

УДК 004.925.4

С.И. ВЯТКИН
Институт автоматизации и электротехники СО РАН
С.А. РОМАНИУК, Л. А. САВИЦКАЯ, Т.А. СТРЕЛЬЧУК
Винницкий национальный технический университет

МЕТОД АНИЗОТРОПНОЙ ФИЛЬТРАЦИИ ТЕКСТУРЫ С ИСПОЛЬЗОВАНИЕМ ГРАФИЧЕСКИХ УСКОРИТЕЛЕЙ

Предложен метод анизотропной фильтрации в режиме реального времени на графическом акселераторе. Как более качественная альтернатива трилинейной фильтрации представлен метод приближения эллипса с большим эксцентриситетом несколькими эллипсами с меньшими эксцентриситетами.

Ключевые слова: анизотропная фильтрация, текстурирование, видокарта, графический ускоритель.

С.І. ВЯТКИН
Інститут автоматизації і електротехніки СВ РАН
С.О. РОМАНИУК, Л.А. САВИЦЬКА, Т.О. СТРЕЛЬЧУК
Вінницький національний технічний університет

МЕТОД АНИЗОТРОПНОЇ ФІЛЬТРАЦІЇ ТЕКСТУРИ З ВИКОРИСТАННЯМ ГРАФІЧНИХ ПРИСКОРЮВАЧІВ

Пропонується реалізація методу анизотропної фільтрації у режимі реального часу на графічному акселераторі. Як більш якісна альтернатива трілінійної фільтрації пропонується метод наближення еліпса з великим ексцентриситетом декількома еліпсами з меншими ексцентриситетами.

Ключові слова: анизотропна фільтрація, текстурування, відеокарта, графічний прискорювач.

S.I. VIATKIN
Institute of Automation and Electrometry SB RAS
S.O. ROMANIUK, L.A. SAVYTSKA, T.O. STRELCHUK
Vinnytsia National Technical University

METHOD OF ANISOTROPIC TEXTURE FILTERING USING GRAPHICS ACCELERATORS

The method of anisotropic filtering in real time on the graphic accelerator . As a higher quality alternative to tri-linear filtering approximation method presented an ellipse with an eccentricity of several large ellipses with eccentricities less for a high degree of anisotropy.

Keywords: anisotropic filtering, texturing, video card, graphics accelerator.

Постановка проблеми

Применяемые в трехмерной графике методы наложения текстур, используются для визуализации трехмерных сцен с высокой степенью детализации. Генерация текстуры заключается в проецировании изображения на трехмерную поверхность, таким образом, обеспечивается дополнительная детализация объекта без усложнения его геометрии. При этом появляется множество разнообразных ошибок визуализации, называемых артефактами [1]. Было разработано множество различных методов, которые уменьшают количество подобных артефактов визуализации. Для имитации реалистичных сцен необходимо использовать большое количество детализированных текстур.

Поточечная выборка – самый простой метод определения цвета пикселя на основе текстурного изображения. Выбирается текстель, ближе всех расположенный к центру светового пятна. Вследствие того, что цвет пикселя определяют несколько текстелей, а выбирается только один, происходит ошибка. Кроме того, форма светового пятна может измениться в зависимости от наклона грани по отношению к наблюдателю [2].

Главное преимущество данного метода фильтрации – это низкие требования к ширине полосы пропускания памяти, т.к. для определения цвета пикселя, нужно выбрать всего лишь один текстель из текстурной памяти. Главный недостаток – ухудшение качества изображений при приближении полигона к точке наблюдения, когда количество пикселей становится больше количества текстелей. Кроме того, необходимо устранять артефакты неправильного расчета глубины сцены.

Билинейная фильтрация использует интерполяционную технику. Для определения текстелей, которые должны быть задействованы при интерполяции, используется форма светового пятна в виде круга.

Круг аппроксимируется четырьмя текселями. Этот метод фильтрации эффективнее поточечной выборки, поскольку частично учитывает форму светового пятна и использует интерполяцию. Так, если полигон приближается слишком близко к точке наблюдения, то для интерполяции потребуется больше текселей, чем в действительности доступно.

Главный недостаток билинейной фильтрации заключается в том, что аппроксимация выполняется корректно только для полигонов, которые расположены параллельно экрану или точке наблюдения. Если полигон развернут под углом, то получается неправильная аппроксимация, так как необходимо аппроксимировать эллипс, а не круг. При билинейной фильтрации требуется считывать по четыре текселя из текстурной памяти для определения цвета каждого выводимого на экран пикселя, а это значит, что требования к ширине полосы пропускания памяти увеличиваются в четыре раза, по сравнению с поточечной выборкой. Существует класс артефактов визуализации, появляющихся в результате наложения текстур, так называемый "depth aliasing" (другое название Z-aliasing), которые билинейная фильтрация не может устранить. Ошибки определения глубины сцены возникают от того, что при удалении от точки наблюдения, текстурное изображение, наложенное на уменьшившийся в размерах объект, становится все более и более сжатым. Эти ошибки визуализации особенно проявляются при анимации, когда время движения видны мерцания и эффект медленного движения в той части изображения, которая должна быть неподвижной.

Для устранения артефактов "depth aliasing" и использовании того факта, что объекты на расстоянии выглядят менее детализированными, чем те, что находятся ближе к точке наблюдения, применяется техника, известная как mip-mapping [3]. Mip-mapping – это метод наложения текстур, имеющих разную степень или уровень детализации, в котором выбирается текстура с необходимой детализацией в зависимости от расстояния до точки наблюдения. Mip-Map текстура состоит из набора заранее отфильтрованных и масштабированных изображений. Каждый пиксель представляется в виде среднего от четырех пикселей из предыдущего уровня с более высоким разрешением. Поэтому изображение, связанное с каждым уровнем текстуры в четыре раза меньше по размеру предыдущего mip-map уровня.

Уровень детализации – Level of Detail (LOD) – используется для определения, какой mip-map уровень (или какую степень детализации) следует выбрать для наложения текстуры на объект. LOD должен соответствовать числу текселей, накладываемых на пиксель, то есть, какое количество элементов текстуры должно быть наложено на элемент выводимого на экран изображения. Например, если текстурирование происходит с соотношением близким к 1:1, то LOD будет 0, а значит и будет использоваться mip-map уровень с самым высоким разрешением. Если четыре текселя накладываются на один пиксель, то LOD будет 1 и будет использоваться следующий mip-map уровень с меньшим разрешением.

Если для каждого пикселя в треугольнике вычисляется значение LOD для определения того, какая mip-map текстура имеет нужный уровень фильтрации, то в результате предотвращается появление ошибок визуализации и излишней размытости. LOD должен вычисляться для каждого пикселя в треугольнике с целью избежания разрывов по краям треугольников. Этот метод называется per-pixel mip-mapping (попиксельное mip-текстурирование). Если LOD вычисляется всего лишь раз для всего треугольника, то следствием использования этого значения для всех пикселей треугольника становится появление эффекта размытости, когда некоторые треугольники, из которых состоит анимированный объект, становятся размытыми и содержат неровности. Этот менее эффективный метод mip-map текстурирования известен как "per-polygon mip-mapping" (т.е. mip-текстурирование для полигона).

При удалении объекта от точки наблюдения, происходит переход от низкого mip-map уровня, соответствующего изображению с высокой детализацией, к высокому mip-map уровню, соответствующего изображению с высокой степенью фильтрации и низкой детализацией. В момент нахождения объекта в переходном состоянии от одного mip-map уровня к другому, появляется особый тип ошибок визуализации, известных под названием "mip-banding" - полосатость или слоеность, когда появляются различные границы перехода от одного mip-map уровня к другому. "Mip-banding" происходит, когда значение LOD округляется до целого и только один соответствующий уровень текстуры используется для генерации пикселя. После перехода LOD к более высокому уровню, соседние пиксели будут обрабатываться с различными уровнями, и будут иметь различное количество примененных к ним фильтров. Результатом будет появление линии или нескольких линий, проходящих через треугольник в местах, где происходит переход между уровнями. Это один из недостатков использования только билинейного или поточечного mip-текстурирования. Особенно такие артефакты заметны при анимации, потому-что человеческий глаз очень чувствителен к смещениям и может легко заметить место резкого перехода между уровнями фильтрации при движении объекта.

Трилинейная фильтрация включает mip-текстурирование и билинейную интерполяцию. Билинейная интерполяция осуществляется на двух mip уровнях, в результате вычисляются два текселя, по одному для каждого mip уровня. Цвет пикселя определяется в результате интерполяции по цветам двух mip-текстур. Таким образом, mip уровни представляют собой заранее рассчитанные более детализированные версии

исходной текстуры, поэтому получается более хорошая аппроксимация текселей, расположенных в световом пятне.

Эта техника обеспечивает лучшую фильтрацию. Однако требования к ширине полосы пропускания памяти удваиваются, по сравнению с билинейной фильтрацией, так как необходимо считывать восемь текселей из текстурной памяти. Использование метода mip-map текстурирования обеспечивает лучшую аппроксимацию (используется большее число текселей, расположенных в световом пятне) по всем текселям в световом пятне, благодаря использованию заранее рассчитанных mip-текстур.

При использовании трилинейной фильтрации возможен вывод на экран текстурированного объекта с плавно выполненными переходами от одного mip уровня к следующему, что достигается за счет определения LOD путем интерполяции двух соседних mip-map уровней. Таким образом, решается большинство проблем, связанных с mip-текстурированием и ошибками из-за неправильного расчета глубины сцены ("depth aliasing").

Чтобы добиться лучшего качества, необходимо использовать все тексели, покрываемые световым пятном и усреднять их значение. Однако это сильно влияет на пропускную способность памяти – ее может не хватить, и выполнить такую выборку с усреднением нетривиальная задача. Можно использовать разнообразные фильтры для аппроксимации формы светового пятна в виде эллипса для нескольких возможных углов положения полигона относительно точки зрения. Существуют техники фильтрации, которые используют от 16 до 32 текселей из текстуры для определения цвета пикселя. Правда использование подобной техники фильтрации требует существенно большей ширины полосы пропускания памяти, а это почти всегда невозможно в существующих системах визуализации без применения дорогостоящих архитектур памяти. В системах визуализации, использующих тайлы [4, 5], существенно экономятся ресурсы полосы пропускания памяти, что позволяет использовать анизотропную фильтрацию. Визуализация с применением анизотропной фильтрации обеспечивает лучшее качество изображения, за счет лучшей глубины детализации и более точного представления текстур, наложенных на полигоны, которые расположены не параллельно экрану или точке наблюдения [6].

В данной работе предлагается метод анизотропной фильтрации в режиме реального времени на графическом акселераторе.

Описание метода

Основная идея метода – это приближение эллипса с большим эксцентриситетом несколькими эллипсами с меньшими эксцентриситетами. Таким образом, можно преодолеть ограничения, наложенные аппаратными средствами и обеспечить фильтрацию лучшим фильтром с высокой степенью анизотропии.

Эксцентриситет — числовая характеристика конического сечения, показывающая степень его отклонения от окружности. Обычно обозначается "e".

Все невырожденные конические сечения, кроме окружности, можно описать следующим способом:

выберем на плоскости точку F и прямую d и зададим вещественное число $e > 0$. Тогда геометрическое место точек M , для которых отношение расстояний до точки F и до прямой d равно e раз, является коническим сечением. То есть, если M' есть проекция M на d , то

$$FM = e \|MM'\|$$

Эксцентриситет инвариантен относительно движений плоскости и преобразований подобия.

Поместим эллипсы с меньшими эксцентриситетами в линию вдоль главной оси эллипса. Длина линии L и число эллипсов N_e вычисляется следующим образом

$$N_e = 2 \left(\frac{R_{maj}}{\alpha R_{min}} \right) - 1$$

$$L = 2(R_{maj} - \alpha R_{min})$$

где α - степень анизотропии. Для простоты рассматриваем нечетное число эллипсов. Эллипсы располагаются вокруг точки (u_m, v_m) фильтра, следующим образом

$$\vartheta = a \tan\left(\frac{B}{A-C}\right) / 2$$

$$du = \cos(\vartheta) \frac{L}{N_e - 1}$$

$$dv = \sin(\vartheta) \frac{L}{N_e - 1}$$

$$(u_n, v_n) = (u_m, v_m) + \frac{n}{2}(du, dv), \quad n = 0, \pm 2, \pm 4, \dots$$

где (u_m, v_m) – положение n -ого эллипса. Чтобы лучше соответствовать форме фильтра EWA, эллипсы взвешены пропорционально их расстоянию от центра следа, согласно функции Гаусса.

Несколько эллипсов приближают анизотропную область с максимальной анизотропией N . Чтобы лучше покрыть область эллиптического фильтра, увеличиваем число эллипсов в зависимости от желаемого качества. Использование адаптивного числа эллипсов создает нерегулярную рабочую нагрузку за пиксель для графического планировщика аппаратных средств, и этого лучше избегать. Определение количества эллипсов с постоянным числом обеспечивает лучшую производительность. В тестах использовались от трех до семи эллипсов на аппаратных средствах NVIDIA. Больше чем пять эллипсов не понадобилось, так как в ходе исследований не было зафиксировано никакого существенного улучшения в качестве изображения.

Если эллипсы превысят степени эллипса с большим эксцентриситетом, например, размещая их вне половины длины центральной линии в каждом направлении, то тогда произойдет смаз (размывание). Если эллипсы не покроют эллипс с большим эксцентриситетом, то произойдет алиазинг. Для крайних случаев, максимальное количество эллипсов ограничено. Метод всегда обеспечивает лучшее качество по сравнению с фильтрацией текстуры, реализованной в аппаратных средствах.

При тестировании использовался гауссовский фильтр с отклонением 0.5 и шириной в один пиксель. Настройка отклонения фильтра влияет на точность или гладкость изображения. Меньшее отклонение создает более резкое изображение, но может привести к алиазингу, когда используется текстура с высокочастотными компонентами. В результате текстурования установлено, что отклонение 0.5 дает хороший компромисс между предотвращением алиазинга и смазом изображения.

Тестирование метода проводилось на Intel Core 2 Duo CPU E8400 3.0 GHz и 470 GTX с разрешением 1024x768. Измерялись как перцепционная метрика, так и среднеквадратическая ошибка.

Выводы

Трилинейная фильтрация была раньше недоступна, большинство приложений использовали билинейную фильтрацию для обеспечения минимального качества изображения. Затем поддержка трилинейной фильтрации стала стандартной функцией графических акселераторов, так как она обеспечивает вывод трехмерного изображения с высоким качеством при наличии некоторого количества артефактов визуализации. Метод анизотропной фильтрации текстуры, представленный в данной работе, обеспечивает еще лучшее качество изображения.

Список использованной литературы

1. Вяткин С.И. Антиалиазинг текстур /О.В. Романюк, А.В. Мельник // V International Conference on Optoelectronic Information Technologies «PHOTONICS-ODS 2010» (Ukraine, Vinnytsia, VNTU, September 28–30, 2010). – С. 82–83.
2. Вяткин С.И. Метод вычисления текстурных координат для отображения текстуры на плоские полигоны / О.Н. Романюк, С.О. Кришук //Международный научно-технический журнал «Измерительная и вычислительная техника в технологических процессах», 2012, Хмельницкий национальный университет, Хмельницкий, Украина, 2012, № 2.– С. 86–89.
3. Вяткин С.И. Вычисление текстуры с предфильтрацией для рендеринга полигонов в реальном времени / О.В. Романюк // Материалы IV Международной научно-технической конференции "Моделирование и компьютерная графика" (г. Донецк, Украина, Дон НТУ, 5--8 октября 2011).– С. 47-50.
4. Вяткин С.И. Особенности текстурирования в тайловой технологии визуализации / А.Н. Романюк, Р.Ю. Довгалоук // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника», 2012, Донецк: ДонНТУ, вып. 15 (203). – С. 79–82.
5. Вяткин С.И. Формирование реалистических изображений с использованием пошагового вычисления текселей / О.В. Романюк, А.А. Стахов // Сборник научных трудов по материалам международной научно-практической конференции «Современные направления теоретических и прикладных исследований», 2011, т. 3, Технические науки, Одесса. Изд.: Черноморье.– С. 65–69.
6. McCormack, J., Perry, R., Farkas, K. I., and Jouppe, N. P. Feline: fast elliptical lines for anisotropic texture mapping. In SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., pp. 243–250. 1999.