

УДК 004.932

Отображение контурной текстуры на произвольно ориентированные плоские и криволинейные поверхности

С. И. Вяткин¹, О. В. Романюк², Н. С. Костюкова³

¹Институт автоматизации и электрометрии СО РАН, Россия

²Винницкий национальный технический университет, Украина

³Донецкий национальный технический университет

oksana_gelika@mail.ru

Abstract

Vyatkin S.I., Romaniuk O. V., Kostyukova N.S. Representation of Contour Texture onto Arbitrary-Oriented Planar and Curved Surfaces. There had been considered the issues of contour texture application for creation and representation of free-formed contours within texture map which are not blurred when approaching. There had been proposed the method of representation of contour texture onto arbitrary-oriented planar and curved surfaces.

Введение

Отображение текстуры [1, 2] на поверхности является эффективным методом повышения реализма в системах компьютерной графики. Применение текстур позволяет, в первую очередь, моделировать цветовой рисунок на поверхностях, а также прозрачность, резкие границы, движущиеся объекты и многие другие спецэффекты [3]. Текстура значительно повышает визуальную сложность изображения при относительно малом количестве вычислений. Термин "текстура" определяется в компьютерной графике как двумерное изображение, отображаемое на трехмерную плоскость или трехмерную криволинейную поверхность, или трехмерное изображение отображаемое на трехмерное пространство.

Анализ методов и постановка задачи

При отображении текстуры приходится решать две проблемы: геометрическое отображение массива текстуры на поверхность и фильтрация изображения, которая устраняет на изображении ступенчатые линии, муары и т.д. – все, что в литературе по компьютерной графике обозначается термином элайзинг (англ. Aliasing).

Процесс отображения текстурных карт на плоские грани включает два этапа.

Первый – перспективное преобразование, т.е. вычисление координат текстурной карты (u, v) , соответствующих координатам (X_s, Y_s) пиксела на экране. Плоскость параметризуется, если задать на ней единичные текстурные вектора U и V и точку привязки. Задача состоит в том, чтобы, зная координаты экрана (X_s, Y_s) , получить текстурные координаты U и V соответствующей точки плоскости в системе координат наблюдателя. Это преобразование может быть записано в матричном виде [1]:

$$[U, V, W, W] = [X_s, Y_s, I] \begin{bmatrix} A_u & A_v & A_z \\ B_u & B_v & B_z \\ C_u & C_v & C_z \end{bmatrix} \quad (1)$$

Текстурные координаты связаны с экранными координатами дробно-линейной функцией:

$$u = \frac{A_u X_s + B_u Y_s + C_u}{A_z X_s + B_z Y_s + C_z} \quad (2a)$$

$$v = \frac{A_v X_s + B_v Y_s + C_v}{A_z X_s + B_z Y_s + C_z}, \quad (2b)$$

где (X_s, Y_s) – экранные координаты пиксела, A_i, B_i, C_i – элементы матрицы преобразования координат.

Второй этап отображения текстуры – фильтрация, необходимая для предотвращения элайзинга. Наиболее часто применяются так называемые MIP-карты (пирамидальные) текстурные карты [4] (рис. 1).



Рисунок 1 – Пирамидальные текстурные карты

Путем предварительной фильтрации получают набор квадратных текстурных карт с разным разрешением для каждого объекта. Каждой текстурной карте ставят в соответствие целочисленное значение так называемого уровня детальности (LOD-level of detail, англ.).

В зависимости от расстояния до грани и ее ориентации выбираются для работы две текстурные карты с соседними уровнями детальности. Критерием выбора является линейный размер проекции пиксела на грань. Если проекция пиксела покрывает менее двух текстурных элементов (texel-элемент текстурной карты), наблюдается элайзинг. У грани, с углами наклона (между плоскостями грани и экрана), близкими к нулю, текстурный рисунок сильно размывается. Этот эффект носит название *смаза* [1] (рис. 2 и 3.).

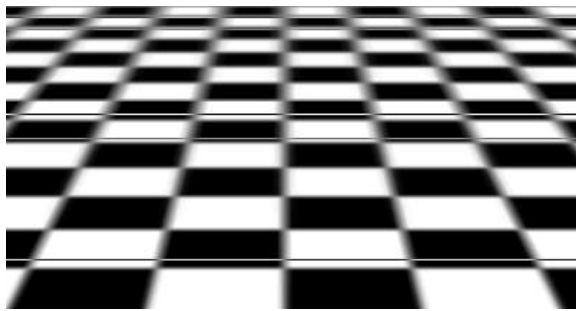


Рисунок 2 – Появление артефактов при неаккуратном вычислении текстурных координат

Затем в соответствии с текстурными координатами из каждой текстурной карты считываются по четыре текстела. Трилинейная интерполяция по этим восьми значениям завершает процесс фильтрации. Коэффициентами трилинейной интерполяции являются дробные части текстурных координат и уровня детальности. Путем видоизменения процесса фильтрации получают некоторые специальные эффекты. Например, можно получить текстурный рисунок с постоянной шириной контура.

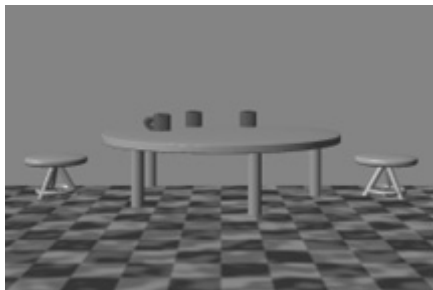


Рисунок 3 – Сцена с функционально заданными объектами и текстурой без артефактов

В данной работе предлагается метод отображения контурной текстуры на произвольно ориентированные плоские и криволинейные

поверхности. Область применения такой текстуры - отображение букв, символов, границ раздела и т.д.

Контурная текстура

Контурная текстура должна обеспечивать возможность создания и отображения контуров произвольной формы в рамках текстурной карты, не размывающихся при приближении. Перечислим основные свойства контурных текстурных карт:

- ширина контура составляет 1-2 пиксела;
- при приближении (действительное значение LOD отрицательно);
- контур не размывается, ширина контура сохраняется;
- в области с неотрицательными значениями LOD контурные текстуры переходят в обычные, RGBT – текстуры;
- форма контура достаточно произвольна, т.е. для задания контуров требуется более одного бита на текстел.

Рассмотрим выбор уровня детализации текстурной карты. В общем случае проекция "круглого пиксела" на плоскость есть эллипс [1, 5], большую ось которого возьмем в качестве диаметра \varnothing фильтра. Размер эллипса зависит от:

- интервала выборки $\delta = \max(\delta x, \delta y)$;
- расстояния от наблюдателя до точки пересечения луча зрения с плоскостью P (D);
- угла наклона плоскости к лучу зрения.

Большая ось эллипса лежит на линии пересечения двух плоскостей. Одна из них – текстурированная плоскость, а другая – такая, что в ней лежит нормаль к плоскости и луч зрения. Из этого следует, что $\varnothing = 2\delta D / \cos \theta$, где θ - угол между нормалью к плоскости и лучом зрения, $\cos \theta = d / D$, где d – расстояние от начала системы координат до плоскости. D - расстояние от наблюдателя до точки пересечения луча зрения с поверхностью объекта. Поэтому

$$\varnothing = 2\delta D^2 / d \approx 2\delta Z^2 / d \quad (3).$$

Для получения окончательного результата, т. е. цвета и прозрачности в каждом пикселе требуется произвести трилинейную интерполяцию [4]:

$$\begin{aligned} result &= F_level + \gamma(C_level - F_level), \\ F_level &= F_bottom + f\beta(F_top - F_bottom), \\ F_bottom &= f(fu, fv) + f\alpha(f(fu+1, fv) - f(fu, fv)), \\ F_top &= f(fu, fv+1) + \\ &+ f\alpha(f(fu+1, fv+1) - f(fu, fv+1)), \\ C_level &= C_bottom + c\beta(C_top - C_bottom), \\ C_bottom &= c(cu, cv) + c\alpha(c(cu+1, cv) - c(cu, cv)), \\ C_top &= c(cu, cv+1) + c\alpha(c(cu+1, cv+1) - c(cu, cv+1)), \end{aligned}$$

где $f\alpha$, $f\beta$ - дробные части текстурных координат

высокого уровня детальности; ca, cb - дробные части текстурных координат низкого уровня детальности; $c[ci, cv], f[fi, fv]$ - значения цветовых компонент в точке $[u, v]$ для каждого из уровней детальности.

Исходные данные - восьмиразрядные значения компонент цвета (прозрачности). Коэффициенты билинейной интерполяции α и β для каждого из выбранных уровней детальности получаются из U и V адресов. Коэффициент линейной интерполяции между разными уровнями детальности g выбирается в зависимости от наибольшего диаметра проекции пиксела (эллипса) на текстурную плоскость.

Для обработки контурных текстурных карт необходим канал прозрачности. Это позволяет получать полноцветные изображения внутри контуров (рис. 4 и 5). Однако для получения полноценной картины потребуется двойной слой текстурированных граней: один - для изображения внутри контура, второй - вне его.

После билинейной интерполяции, такой же, как и в остальных каналах, для выделения контура производятся следующие вычисления:

$$T = \frac{t - 0.5}{Ps} + 0.5, \quad (4)$$

далее следует коррекция:

$$T = \begin{cases} 0, & \text{если } T < 0 \\ T, & \text{если } 0 < T < 1 \\ 1, & \text{если } T > 1 \end{cases} \quad (5)$$

где T - окончательный результат, t - результат билинейной интерполяции, Ps - размер проекции пиксела.

Дополнительно производится операция:

$$T = (t - 0.5) \ll \text{Log}_2 Ps + \gamma'(t - 0.5) \ll \ll \text{Log}_2 Ps + 0.5, \quad (6)$$

где γ' - дробная часть размера проекции пиксела.

Для вычисления текстурного адреса необходимы текстурные координаты (2а, 2б), уровень детальности пиксела, величина общего порядка текстурных координат и смещение локальных и глобальных текстурных координат друг относительно друга. Далее каждая из текстурных координат разделяется на четыре числа.

Коэффициенты билинейной интерполяции α и β используются для трilinearной интерполяции текстуры.

$$fu = ((fiu - du) \& 0111) + (du \& 011),$$

$$cu = ((ciu - du/2) \& 011) + ((du/2) \& 011),$$

где (fiu, fiv) и (ciu, civ) - текстурные координаты текущего пиксела, масштабированные в соответствии с уровнем детальности; du (dv) -

смещение сетки локальных текстурных координат относительно глобальных, имеют смысл только для сжатых текстурных карт (равны нулю для несжатой текстуры). Для fv, cv вычисления аналогичны. Тектурный адрес формируется из величин fu, fv, cu, cv .



Рисунок 4 – Обычная текстура

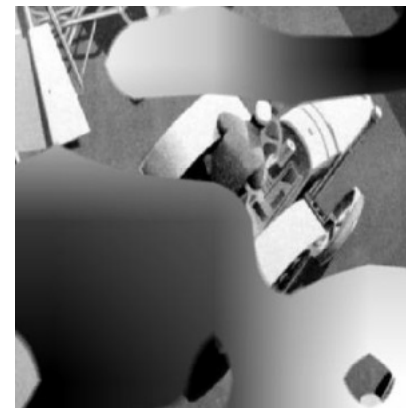


Рисунок 5 – Контурная текстура (сверху), обычная текстура (снизу); приведенные примеры отличаются только интерперетацией данных в канале прозрачности

При отображении текстуры на криволинейную функционально заданную поверхность необходимо преобразование координат из текстурного пространства (U, V, W) в пространство объекта (X, Y, Z) . Реализованы

три вида преобразования объектных координат в текстурные координаты [6]:

1. *Линейное:*

$$\begin{aligned} U &= (1 + X) / 2, \\ V &= (1 + Y) / 2, \\ W &= (1 + Z) / 2. \end{aligned} \quad (7)$$

2. *Цилиндрическое:*

$$\begin{aligned} U &= \text{Arc tan}(Y / X) / 2\pi, \\ V &= 1 + Y) / 2, \\ W &= \sqrt[3]{(X^2 + Y^2)}. \end{aligned} \quad (8)$$

3. *Сферическое:*

$$\begin{aligned} U &= \text{Arc tan}(Y / X) / 2\pi, \\ V &= \text{Arc tan}(\sqrt[3]{(X^2 + Y^2)} / Z) / 2\pi, \\ W &= \sqrt[3]{(X^2 + Y^2 + Z^2)}. \end{aligned} \quad (9)$$

На рисунке 6 показан результат отображения контурной текстуры на криволинейные поверхности функционально заданного объекта.

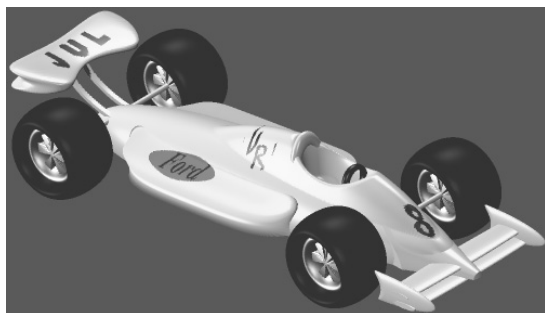


Рисунок 6 – Контурная текстура на криволинейных поверхностях функционально заданного объекта

Заключение

Основное направления развития компьютерной графики – повышение качества изображения, производительности графических систем и рациональное использование памяти. Последнее связано с тем, что изображения фото-графического качества с разрешением 1024x1024, используемые для текстурирования объектов, требуют около 5 мегабайт оперативной памяти каждая.

Общая производительность систем определяется быстродействием геометрического процессора (геометрические преобразования) и пиксельного процессора (растровые операции). Анализ существующих систем показывает, что для получения фотореалистичных изображений в реальном времени следует поднять производительность растровых операций.

Решить проблему нехватки памяти может применение сжатых текстурных карт. Кроме того, это снижает интенсивность обмена с текстурной памятью и дает возможность использования видеотекстур, т. е. получаемых непосредственно с видеокамеры.

Эти и другие задачи трехмерной графики требуют от оборудования производительности, недостижимой для универсальных процессоров. Поэтому современные зарубежные системы трехмерной графики не обходятся без графических ускорителей. В данной работе предложен метод отображения контурной текстуры на произвольно ориентированные плоские и криволинейные поверхности, с помощью которого повышается качество изображения при экономном использовании текстурной памяти.

Литература

1. Heckbert Paul S. Survey of Texture Mapping/ Paul S. Heckbert // IEEE Comput. Graph. and Applicat. – 1986. – Vol.6, N11. – P. 56-67.
2. Catmull E. A Subdivision Algorithm for Computer Display of Curved Surfaces / E. Catmull // PhD Thesis, University of Utah, 1974.
3. Haeberli Paul. and Mark Segal. Texture Mapping as a Fundamental Drawing Primitive /Paul Haeberli, Mark Segal // *Fourth Eurographics Workshop on Rendering*. – June, 1993. – P. 259-266.
4. Williams L. Pyramidal Parametrics /L. Williams//, *Computer Graphics (SIGGRAPH '83 Proceedings)*. – July, 1983. – P. 1-11.
5. Ковалев А.М. Текстура на произвольно ориентированных плоских поверхностях / А.М. Ковалев, Ю.В. Тарасов // *Автометрия*. – 1986, №6. – С. 24.
6. Вяткин С.И., Отображение текстуры на плоские и криволинейные поверхности, свободные формы и объемы / С.И. Вяткин, Б.С. Долговесов, Н.Р. Каипов // *Автометрия*. – 2002, – №1. – С.17-24.

Поступила в редакцию 15.03.2010