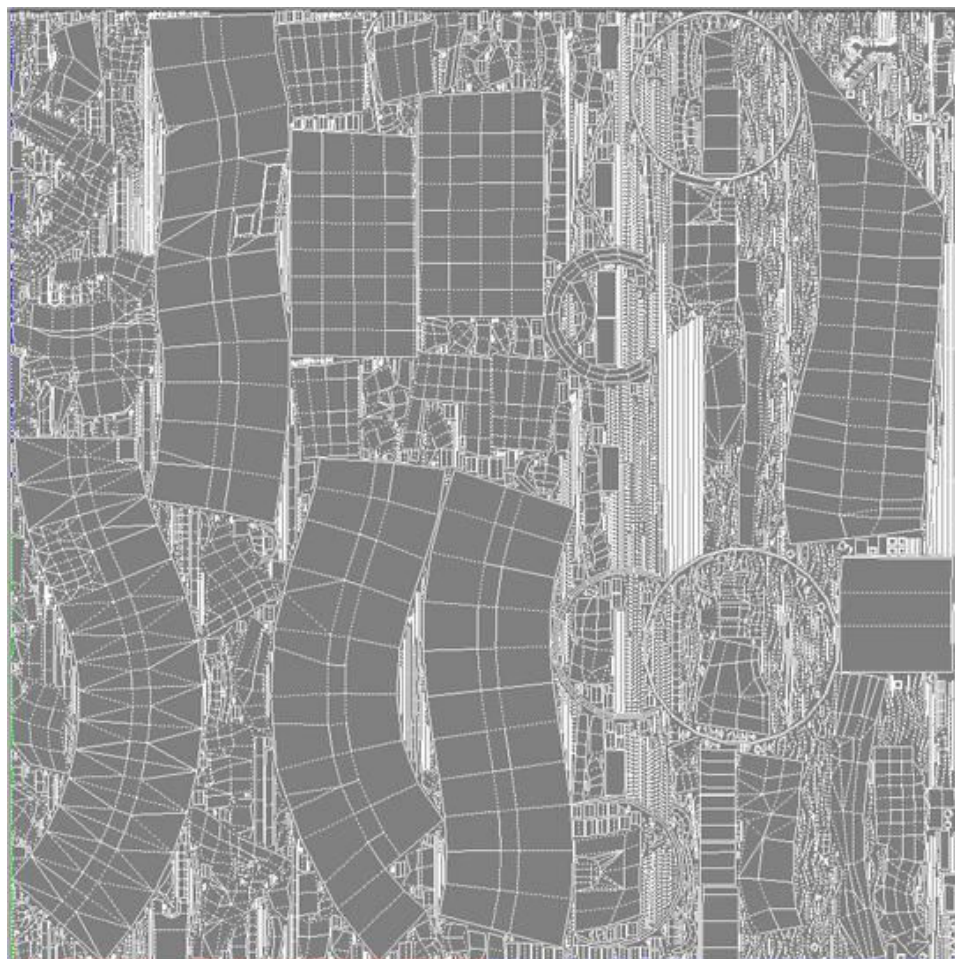


Рис. 3. Пример текстурной раскладки карты освещенности

Тексель (сокращение от англ. *Texture element*) – минимальная единица текстуры трехмерного объекта. Фактически это то же самое, что и пиксель, но отличие в том, что тексель – уже обработанный (наложенный на полигон) пиксель текстуры. Для идеального отображения текстуры число текселов должно совпадать с числом пикселей.

Люксель – тексель карты освещения (*Lightmap* текстуры).

Артефакт – ошибка в изображении, которая является следствием ошибки на этапе визуализации трехмерной сцены. Артефакты могут быть связаны как с неточными методами текстурирования или сглаживания (а так же с ошибками в коде графического движка), так и с аппаратными ошибками. В первом случае могут проявляться разнообразные искажения, связанные с методом текстурирования и последующей фильтрацией. Во втором случае на экране могут появляться разного рода ошибки: белые точки, или области,

искажение цвета, неправильное текстурирование, и другие проявления.

1.2. Стандартная реализация

Стандартная реализация применяется во множестве графических приложений, включая в себя пакеты для создания и обработки 2D и 3D изображения, такие как *Autodesk Maya* и *Autodesk 3D Studio Max*. При этом создание текстурных раскладок в основном лежит на плечах пользователя, что не позволяет создавать их близкими к оптимальным.

На тему реализации автоматической генерации карт освещенности было написано множество работ (например, [3, 4]).

Базовый алгоритм производит расчеты за один проход – освещение сцены рассчитывается один раз. При этом на этапе подготовки данных производятся трудоемкие графические вычисления. Результат представляет собой набор текстур и соответствующие текстурные координаты для моделей (*UV*-раскладка моделей).

Базовый алгоритм расчета карт освещенности, рассмотренный в работе [3] состоит из следующих этапов.

1. Создание глобальной *UV*-раскладки на текстуре, при помощи разрезания моделей на составляющие замкнутые части.
2. Расчет освещенности и теней (возможно применение различных алгоритмов расчета, например *Ray Tracing*).
3. Генерация *Lightmap* текстур по существующей координатной *UV*-раскладке и рассчитанной освещенности объектов на сцене.
4. Фильтрация полученных текстур, которая производится для скрытия графических дефектов и улучшения внешнего вида конечного изображения.

При генерации карта освещенности заполняется черными пикселями с

нулевым значением альфа-канала. Далее, для каждого люкселя (текселя карты освещенности) находятся трехмерные координаты точки на полигоне 3D-модели. Для этой точки необходимо построить список всех источников света, которые влияют на ее освещение: вектора из данной точки до источников света проверяются на пересечение с геометрией сцены, и если пересечение имеет место, то этот источник света не освещает точку (относительно него точка в тени). Остальные источники увеличивают значение текстеля карты освещенности на величину, зависящую от используемой модели освещения и положения источника света относительно точки. В целях улучшения внешнего вида картинка, к картам освещенности часто применяется билинейная фильтрация. Эти операции повторяются для каждого освещаемого полигона сцены. В итоге записываются все пиксели карты освещенности. При этом части текстуры, которые не содержат UV-раскладок 3D-моделей легко найти по значению альфа-канала. На рис. 4 представлен пример, на котором финальное изображение формируется из диффузной текстуры и карты освещенности.

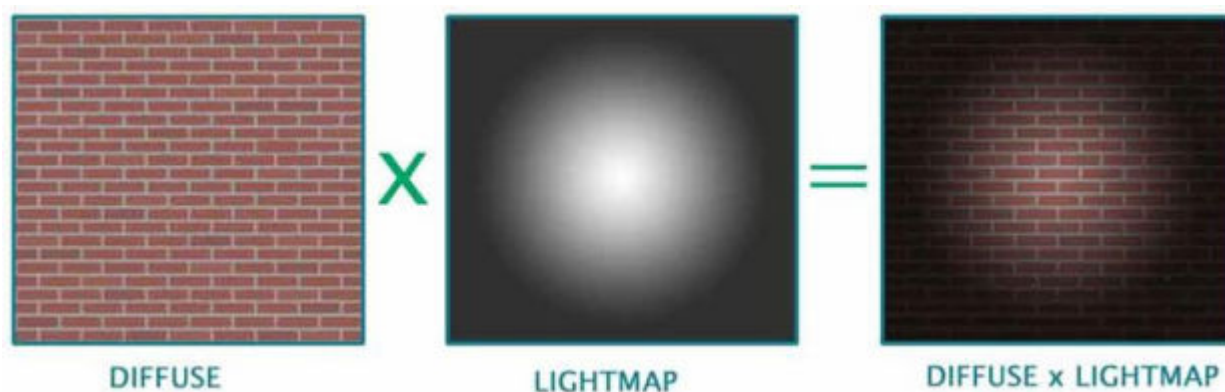


Рис. 4. Диффузная текстура, карта освещенности и конечное изображение

Ниже (рис. 5) представлены для сравнения скриншоты одного из игровых проектов студии. Сверху слева изображена сцена с точки зрения геометрии, то есть, нет источников света и диффузных текстур. Сверху справа представлена та же сцена, но с применением карт освещенности, снизу слева: с применением диффузных текстур. Изображение снизу справа является комбинацией двух предыдущих методов.

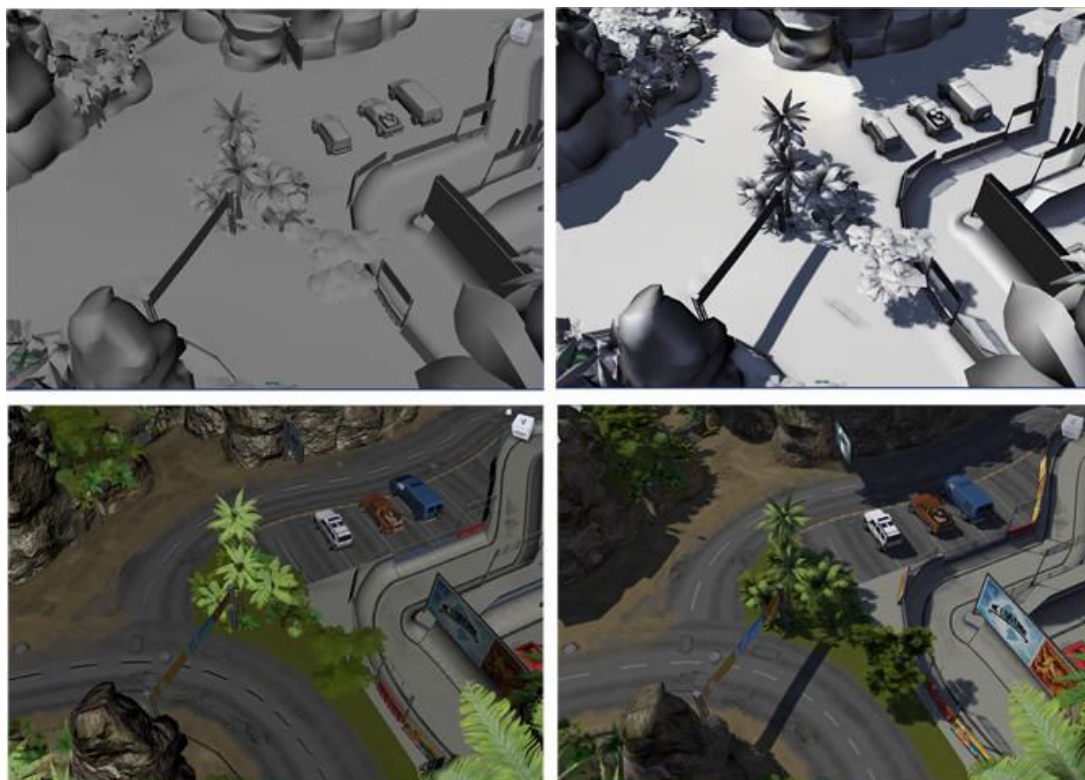


Рис. 5. Сравнение изображений 3D-сцены

Видно, что при помощи карт освещенности можно добиться более реалистичного изображения, практически без дополнительных вычислительных затрат, производимых в режиме реального времени приложения.

Также стоит отметить, что от разрешения *Lightmap* текстур сильно зависит конечное качество изображения (рис. 6). При этом возможна потеря части графической информации, что может повлечь за собой дефекты (графические артефакты). Очевидно, что большее разрешение (размер) карт освещенности даст более хорошее качество конечного изображения при всех остальных неизменных параметрах генерации *Lightmap* текстур.



Рис. 6. Зависимость результирующего изображения от разрешения «Lightmap» текстур

Также возможна комбинация текстур освещения из нескольких этапов графических вычислений. При этом создается несколько текстур, каждая из которых содержит результат расчета конкретного визуального эффекта. Одним из примеров является сочетание диффузного освещения и эффекта глобальной иллюминации, рассчитанного при помощи технологии *Ambient occlusion*.

Как видно на примере (рис. 7), технология комбинации карт освещенности позволяет добиться более качественного результирующего

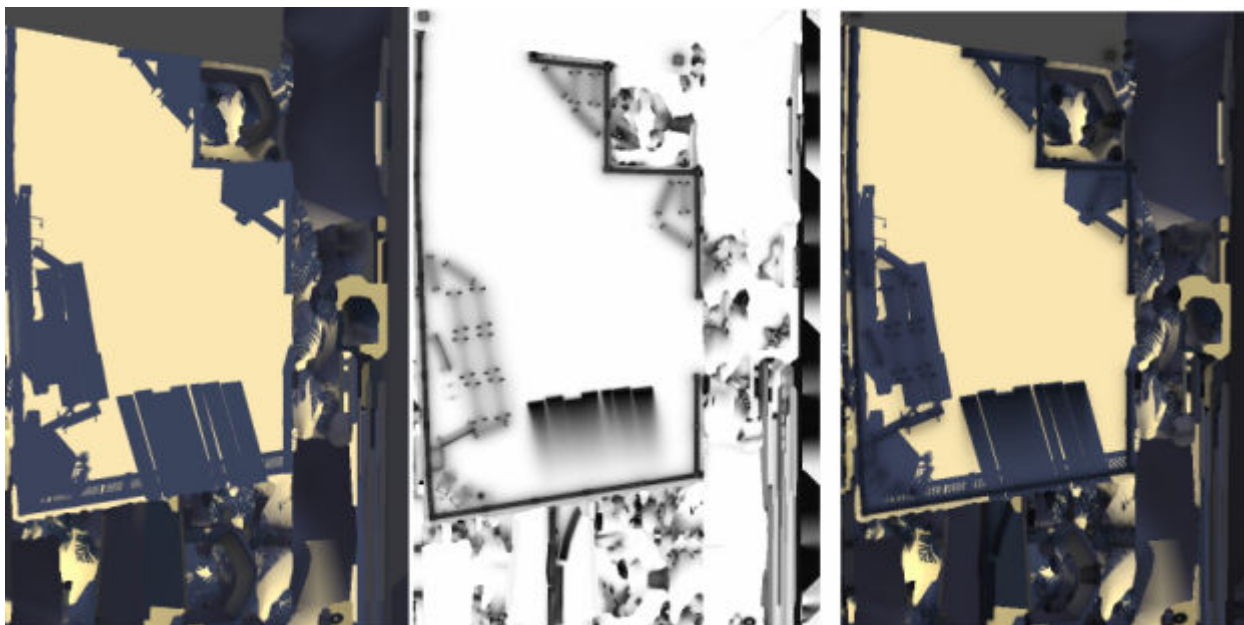


Рис. 7. Пример комбинации карт освещенности.

(слева направо: расчет диффузного освещения, расчет глобальной иллюминации, результирующее изображение)

изображения. Таким образом, можно рассчитывать и совмещать многие эффекты, которые не зависят от положения и угла камеры обзора и являются статическими.

1.3. Достоинства и недостатки стандартной реализации

Одной из основных проблем при создании карт освещенности является то, что алгоритмически сложно создать текстурную раскладку близкую к оптимальной. Особенно влияет на эту проблему то, что часто *UV*-раскладка создается конечным пользователем вручную. Отметим также, что при этом не учитывается количество и положение источников света, а также, в частности, влияние геометрии сцены на освещение конкретных полигонов *3D*-моделей. Все это в конечном итоге сказывается на качестве получаемого изображения, при этом качество имеет наивысший приоритет при расчете.

Отметим сначала преимущества стандартной технологии:

1. Достаточно простая реализация, что позволяет довольно быстро реализовать эту технологию.
2. Поддерживается большим числом стороннего программного обеспечения.
3. Меньшее время, затрачиваемое на расчет карт освещенности (однако стоит отметить, что время расчета *Lightmap* текстур в принципе не важно, так как эти операции происходят на этапе предрасчета данных).

Однако все эти преимущества влияют на скорость расчета *Lightmap* текстур, что несущественно, так как эта трудоемкость не влияет на скорость работы графического приложения для конечной *3D*-сцены в режиме реального времени.

Стандартная технология имеет следующие недостатки, которые наиболее критичны:

1. Использование не всей поверхности текстуры (неиспользованные

части занимают видео память при отрисовке).

2. Неоптимальное хранение данных (достаточно большие участки одной освещенности).
3. Не учитывается геометрия сцены, а также положение и тип источников света.

Все эти недостатки позволяет устранить построение оптимальной текстурной раскладки.

1.4. Фильтрация полученных текстур

После получения карт освещенности на практике к ним часто применяют различные фильтры для улучшения конечного визуального качества изображения.

«Билинейная фильтрация» – процесс извлечения нескольких пикселей исходной текстуры с последующим усреднением их значений для получения окончательного значения пикселя. Очевидно, что понятие «билинейная фильтрация» применимо только к двумерным текстурам.

Фильтрация может проходить в один или несколько этапов:

1. Билинейная фильтрация. Она применяется для сглаживания результирующей текстуры.
2. Фильтрация по трехмерным моделям сцены. Аналогичная фильтрация, только пиксели для усреднения берутся согласно данным в текстурной раскладке 3D-модели. То есть соседними будут пиксели. В качестве структур данных для поиска соседних точек могут использоваться «*k-d tree*», «*octree*» и другие аналогичные структуры данных.
3. Заполнение частей текстуры с нулевым значением в альфа-канале. Такая фильтрация нужна для скрытия швов на трехмерной модели в

местах стыков *UV*-раскладок.

Применение различных фильтраций позволяет не только улучшить конечное качество, но и устранить некоторые артефакты стандартной реализации расчета *Lightmap* текстур, такие как швы на стыках *UV*-раскладок, сильная пикселизация финальной текстуры, искажение изображения при использовании «трилинейной фильтрации» и многие другие дефекты.

Выводы по главе 1

Выполнен обзор стандартных методов генерации карт освещенности. Установлено, что стандартная реализация содержит недостатки, которые сказываются на качестве конечного изображения. Также установлено, что этот алгоритм не учитывает геометрию и положение источников света на сцене, что означает, что он не является оптимальным.

Улучшение алгоритма расчета карт освещенности путем адаптивной модификации текстурной раскладки устраняет недостатки стандартной реализации.

Глава 2. Получение градиентной информации исходных карт освещенности

В этой главе описан метод получения градиентной информации исходных карт освещенности. Разработана модификация фильтра для решения поставленной задачи. Дано обоснование его преимущества над фильтрами с большим числом выборок.

2.1. Wavelet-преобразования

Вейвлеты (от англ. *wavelet*), всплески (написание вэйвлеты уже почти не употребляется) – это математические функции, позволяющие анализировать различные частотные компоненты данных.

Основные свойства базовых вейвлет функций:

$$1. \int_{-\infty}^{+\infty} \varphi(t) dt = 0.$$

$$2. \int_{-\infty}^{+\infty} |\varphi(t)|^2 dt = 1.$$

Дискретные вейвлет-преобразования (ДВП) относятся к вейвлет-преобразованиям, в которых вейвлеты представлены дискретными сигналами. Эти сигналы иногда называют выборками. Часто качество фильтрации зависит от числа таких выборок, но оно также влияет на производительность.

Дискретное вейвлет-преобразование входного сигнала получают применением набора фильтров. Сначала сигнал пропускается через низкочастотный («*low-pass*») фильтр g :

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n - k].$$

Одновременно сигнал раскладывается с помощью высокочастотного

(«*high-pass*») фильтра h . В результате получаются коэффициенты аппроксимации и детализирующие коэффициенты после *low-pass* и *high-pass* фильтров соответственно:

$$y_{low}[n] = \sum_{k=-\infty}^{\infty} x[k]g[2n - k];$$

$$y_{high}[n] = \sum_{k=-\infty}^{\infty} x[k]h[2n - k].$$

Так как половина частотного диапазона сигнала была отфильтрована, то, согласно теореме Котельникова, отсчёты сигналов можно проредить в два раза. Графическая иллюстрация общей схемы дискретного вейвлет-преобразования представлена на рис. 8.

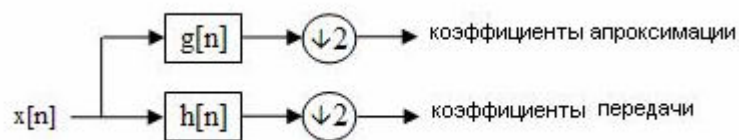


Рис. 8. Принцип работы дискретного вейвлет-преобразования

Такое разложение вдвое уменьшило разрешение по времени в силу прореживания сигнала. Однако каждый из получившихся сигналов представляет половину частотной полосы исходного сигнала, так что частотное разрешение удвоилось.

2.2. Получение градиентной информации

В работе для получения градиентной информации исходной карты освещенности использовался метод *WPD* (*Wavelet packet decomposition*). Фильтрация происходит в два этапа: сначала по столбцам, потом по строкам (рис. 9).

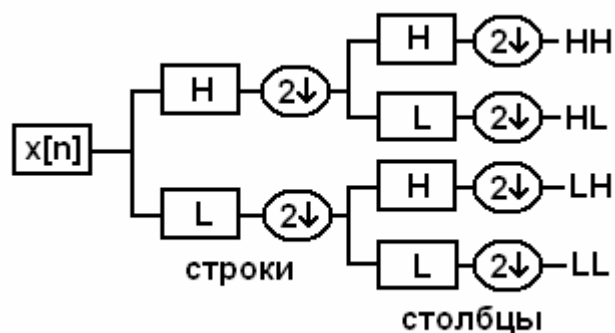


Рис. 9. Фильтрация исходной карты освещенности.

Для получения градиентной информации блок *WPD* получает на вход изображение, передаваемое в виде массива пикселей, а на выход выдает четыре матрицы коэффициентов *HH*, *HL*, *LH* и *LL* соответственно. При этом на рис. 9 блоки *L* и *H* обозначают использование фильтров *low-pass* и *high-pass* соответственно. Фильтр *low-pass* получает интенсивность низких частот, то есть просто получается сглаженное (аппроксимированное) исходное изображение, а фильтр *high-pass* получает различие между соответствующими пикселями изображения – детализирующие коэффициенты. После каждой ступени обработки полученное изображение масштабируется с уменьшением вдвое по оси *X* или *Y* в зависимости от того, к строкам или столбцам изображения применялся фильтр.

Для нашего случая не обязательно генерировать все четыре матрицы коэффициентов. Так как нас интересует только высокочастотная информация, то достаточно получить пару *HH* и *HL*-матриц соответственно. В качестве требуемого фильтра был использован фильтра Добеши. Подробнее про *WPD*-метод можно найти в работе [7].

2.3. Реализация фильтра Добеши

Сначала в качестве *wavelet*-функции применялся *haar* фильтр, однако он плохо работает с зашумленным входным изображением. Затем в качестве *wavelet*-функции был применен *D4*-фильтр (функция Добеши, англ. *Daubechies wavelets*), который лучше справляется с шумами за счет того, что

он рассматривает четыре точки (как видно из названия) с различными коэффициентами. Отметим, что фильтр *haar* является *D2* функцией, то есть всплеском Добеши второго порядка. Схема *D4* фильтра для *WPD*-метода показана на рис. 10.

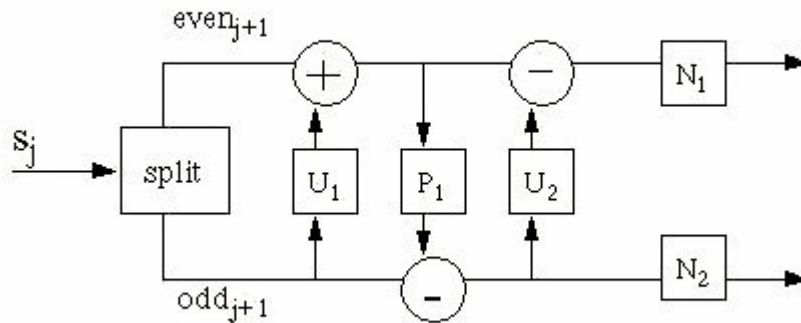


Рис. 10. Схема работы всплесков Добеши четвертого порядка

Разбиение исходного изображения на две части (процедура *split*) бывает двух видов: по столбцам и строкам (рис. 11).

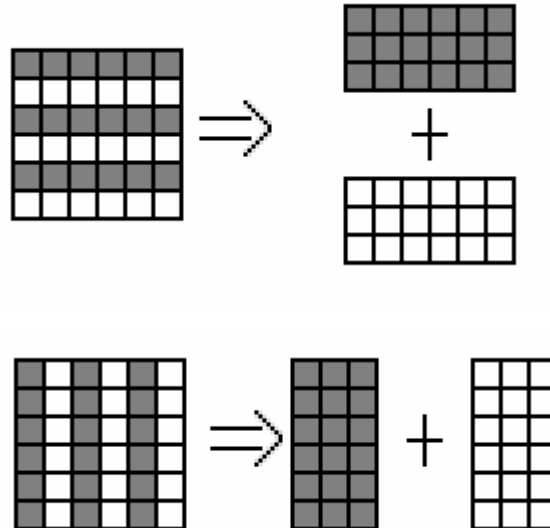


Рис. 11. Схема работы блока «*split*» (построчная и по столбцам)

При этом отметим, что процедура *split* не только подготавливает данные для работы, но и производит соответствующее масштабирование изображения. Следовательно, сначала происходит уменьшения размера отдельного изображения, а после этого производится фильтрация. Причем за

один проход происходит применение обоих фильтров, как *low-pass*, так и *high-pass*.

Фильтрация изображения проходит, как видно из блок-схемы, в четыре стадии. Блоки *update* производят расчет первой половины изображения после разделения, в которое записываются аппроксимированные коэффициенты входных данных – происходит применение низкочастотного фильтра *low-pass*.

Update 1:

for n = 0 to half - 1

$$S[n] = S[n] + \sqrt{3}S[\text{half} + n]$$

Update 2:

for n = 0 to half - 2

$$S[n] = S[n] - S[\text{half} + n + 1]$$

$$S[\text{half} - 1] = S[\text{half} - 1] - S[\text{half}]$$

Блок *predict* производит расчет второй половины изображения, в которое записывается градиентная информация относительно исходных данных. Следовательно, применяется высокочастотный фильтр *high-pass*.

Predict:

$$S[\text{half}] = S[\text{half}] - \frac{\sqrt{3}}{4}S[0] - \frac{\sqrt{3} - 2}{4}S[\text{half} - 1]$$

for n = 1 to half - 1

$$S[\text{half} + n] = S[\text{half} + n] - \frac{\sqrt{3}}{4}S[n] - \frac{\sqrt{3} - 2}{4}S[n - 1]$$

Блок *normalize* производит нормализацию полученных данных.

Normalize:

for n = 0 to half - 1

$$S[n] = \frac{\sqrt{3} - 1}{\sqrt{2}}S[n]$$

$$S[n + \text{half}] = \frac{\sqrt{3} + 1}{\sqrt{2}}S[n + \text{half}]$$

В зависимости от номера шага применяется фильтрация изображения по оси X или по оси Y .

Также были реализованы фильтры $D12$, $D16$, $D20$ – всплески Добеши с 12, 16 и 20 выборками соответственно. Однако несмотря на увеличение количества анализируемых пикселей, конечный результат фильтрации практически не отличается. Подробнее про реализацию фильтров Добеши можно найти в работе [8].

2.4. Модификация фильтров Добеши и Коэн-Добеши

Для того чтобы учитывались данные о фронте тень-свет для более качественного сжатия малоинформативных частей текстуры была произведена модификация стандартных фильтров Добеши и Коэн–Добеши. Такая модификация возможна для любого количества выборок в фильтре. Основной идеей является запись значения разных фильтров в соответствующие цветовые каналы изображения. При этом возможно хранение данных типа *float* (32 *bit*), если используется *HDR*-представление (например, формат *TIFF*) для более точных расчетов. Также возможно хранение результирующего изображения в пользовательской структуре данных.

При фильтрации исходной текстуры данные об изменении цвета относительно пикселя записываются в различные каналы результирующего изображения, в зависимости от направления. Данные по вертикали записывались в каналы r и b , а значения по горизонтали в каналы g и a соответственно. Схема такого метода представлена на рис. 12.

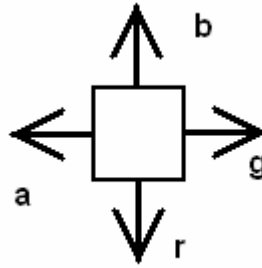


Рис. 12. Соответствие направлений фильтров и каналов результирующего изображения

Изображение подвергается обработке нескольких идентичных фильтров, отличающихся только направлением. После этого значения каждого из результатов фильтрации записываются в соответствующие каналы. Достигается большая плотность записи данных, так как нет дублирующейся информации по сравнению с черно-белым изображением в стандартной реализации фильтра. Этот подход был использован для того, чтобы при физической эмуляции процесса диффузии учитывать направление «перетекания» энергии между ячейками системы. Пример результата такого фильтра представлен на рис. 13.

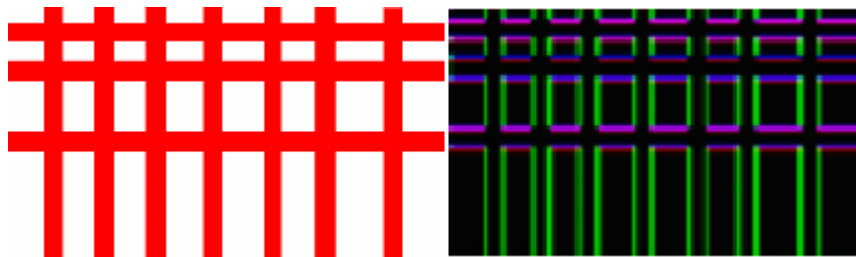


Рис. 13. Пример фильтрации изображения

При этом блок *predict* для учета относительной разницы значений, а не абсолютной изменяется следующим образом:

Predict:

$$S[half] = abs \left(S[half] - \frac{\sqrt{3}}{4} S[0] - \frac{\sqrt{3} - 2}{4} S[half - 1] \right)$$

for $n = 1$ to $half - 1$

$$S[half + n] = abs \left(S[half + n] - \frac{\sqrt{3}}{4} S[n] - \frac{\sqrt{3} - 2}{4} S[n - 1] \right)$$

Также была решена задача обработки стыков между текстурными раскладками. Для этого проверяется альфа-канал исходной карты

освещенности, при нулевом значении выявляется стык. Такие участки не обрабатываются фильтром. Они заполняются на результирующем изображении черным прозрачным цветом, то есть значения во всех каналах равны нулю. Благодаря этому при модификации текстурной раскладки можно сохранить необходимую толщину стыка в пикселях (относительно не модифицированного изображения). Это необходимо для корректной билинейной фильтрации текстуры на *3D*-объекте. Так как координата вершины в *UV*-раскладке может соответствовать стыку двух пикселей, а при фильтрации используется не только значения текущего пикселя, но и значения окружающих, то возможно смешение цвета пикселей, соответствующих разным раскладкам.

Ниже представлен результат применения *D4*-фильтра к карте освещенности одного из игровых проектов студии (рис. 14).

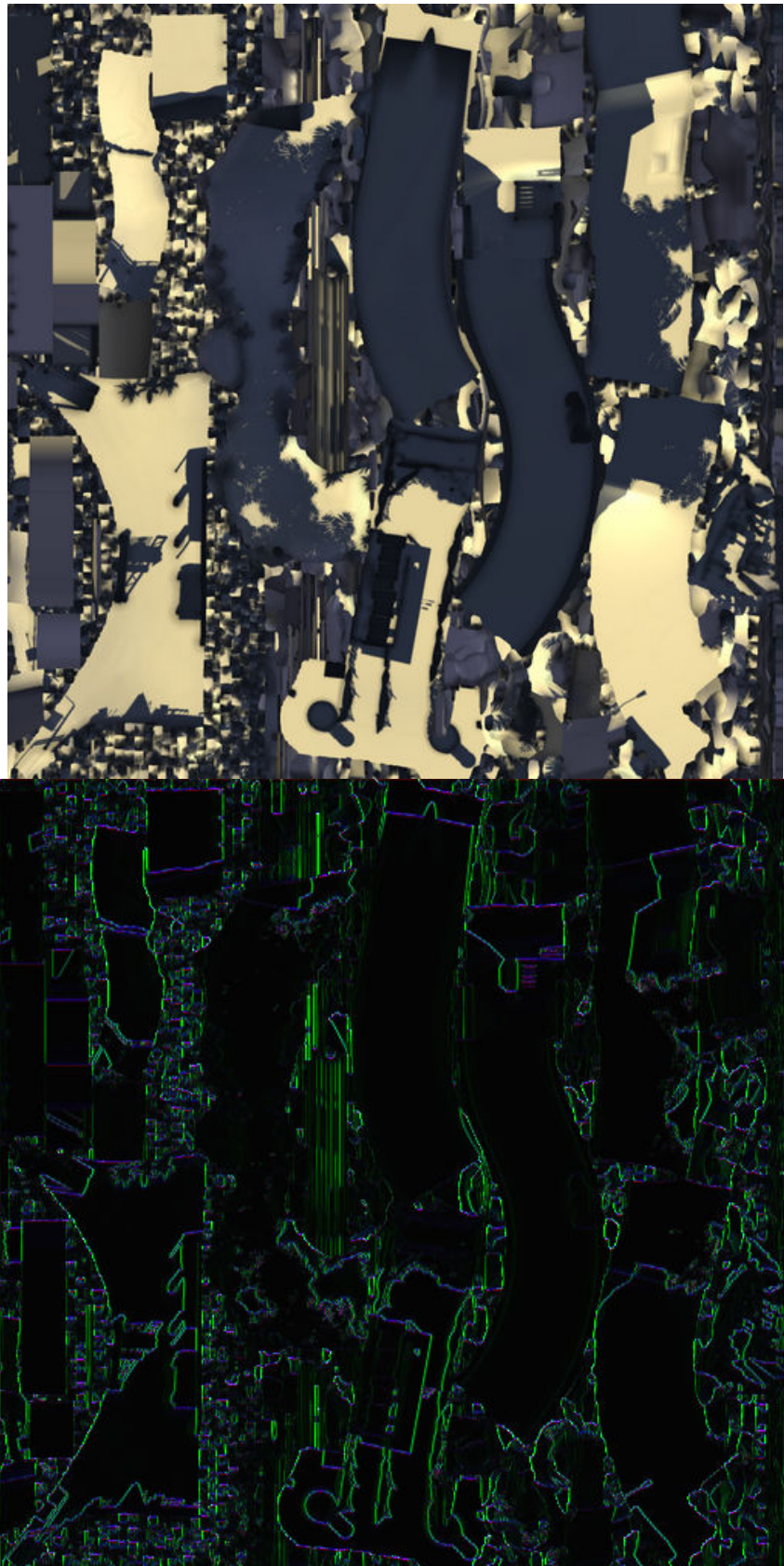


Рис. 14. Пример применения D4 фильтра к карте освещенности

Полученный фильтр учитывает не только изменения цвета пикселей изображения относительно друг друга, но и хранит информацию о

направлении этого изменения – получается градиентная информация исходного изображения.

2.5. Сравнения фильтров *D4*, *D12*, *D20*

В работе помимо фильтра *Добеши* с четырьмя выборками были реализованы фильтры *Добеши* с 12 и 20 выборками, а также *Коэн-Добеши* 9/7 и 5/3 алгоритмы фильтрации и их соответствующие модификации. Был проведен анализ изображений карт освещенности, полученных путем применения соответствующих алгоритмов фильтрации.

Из-за большого числа выборок в фильтрах *D12* и *D20* происходит смещение максимумов детализирующих коэффициентов (*wavelet function*), вследствие этого, при использовании фильтра с четырьмя выделенными направлениями фильтрации происходит «раздвоение» получаемого изображения, так как каждому исходному пикселю соответствуют разные пиксели в результирующем изображении (рис. 15).

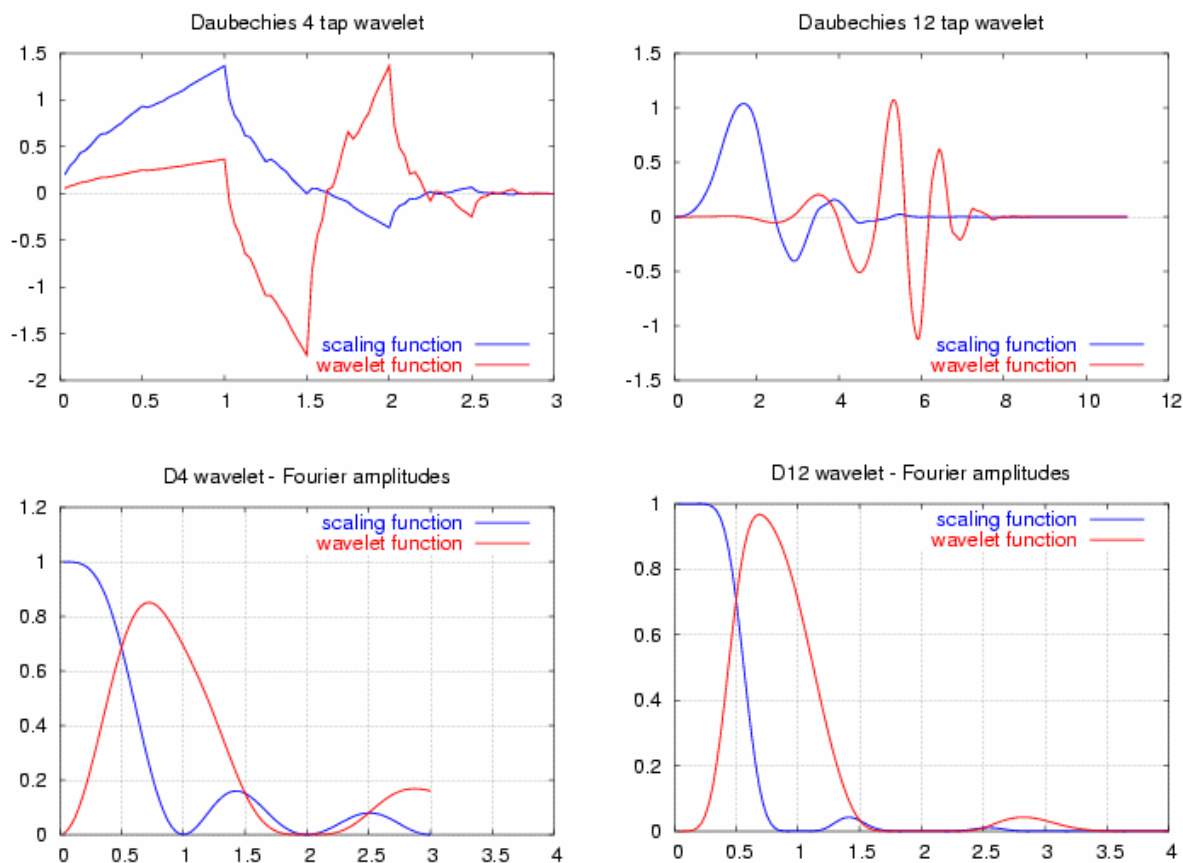


Рис. 15 Значение весов выборок и результирующая амплитуда сигнала для $D4$ и $D12$ фильтров

Также при практической реализации фильтров возникла проблема в хранении исходных данных входного изображения. Реализация фильтра $D4$, не использует дополнительной памяти, однако для фильтров $D12$ и $D20$ целесообразней оказалось дублирование входных массивов, чем написание более сложного алгоритма без дополнительной памяти.

Выводы по главе 2

Описана реализация фильтра для получения информации из исходных текстур. Разработана модификация $D4$ -фильтра, требуемая для решения поставленной задачи.

Глава 3. Алгоритм адаптивной модификации текстурной раскладки

В этой главе описан способ адаптивной модификации текстурной раскладки исходной карты освещенности. Разобраны основные алгоритмы и методы, используемые в работе.

3.1. Основная идея алгоритма

Как следует из выводов главы 1, стандартная реализация построения карт освещенности имеет ряд серьезных недостатков. Наиболее значимым из них является то, что алгоритм не учитывает геометрию, положение и тип источников света при создании текстурной раскладки обрабатываемой 3D-сцены. Это сильно сказывается на конечном качестве теней, в особенности на переходы свет – тень. В силу этих недостатков возникла необходимость в создании более оптимального алгоритма, который бы устранял недостатки стандартной реализации.

Основной идеей является модификация текстурной раскладки на основе уже рассчитанных карт освещенности. Это позволяет провести анализ освещения 3D-объектов с точки зрения геометрии, что позволяет устранить недостатки стандартного метода генерации *Lightmap* текстур. При этом улучшенный алгоритм использует стандартную реализацию как одну из своих составных частей.

Ниже представлена (рис. 16) схема алгоритма расчета карт освещенности методом адаптивной модификации текстурной раскладки.

Улучшенная реализация алгоритма:

1. Стандартная реализация.

2. Модификация UV -раскладки для улучшения качества.
3. Перерасчет освещенности и теней, на основе модифицированных текстурных раскладок.
4. Комбинирование карт освещенности, содержащих расчеты различных эффектов.



Рис. 16. Схема улучшенного алгоритма

Как видно из предложенной выше схемы, базовый алгоритм используется дважды – для первичной генерации *Lightmap* текстур и для перерасчета $3D$ -сцены с модифицированными текстурными раскладками. При этом суть алгоритма заключается в том, производится дополнительный анализ полученных результатов первого «прохода», что позволяет применять данную технологию в качестве оболочки над стандартным инструментарием в большинстве пакетов $3D$ -моделирования.

Исходя из корректности модифицирования UV -раскладки докажем корректность алгоритма (доказательство корректности преобразований UV -раскладки представлено в разд. 3.4). В качестве метода доказательства будем использовать метод математической индукции. Стандартная реализация расчета карт освещенности является базой индукции. В качестве предположения возьмем UV -раскладку, полученную в результате работы алгоритма. Так как это преобразование является верным, то алгоритм является корректным по методу математической индукции.

3.2. Клеточный автомат, его определение и свойства

Клеточный автомат – набор клеток, образующих некоторую периодическую решетку с заданными правилами перехода, определяющими состояние клетки в следующий момент времени через состояние клеток, находящимися от нее на расстоянии не больше некоторого, в текущий момент времени. Как правило, рассматриваются автоматы, где состояние определяется самой клеткой и ближайшими соседями. В качестве решетки обычно рассматривается кубическая решетка [9].

Следовательно, клеточный автомат является дискретной динамической системой, поведение которой полностью определяется в терминах локальных зависимостей.

Отметим основные свойства классической модели клеточных автоматов.

1. Локальность правил: на новое состояние клетки могут влиять только элементы ее окрестности и, возможно, она сама.
2. Однородность системы: ни одна область решетки не может быть отличена от другой по каким-либо особенностям правил.
3. Множество возможных состояний клетки – конечно. Это условие необходимо, чтобы для получения нового состояния клетки требовалось конечное число операций.
4. Значения во всех клетках меняются одновременно, в конце итерации, а не по мере вычисления. В противном случае порядок перебора клеток решетки, при совершении итерации, существенно влиял бы на результат.

3.3. Адаптивная модификация текстурной раскладки

При помощи дискретных вейвлет преобразований были получены градиентные данные исходных карт освещенности. Следующей задачей является модификация первоначальной текстурной раскладки для увеличения детализации переходов свет–тень.

Основой для решения поставленной задачи был выбран клеточный автомат. Этот метод часто применяют для решения задач типа «диффузия – конвекция». Особенности решения задач диффузии при помощи клеточных автоматов описаны в работах [10, 11]. При помощи этого метода решается задача диффузии энергии между клетками. Каждому пикселю обработанного изображения соответствует своя ячейка. На этапе инициализации задается значение начальной энергии, которое зависит от интенсивности цвета пикселя. Значение энергии может лежать в диапазоне [0..1].

Далее производится процесс передачи (диффузии) энергии между ячейками до тех пор, пока максимум разности энергий между соседними клетками больше некоторого значения.

Для расчета процесса диффузии используется две дублирующие матрицы клеток. Первый массив используется для получения значений при расчете очередного шага итерации, второй для хранения промежуточных результатов. В конце итерации происходит обмен данными между массивами.

Алгоритм 1. Шаг расчета процесса диффузии энергии

Дано: cellData — клеточных автоматов, cellDataStore – матрица для хранения промежуточных результатов

Надо: произвести шаг расчета

```
1: void forwardStep() {
2:     for (int i =0; i < width; i++) {
3:         for (int j = 0; j < height; j++){
4:
5:             cellData[i][j].forwardStep(cellDataStore[i][j],      deltaX,
deltaY);
6:         }
7:     for (int i =0; i < width; i++) {
8:         for (int j = 0; j < height; j++){
9:             cellDataStore[i][j].ReCalcSquare;
10:        }
11:    }
12:    _exchangeData(cellData, cellDataStore);
13:}
```

Этот подход позволяет реализовать четвертое свойство клеточных автоматов: значения во всех клетках меняются одновременно, в конце итерации. Благодаря этому свойству значение шага расчета не зависит от

выбора начальной ячейки, а также от способа перебора клеток, но требует дополнительных затрат памяти для хранения матрицы клеток с промежуточными результатами (рис. 17).

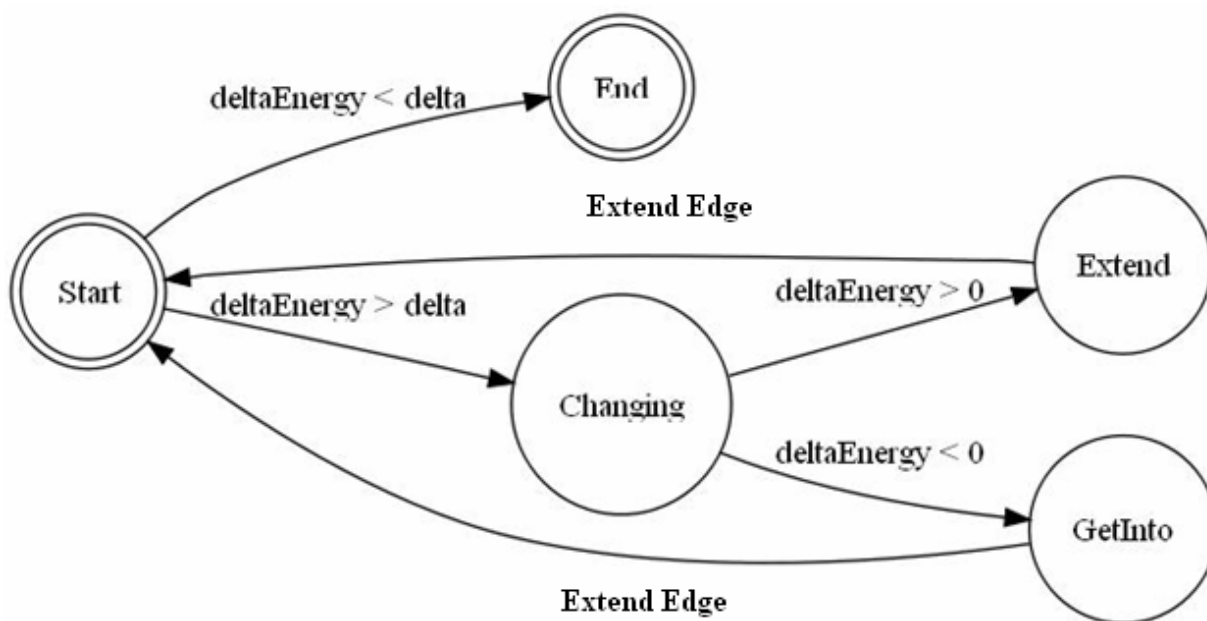


Рис. 17. Схема работы системы

У используемого автомата конечное число состояний, при этом выполняется свойство локальности и однородности системы.

При работе клеточного автомата, на клетку оказывают влияние соседствующие с ней клетки. Так как клеточный автомат является квадратным, то есть клетка имеет форму квадрата, то на нее влияют восемь соседних клеток. Но при расчетах учитываются только те клетки, которые являются соседними по вертикали и горизонтали. Диагональные клетки на текущую клетку влияния не оказывают. Такой подход не влияет на конечный результат, так как при расчете системы модификация значений энергии клеток производится во временную матрицу клеток. Следовательно, на текущую клетку влияют только четыре соседних клетки.

Варианты воздействия окружающих клеток на текущую ячейку представлены на рис. 18, 19.

- Энергия клетки меньше, чем у текущей
- Энергия клетки равна энергии текущей клетки с погрешностью delta
- Энергия клетки больше, чем у текущей

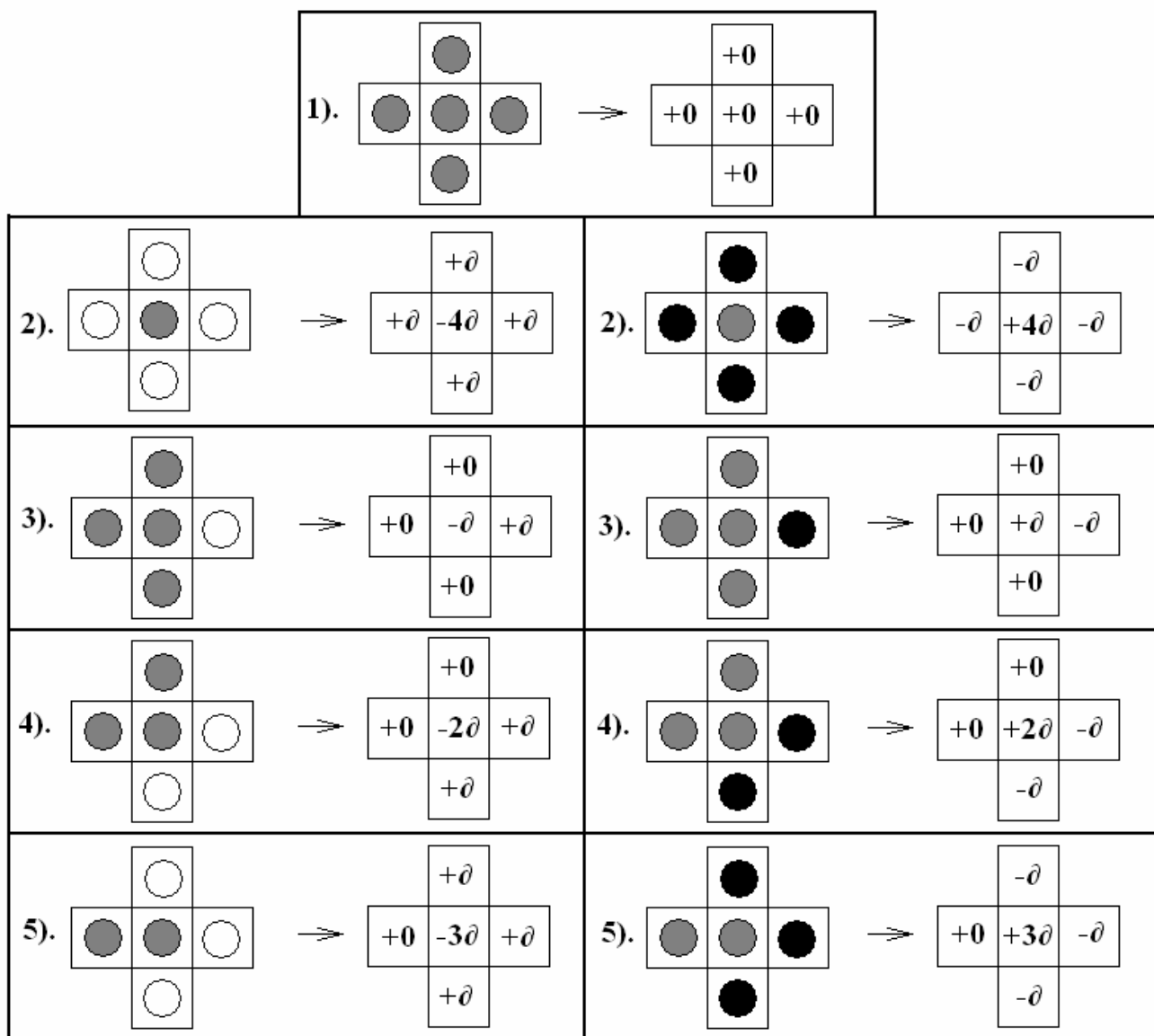


Рис. 18. Варианты состояний клеточного автомата

По представленной выше схеме видно, что в такой системе клеточных автоматов будет выполняться закон сохранения энергии, так как во всех состояниях клеточного автомата суммарная энергия клеток не изменяется.

- Энергия клетки меньше, чем у текущей
- Энергия клетки равна энергии текущей клетки с погрешностью delta
- Энергия клетки больше, чем у текущей

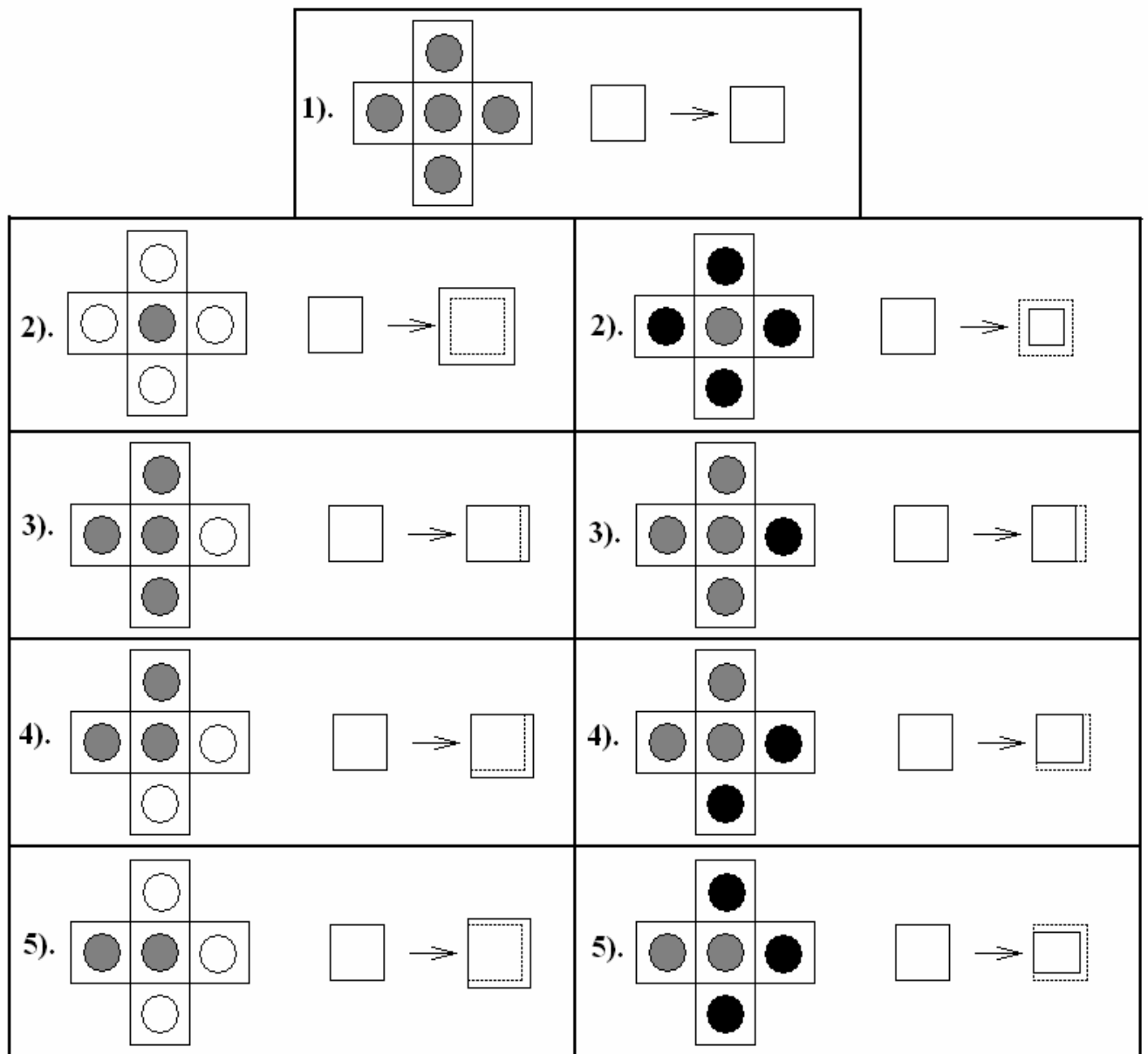


Рис. 19. Варианты геометрических модификаций клетки клеточного автомата

На схеме представлено поведение клетки при условии, что энергия одной или нескольких из соседних клеток меньше, чем энергия текущей ячейки, или соответственно больше. На схеме не представлены пространственные инварианты состояний клеточного автомата. Очевидно, что все варианты состояний можно получить сочетанием различных типов

соседних клеток (тип определяется соотношением энергии соседней клетки с текущей клеткой) и соответствующих поворотов.

Выделим требования к реализации конечного автомата с точки зрения корректности физического процесса диффузии:

1. Для каждой клетки должна быть обеспечена корректность координат ее узлов с точки зрения геометрии. При этом наличие «вывернутых» ячеек невозможно.
2. Коэффициент передач энергии должен зависеть от соотношения объемов взаимодействующих клеток. Это необходимо для того, чтобы учитывать «плотность» энергии клетки.
3. Коэффициент передачи энергии на конкретном шаге расчетов должен быть достаточно маленьким для плавности процесса. Также это повышает точность критерия стабилизации системы. Возможно использование динамического коэффициента передачи энергии между клетками для сокращения числа шагов расчетов.
4. Процесс, протекающий в системе должен быть конечным, то есть процесс передачи энергии между соседними клетками должен приводить к уменьшению энтропии в системе.
5. В системе должны отсутствовать физически некорректные значения, такие как отрицательная энергия клетки.
6. Передача энергии должна производиться от клетки с большим значением к клетке с меньшим значением энергии, то есть для протекающего процесса должно соблюдаться требование, соответствующее второму началу термодинамики.

Ниже представлен пример (рис. 20) работы системы клеточных автоматов размерности 3×3 . Во всех клетках, кроме центральной значение начальной энергии равно 1.0, а в центральной клетке энергия равна 0.0. Это выражается в цвете клетки: белый цвет соответствует условной единице, а черный – нулю.

Из рис. 20 видно как происходит плавная передача энергии центральной клетке, при этом у нее меняется цвет и производится сжатие окружающими ее ячейками. Также видно, что в течение процесса происходило изменение цвета окружающих клеток (цвет стал более серым), что обусловлено уменьшением их энергии.

Также важно отметить симметричность процесса, выражающуюся в том, что конечный результат не зависит от выбора начальной клетки расчета и алгоритма обхода клеток системы.

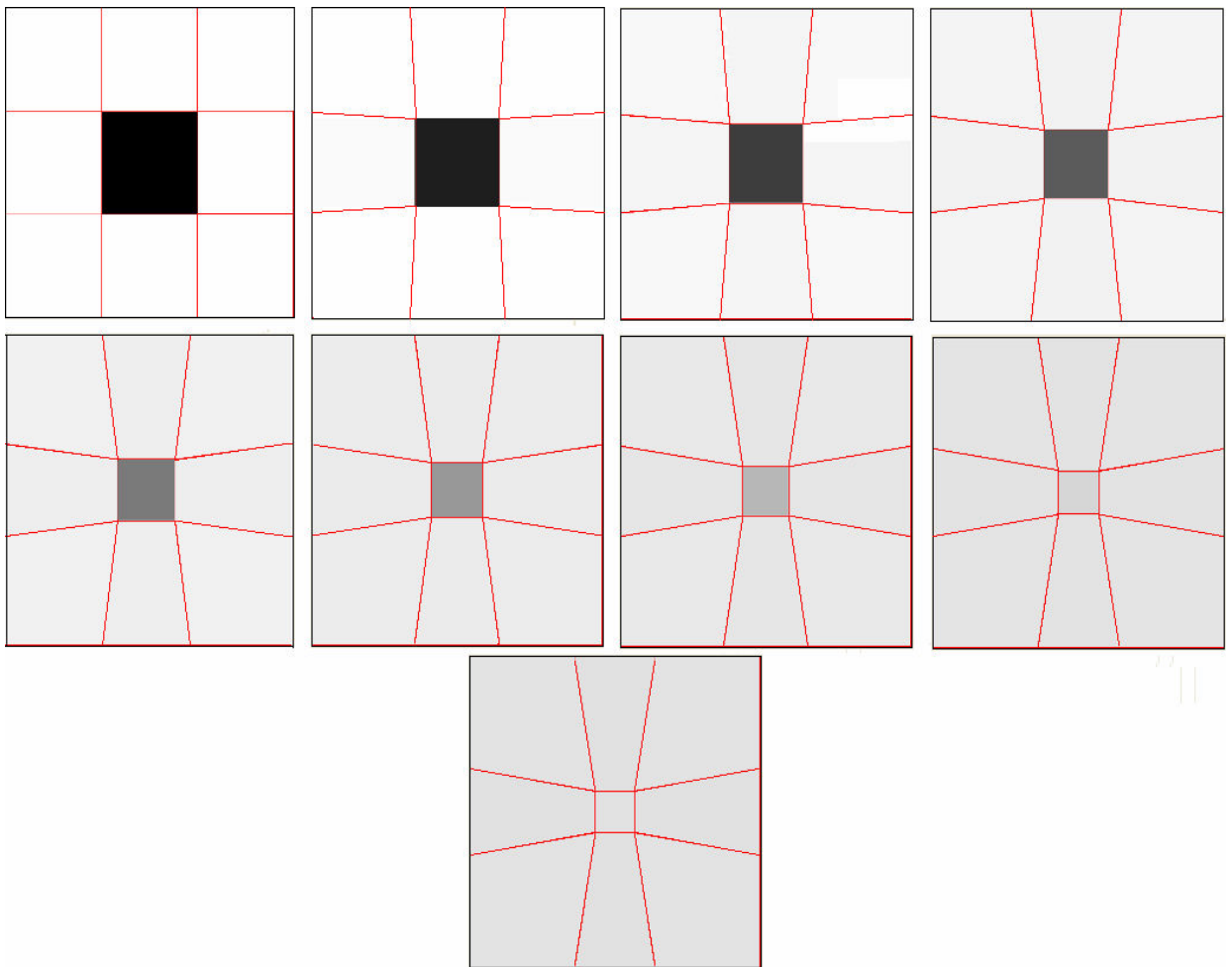


Рис. 20. Пример работы системы клеточных автоматов 3x3 ячейки (между изображениями 10 циклов)

Одним из условий корректности расчетов является то, что координаты вершин получившейся текстурной раскладки должны лежать в промежутке $[0..1]$. Поэтому граничные узлы не могут выходить за пределы – система считается адиабатической. Для того чтобы исключить смещение узлов за границы исходной области считается, что разница энергии граничных клеток

и окружающей среды равна нулю. Этот алгоритм расчета граничных клеток в сочетании с вариантами состояний клеточного автомата позволяют реализовать закон сохранения энергии в системе.

Для обработки граничных клеток было применено три подхода:

1. Граничная клетка ничем не отличается от остальных клеток системы. При этом правило однородности системы не нарушается, но возможно искажение граничных ячеек вследствие того, что для них применяется меньшее количество итераций передачи энергии.
2. Для граничных клеток применяется другой алгоритм расчета сдвига ребер ячейки. При этом нарушается правило однородности системы.
3. В системе создается дополнительное кольцо клеток – если в исходной системе было $N \times M$ ячеек, то в модифицированной системе будет $(N + 1) \times (M + 1)$ ячеек. После стабилизации системы производится коррекция координат узлов исходных граничных ячеек для соответствия диапазону $[0..1]$. Дополнительное кольцо клеток в результирующую сетку не включается.

Для тестов был выбран участок сцены (рис. 20).

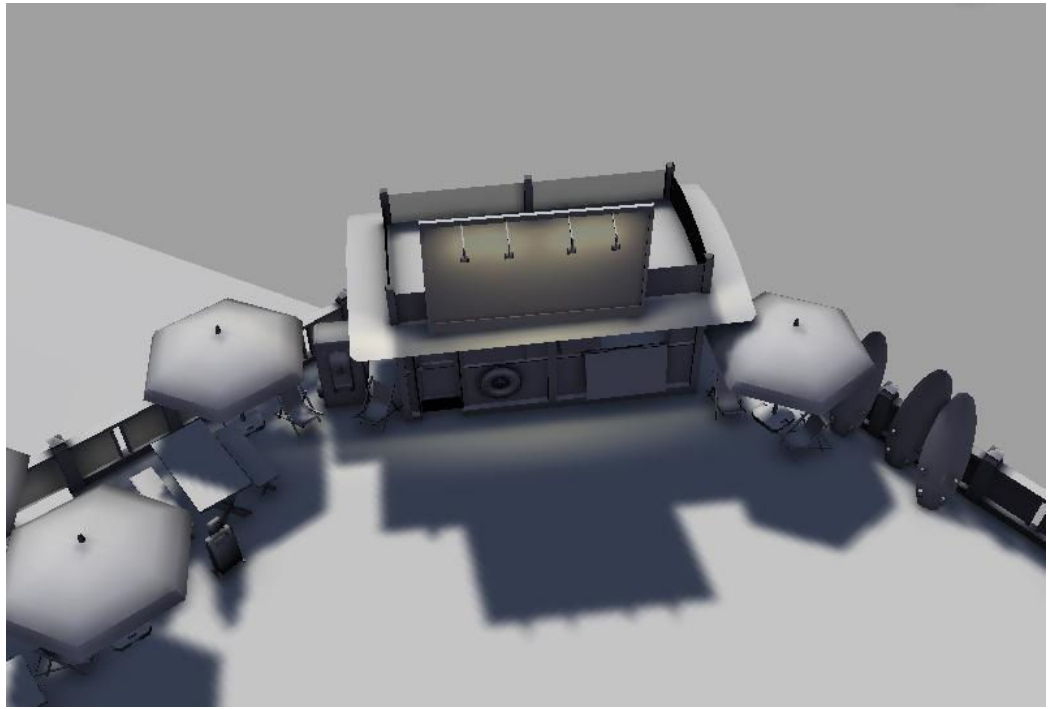


Рис. 20. Тестовый участок сцены

Ниже представлен (рис. 21) пример работы системы клеточных автоматов, построенных на основе участка карты освещенности.

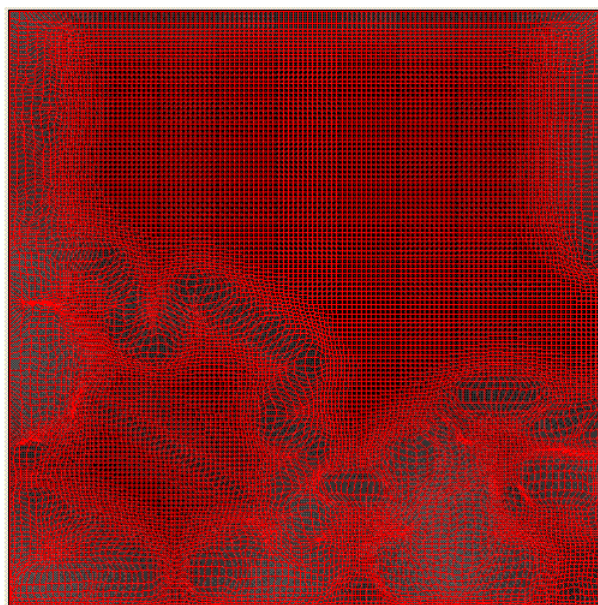
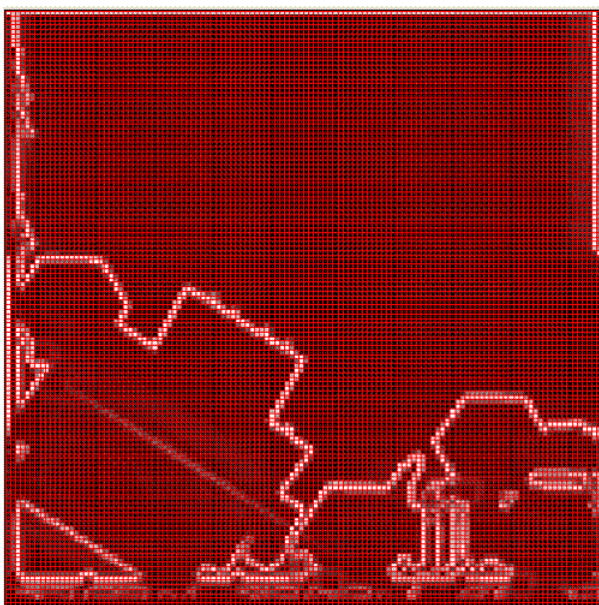


Рис. 21. Пример работы системы клеточных автоматов (слева начальная система, справа система после стабилизации)

Как видно из рис. 21, разрешение переходов свет – тень увеличилось от трех до восьми раз (а зависимости от начальных коэффициентов). Также произошло сжатие малоинформативных частей текстуры – частей с практически одинаковым цветом.

Следует учитывать также, что части исходных текстур с нулевым значением в альфа-канале трансформироваться не будет, для них применяется только модификация сдвига.

3.3.1. Реализация клеточного автомата с выделенным направлением диффузии энергии между ячейками

Реализован клеточный автомат, который хранит четыре разных значения начальной энергии ячейки, в зависимости от направления (то есть из каналов r , g , b , a обработанного изображения). При этом считается, что в каждой клетке содержатся четыре «пружины». Значения коэффициентов жесткости «пружин» задаются на этапе инициализации системы. Очевидно, что при такой реализации диффузия энергии происходит только между клетками определенных направлений. При этом в такой системе происходит нарушение законов сохранения энергии для корректности трансформации ячеек с точки зрения геометрии.

Сам клеточный автомат по принципу работы и количеству вариантов переходов не отличается от клеточного автомата, рассмотренного ранее. Отличается только алгоритм расчета разности энергий между соседними клетками системы. Разность энергии берется по каждому из направлений отдельно, при этом рассматриваются энергии пружин, соответствующих рассматриваемому стыку между ячейками. Пример работы такой системы представлен на рис. 22.

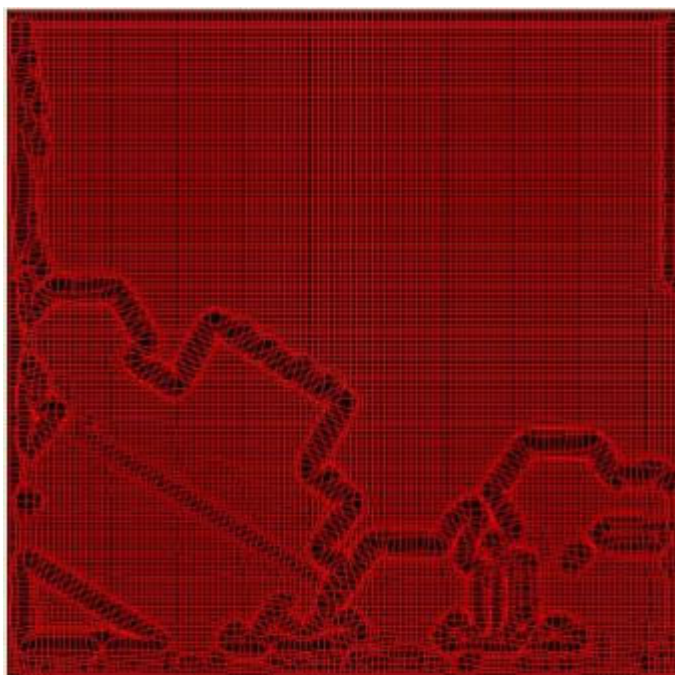


Рис. 22. Пример работы системы клеточных автоматов с четырьмя каналами

3.3.2. Сравнение фильтраций исходной текстуры и модифицированной текстуры

Представлен результат применения *D4*-фильтра к двум изображениям: исходной текстуры и текстуры, полученной из исходной путем применения системы клеточных автоматов и проецирования на регулярную сетку.

На рис. 22 показано преимущество предлагаемого метода по сравнению со стандартной реализацией. Показано, что границы перехода свет-тень стали больше по площади. В следствие изменение цвета на участке перехода стало более плавным – величина градиента стала меньше.

Для того, чтобы показать описанное выше свойство, к полученным текстурам был применен модифицированный *D4*-фильтр. Результатом работы фильтра является массив данных для каждого пикселя изображения, в котором хранится градиентная информация, то есть скорость изменения и направление изменения цвета пикселей.

Из рис. 23 видно, что результат фильтрации модифицированной текстуры содержит менее интенсивные цвета, чем результат фильтрации исходной текстуры. Также заметно, что границы свет-тень стали более «размытыми», что и означает более плавное изменение цвета пикселей.

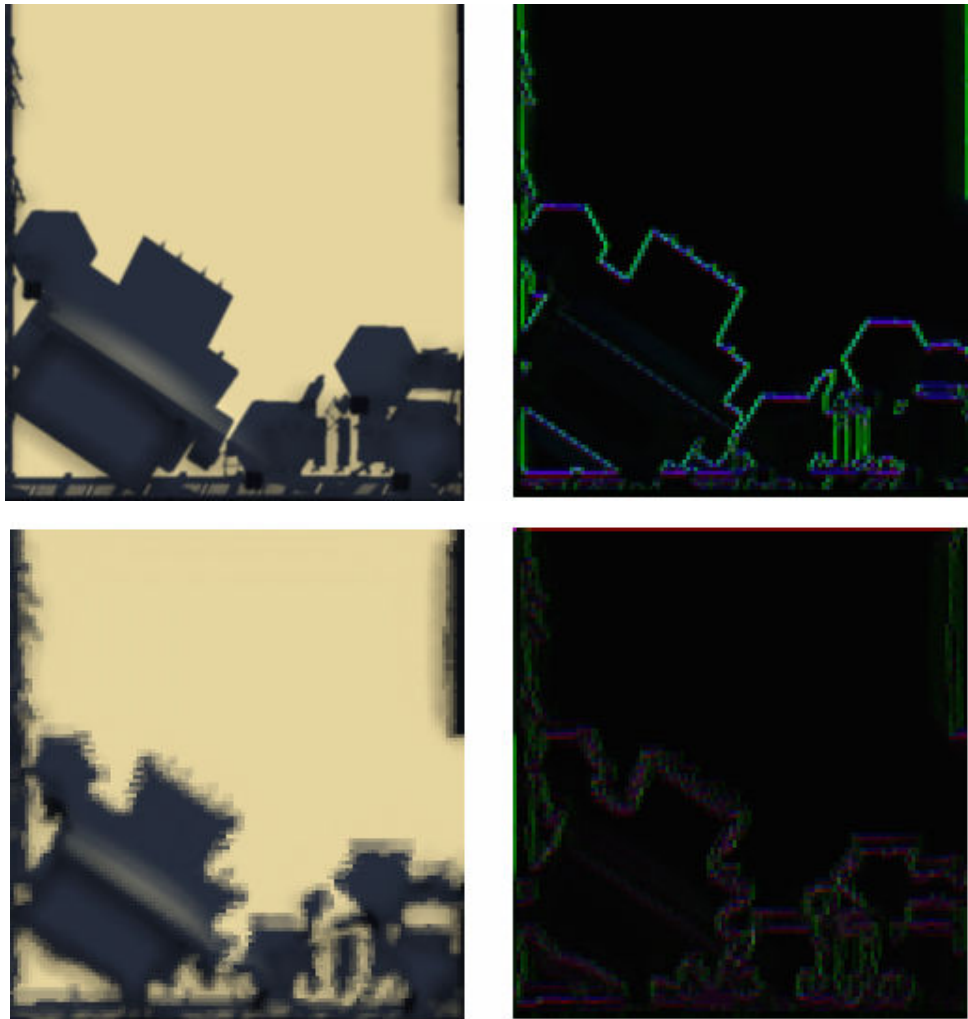


Рис. 23. Сравнение результатов работы $D4$ -фильтра, примененного к исходной текстуре и к текстуре, полученной после работы системы клеточных автоматов.

3.4. Расчет новой текстурной раскладки

После стабилизации системы производится расчет новых UV -координат соответствующих вершин $3D$ -моделей. При этом составляется взаимнооднозначное соотношение координат вершин в исходной и полученной текстурной раскладке. Таким образом, каждый элементарный объект UV -раскладки (полигон) после расчетов отображается в соответствующий полигон в новой системе координат (рис. 24).

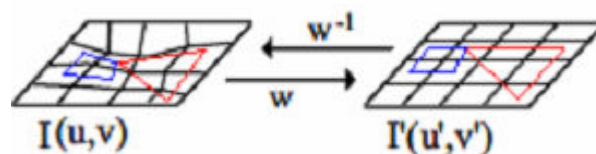


Рис. 24. Соответствие полигонов

Так как при расчетах гарантируется, что для каждой клетки сохраняется корректность координат ее узлов с точки зрения геометрии, то отображение полигонов из соответствующих систем координат является корректным. По сути, это преобразование является оператором из одного метрического пространства в другое.

На рис. 25 представлена трансформация части текстурной раскладки на примере нескольких полигонов.

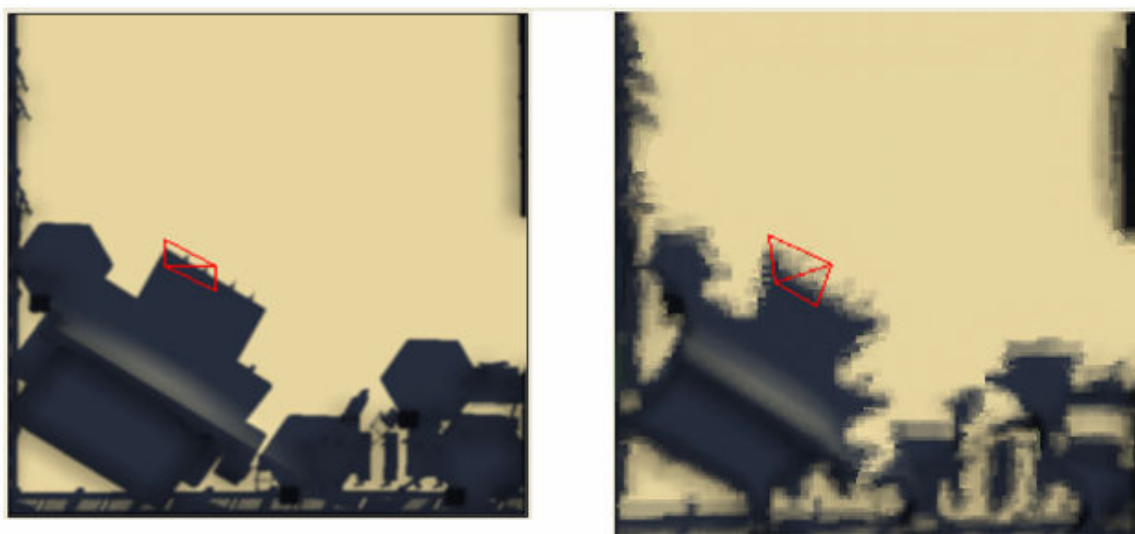


Рис. 25. Пример трансформации текстурной раскладки

Очевидно, что конечный прирост качества от рассмотренного в работе метода сильно зависит от первоначального разбиения $3D$ -модели на полигоны, а также от количества полигонов $3D$ -модели (рис. 26).

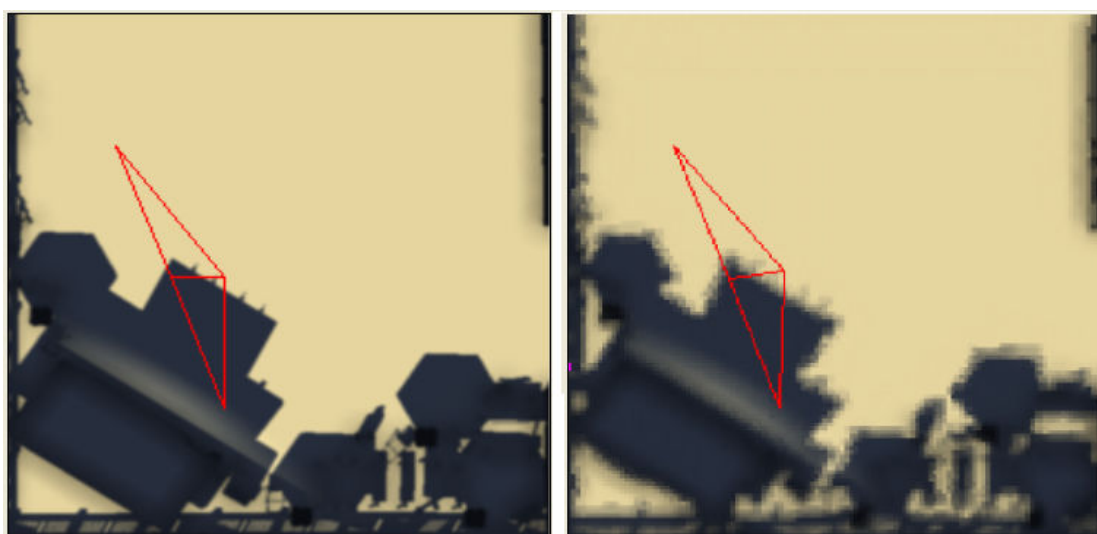


Рис. 26. Пример трансформации текстурной раскладки

Ниже представлен результат применения алгоритма к текстурной раскладке (рис. 27).

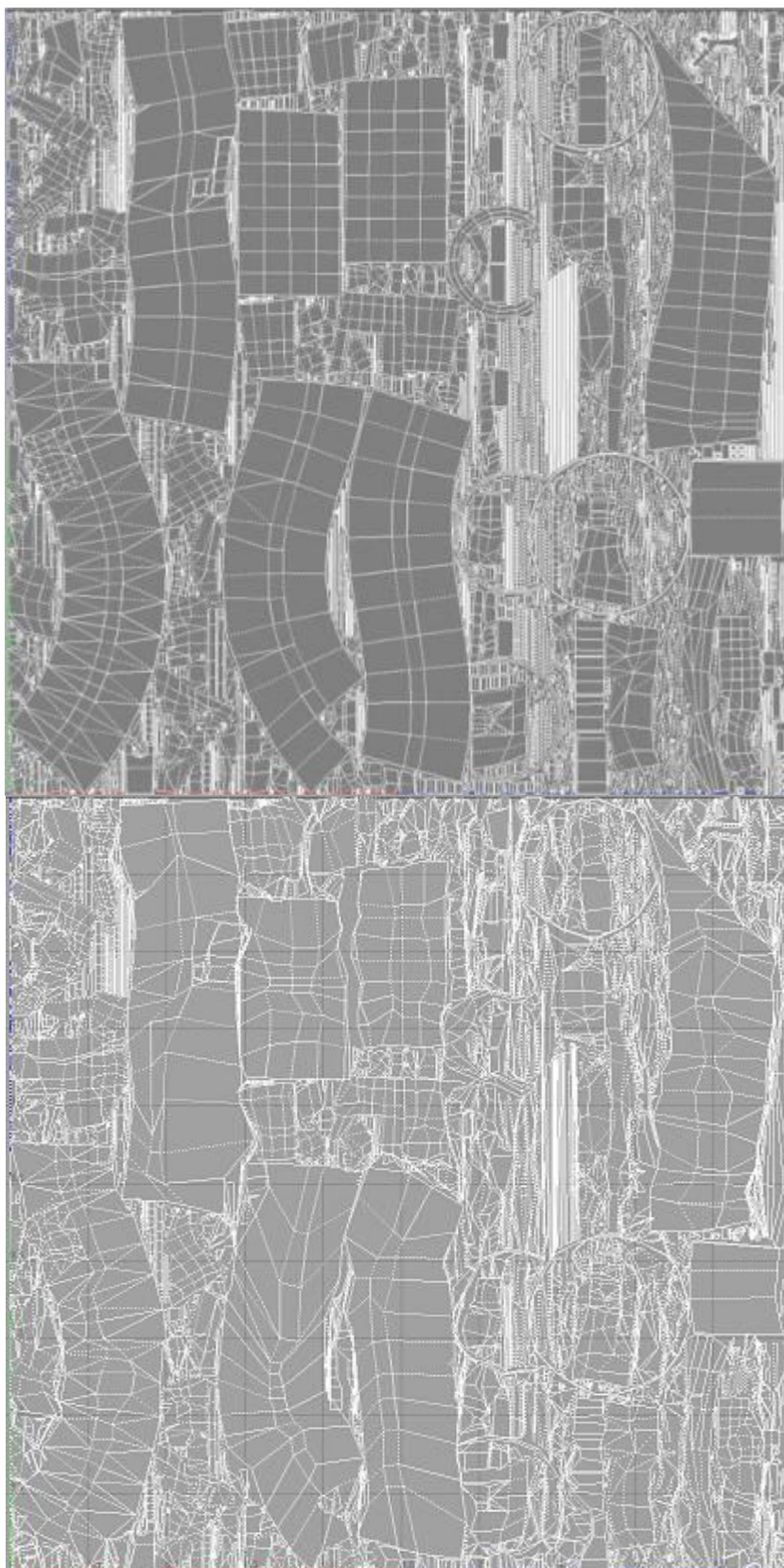


Рис. 27. Сравнение текстурных раскладок до и после модификации

Также в качестве примера приведена часть карты освещенности до и после модификации при помощи предложенного алгоритма (рис. 28).

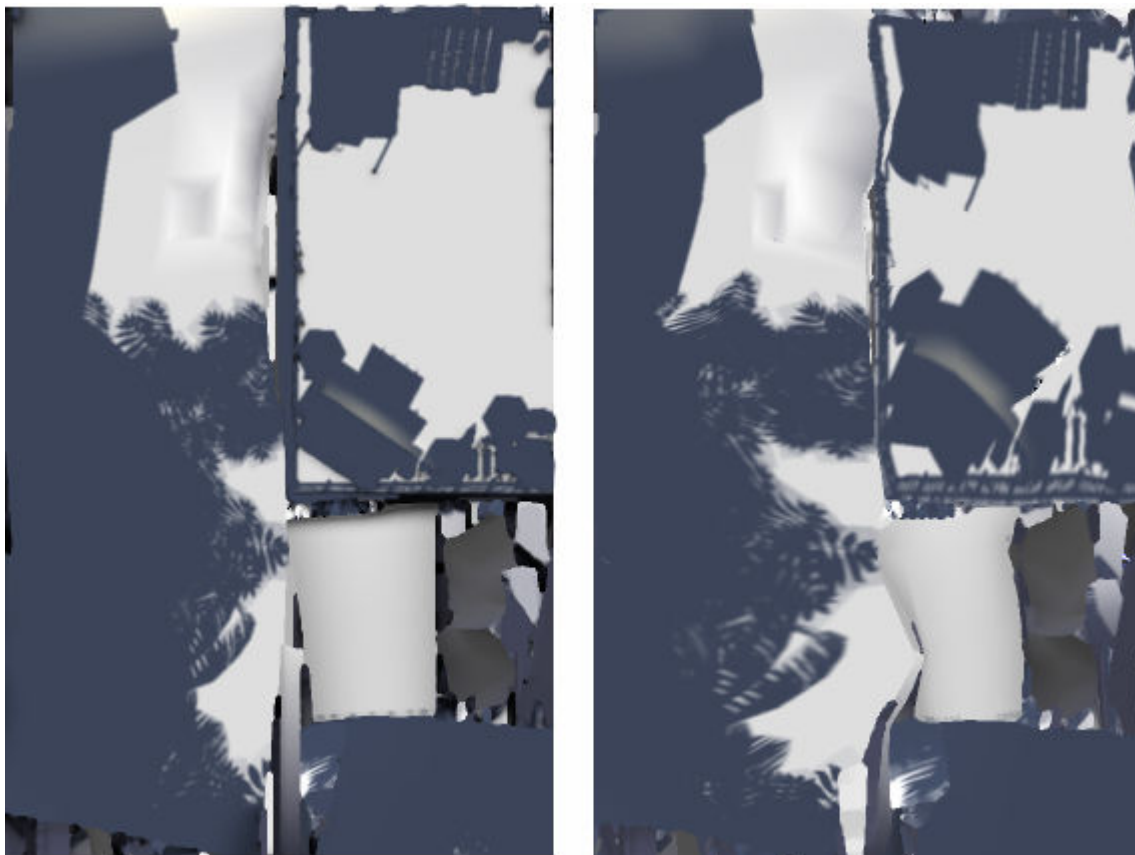


Рис. 28. Сравнение карты освещенности до и после модификации

Для оценки качества карт освещенности, полученных при помощи алгоритма адаптивной модификации текстурной раскладки, по сравнению с *Lightmap*-текстурами, рассчитанными при использовании стандартной реализации, было произведено визуальное сравнение критических мест 3D-сцены с точки зрения освещенности (рис. 29).

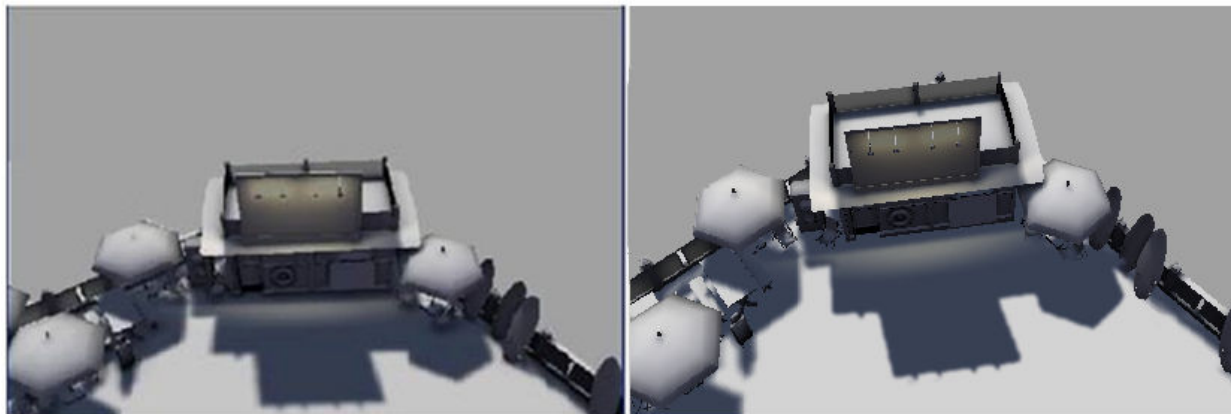


Рис. 29. Сравнение контрольного участка до и после модификации

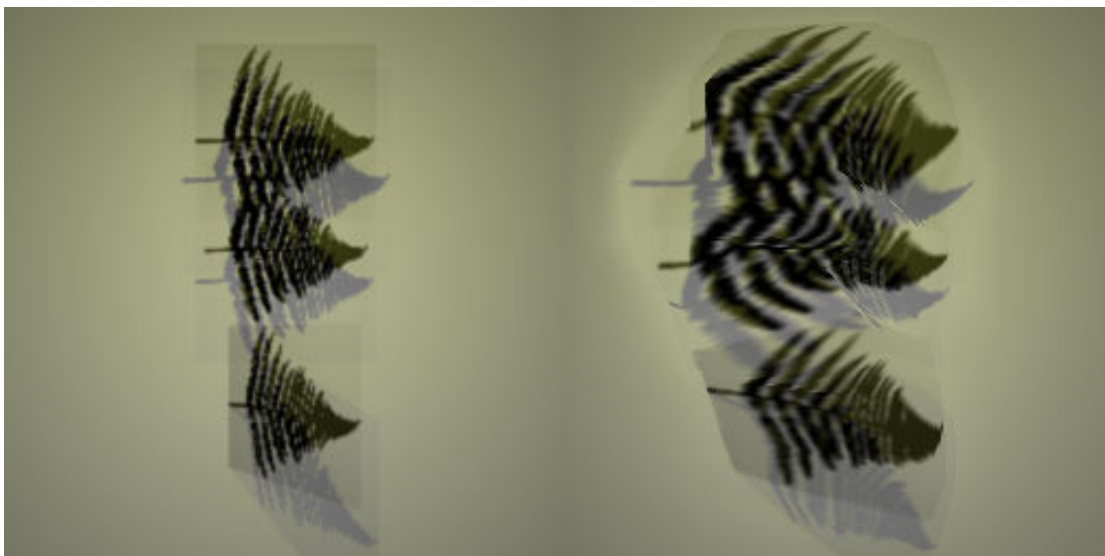


Рис. 30. Сравнение карты освещенности до и после модификации

Также было показано, что на границах переходов свет-тень увеличилось среднее число пикселей на площадь участка перехода (рис. 31).

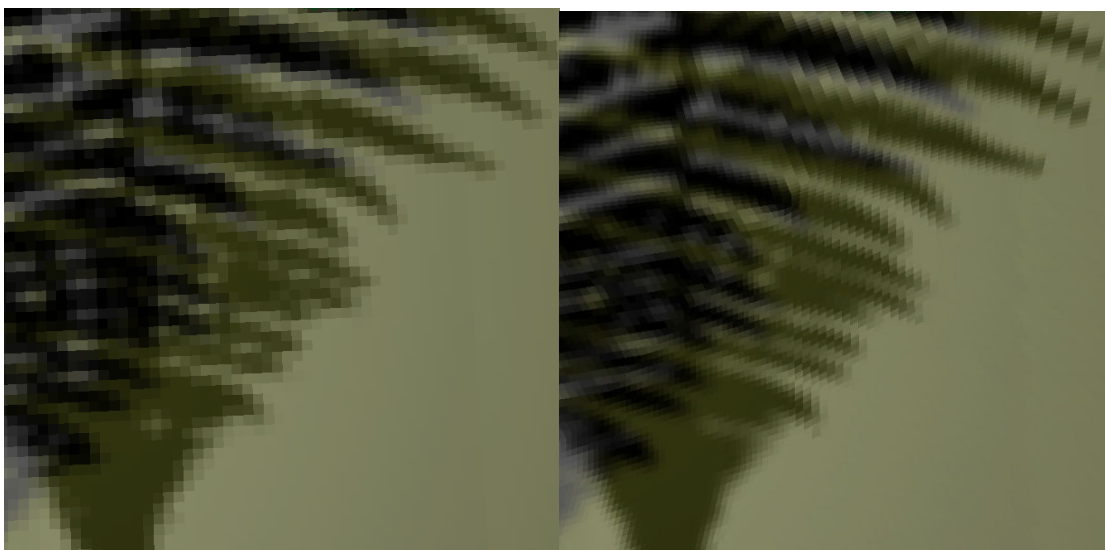


Рис. 31. Изображение части 3D-сцены с использованием карт освещенности до и после модификации (фильтрация текстур отключена)

Было выявлено, что при использовании новых карт освещенности границы свет-тень стали более четкими, особенно хорошо это видно на сложных тенях, таких как тени от листвы, сложных технических сооружений, горных образований и многих других 3D-объектов.

Выводы по главе 3

Выполнен разбор улучшенного алгоритма генерации карт освещенности. Описан принцип действия системы клеточных автоматов, применяемых в работе. Метод опробован на ряде тестовых примеров, что выявило преимущество по сравнению со стандартной реализацией.

Заключение

В работе были получены следующие научные результаты.

Были рассмотрены основные способы построения карт освещенности 3D-сцен и были проанализированы стандартные реализации алгоритма построения карт освещенности. В результате был сделан вывод, что существующий алгоритм генерации *Lightmap* текстур обладает рядом недостатков. Среди них:

1. Использование не всей поверхности текстуры (неиспользованные части занимают видео память при отрисовке).
2. Неоптимальное хранение данных (достаточно большие участки одной освещенности).
3. Не учитывается геометрия сцены и положение и тип источников света.

Был разработан улучшенный алгоритм генерации *Lightmap* текстур. Разработанный подход устраняет перечисленные недостатки, а также привносит некоторые преимущества.

Среди них можно отметить:

1. Простая реализация алгоритма.
2. Использование стандартной реализации, что позволяет реализовать алгоритм в большинстве пакетов 3D-моделирования в виде наборов скриптов и плагинов.
3. Оптимизация текстурной раскладки моделей, участвующих в освещении.
4. Возможность генерации отдельных текстурных раскладок моделей для различных графических эффектов (статических, не зависящих от положения камеры), которые будут близкими к

оптимальным.

Возможны следующие пути развития работы:

1. Оптимизация создания первоначальной текстурной раскладки модели – более оптимальное «разрезания» *3D*-объекта на составные части из которых далее генерируется развертка.
2. Модификация *3D*-моделей для более мелкого дробления зоны перехода свет-тень на полигоны.
3. Модификация текстурной раскладки в три и более этапов.

Список литературы

1. *Mitchell J., McTaggart G., Green C. Shading in Valve's Source Engine. SIGGRAPH Course on Advanced Real-Time Rendering in 3D-Graphics and Games, 2006.*
www.valvesoftware.com/publications/2006/SIGGRAPH06_Course_Shading_InValvesSourceEngine.pdf
2. <http://www.geomerics.com/documents/EnlightenUE3ProductSheet.pdf>
3. *Zhukov S., Iones A., Kronin G. Using light maps to create realistic lighting in real-time applications / Proceedings of WSCG '98. 1998, pp. 464-471.*
4. *Levy B., Petitjean S., Ray N., Maillot J. Least squares conformal maps for automatic texture atlas generation //ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002). 21(3):362–371, July 2002. 3, 4.*
5. *Balmelli L., Taubin G., Bernardini F. Space-optimized texture maps.*
www.balmelli.net/download/balmelli02_100dpi.pdf
6. *Yaohua Hu. Lightmap compression in Halo 3, 2008.*
www.bungie.net/images/Inside/publications/presentations/Lightmap_Compression_2008_02_22.pptx
7. *Kaplan I. The Wavelet Packet Transform. 2002. March.*
http://www.bearcave.com/misl/misl_tech/wavelets/packet/index.html
8. *Kaplan I. Wavelets and Signal Processing. 2001, 2002, 2003.*
http://www.bearcave.com/misl/misl_tech/wavelets/index.html
9. *Топфолу Т., Марголус Н. Машины клеточных автоматов. Мир, 1991.*
10. *Frish U., et al. Lattice gas hydrodynamics in two and three dimensions.// Complex Systems. 1987. Vol. 1, pp. 649–707.*

11. *Wolfram S.* Cellular automation Fluids // J.Stat.Phys. 1986. Vol. 45, pp. 471–526.
12. *Наумов Л. А.* Метод введения обобщенных координат и инструментальное средство для автоматизации проектирования программного обеспечения вычислительных экспериментов с использованием клеточных автоматов. Диссертация на соискание ученой степени кандидата технических наук. СПбГУ ИТМО, 2007.
http://is.ifmo.ru/papers/_ln_Thesis.pdf
13. *Фон Нейман Дж.* Теория самовоспроизводящихся автоматов. М.: Мир, 1971.
14. *Скаков П. С.* Классификация поведения одномерных клеточных автоматов. http://is.ifmo.ru/papers/_skakov_master.pdf
15. *Wolfram S.* A New Kind of Science. Wolfram Media, Inc., 2002.
16. *Gardner M.* The Fantastic Combinations of John Conway's New Solitaire Game «Life» // Scientific American. 1970. № 223, pp. 120–123.
17. *Ulam S.* Random Processes and Transformations // Proceedings Int. Congr. Mathem. 1952. № 2, pp. 264–275.