

УДК 004.925

UDC 004.925

МЕТОД АНТИАЛИАСИНГА ДЛЯ ПРИЛОЖЕНИЙ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ**ANTIALIASING METHOD FOR VIRTUAL REALITY APPLICATIONS***М. А. Левченко*

Донской государственный технический университет, Ростов-на-Дону, Российская Федерация

mishkun.lev@gmail.com

Разработка программных продуктов, реализующих концепцию виртуальной реальности, зачастую связана с рядом высоких требований. В компьютерной графике существует множество методов для борьбы с эффектом алиасинга, который сильно влияет на качество итогового изображения. В данной работе предложен новый метод антиалиасинга, использующий особенности конвейера визуализации приложений виртуальной реальности.

Ключевые слова: антиалиасинг, сглаживание, виртуальная реальность, компьютерная графика.

М. А. Levchenko

Don State Technical University, Rostov-on-Don, Russian Federation

mishkun.lev@gmail.com

Virtual reality has set new standards of visualization realism for developers. Computer graphics has many methods to deal with image aliasing, the quality of which strongly affects the resulting image. In this paper, we propose a new method of anti-aliasing, using the features of rendering pipeline of virtual reality applications.

Keywords: antialiasing, virtual reality, computer graphics.

Введение. Основное преимущество приложений виртуальной реальности, а именно высокое погружение пользователя в компьютерную симуляцию, напрямую зависит от качества визуального сигнала, получаемого пользователем. Одной из угроз качества изображения является эффект алиасинга или «зубчатости». Алиасинг проявляется из-за дискретной природы шейдинга изображений [1]. При рендеринге высокочастотная часть информации, отвечающая за межпиксельные детали изображения, теряется при дискретизации обрабатываемого фрагмента, что продемонстрировано на рис. 1. Алиасинг приводит к таким визуальным артефактам как «зубчатость», потеря детализации, «узорчатость» текстур. На рис. 1 красная линия представляет собой реальную фигуру, закрасненные квадраты — ее дискретное представление.

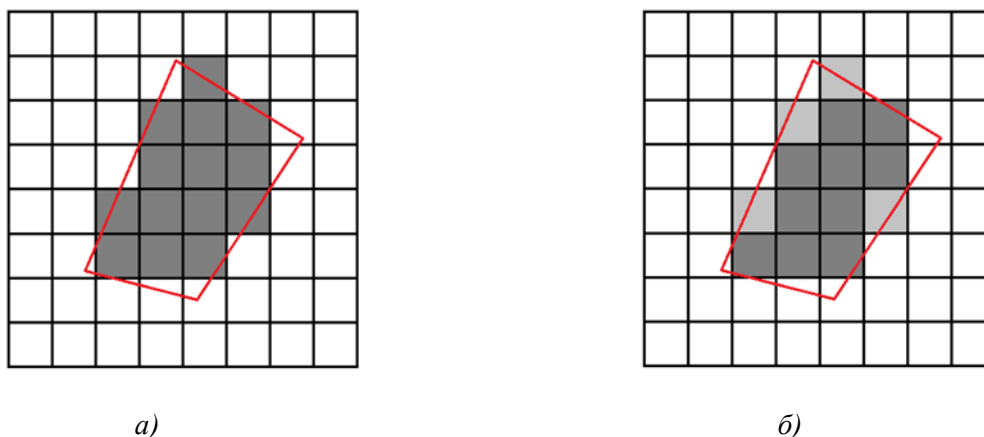


Рис. 1. Дискретизация без антиалиасинга (а) и с антиалиасингом (б)

В современных устройствах виртуальной реальности угловая плотность пикселей существенно ниже, чем поток видеoinформации. Это происходит за счет повышенной частоты кадров и приводит к еще большей заметности артефактов дискретизации изображения для конечного пользователя. Для борьбы с эффектами алиасинга применяются различные алгоритмы антиалиасинга, однако не все из них дают достаточный эффект или способны работать с *Deferred Shading* [2]. В настоящей работе предлагается новый вспомогательный метод антиалиасинга, эксплуатирующий особенности рендеринга изображений для устройств виртуальной реальности, в частности, двукратную перерисовку кадра для каждого глаза.

Описание метода. Автором предложен метод, состоящий в смешивании изображений для левого и правого глаза с учетом реального расположения пикселей. Таким образом достигается двойная выборка межпиксельных деталей и качество антиалиасинга, сопоставимое с *SSAAx2*. Часть накладных расходов, связанная с семплером глубины, уже заложена в стандартный конвейер визуализации. Предлагаемый алгоритм разделяется на четыре шага:

1. семплинг карты глубины;
2. формирование маски смешивания;
3. расчет карты мировой позиции пикселей;
4. смешивание изображений.

Маска смешивания. Исследования показали, что расстояние, равное 20 метрам, является верхней границей восприятия глубины стереоизображений [3], что очевидно из графика, приведенного на рис. 2.

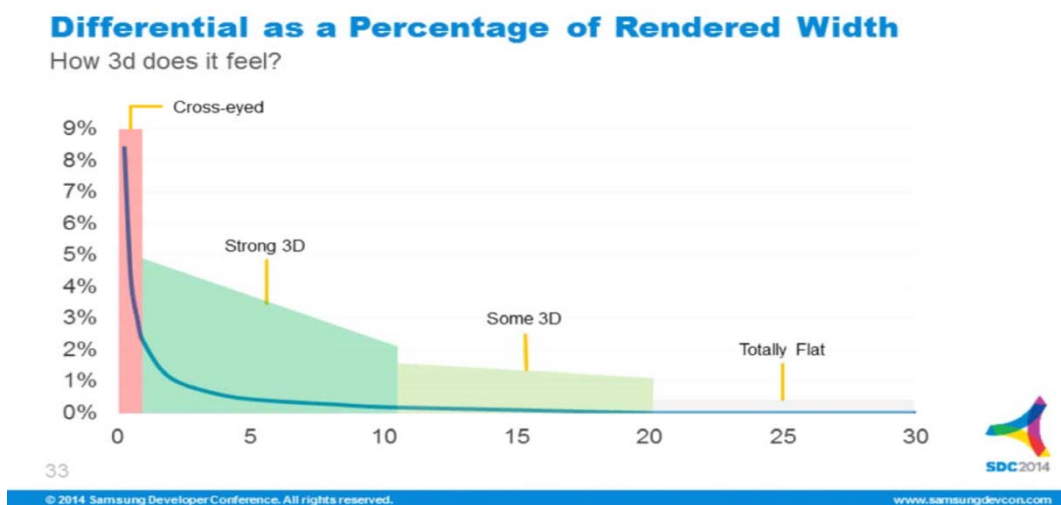


Рис. 2. Зависимость искажения стереоизображения от расстояния до объекта с приведенной субъективной оценкой эффекта глубины

Возможно использовать эту информацию, чтобы улучшить алгоритм, опустив проверку на соответствие мировой позиции для пикселей, значение глубины которых выходит за пределы восприятия человека. Для удобства последующей работы применим следующую пороговую функцию для карты глубины:

$$\text{clamp}(1, 0, 1 - (\text{depth}) * 1 / (20)). \quad (1)$$

Таким образом информация о глубине пикселей за пределами двадцатиметровой отметки будет удалена, а оставшаяся часть — нормализована.

Карта мировой позиции. Карта мировой позиции изображает текстуру, *R*, *G*, и *B* каналы которой будут представлять координаты пикселя в мировых координатах по *X*, *Y* и *Z* соответственно. Для ее расчета будем использовать следующее преобразование:

$$(x, y, z, 1)P_{inv}, \quad (2)$$

где x, y — координаты текстуры глубины от верхнего левого угла экрана; z — нормализованное значение глубины в данной точке; P_{inv} — матрица обратной проекции. Затем, представив полученные данные в виде цветов текстуры, передадим ее в следующий шаг алгоритма.

Алгоритм смешивания. Для устранения эффекта алиасинга необходимо смешивать изображения, полученные для левого и правого глаза при истинности следующего предиката:

$$(m0m1) \parallel ((|W0, W1|) < e), \quad (3)$$

где $m0, m1$ — значения маски смешивания; $W0, W1$ — векторы мировой позиции для левого и правого глаза соответственно. Таким образом достигается смешивание деталей без эффекта «раздвоения» изображения.

Альтернативным шагом смешивания может быть прямой выбор и смешивание значений точек, для которых

$$(|W0, W1|) < e \quad (4)$$

Однако такой алгоритм не подходит для приложений реального времени, так как имеет высокую вычислительную сложность.

Теоретические ограничения. Визуальные искажения растут пропорционально расстоянию до камеры, как продемонстрировано на рис. 2. Это означает, что объекты, находящиеся ближе 5 метров, практически никогда не будут подвергаться смешиванию. Однако, чем дальше объект от камеры, тем больше деталей опускается на межпиксельный уровень, а также требуется большее сглаживание.

Программная реализация. Для подтверждения работы алгоритма была написана пробная реализация на базе игрового движка *Unity3D*. Чтобы выделить информацию о глубине, маске смешивания и мировой позиции были написаны простые шейдеры. Финальная стадия алгоритма была обработана при помощи скрипта на *Python* и библиотеки для обработки изображений *Pillow*. Для основы была создана и визуализирована простая тестовая сцена (рис. 3).

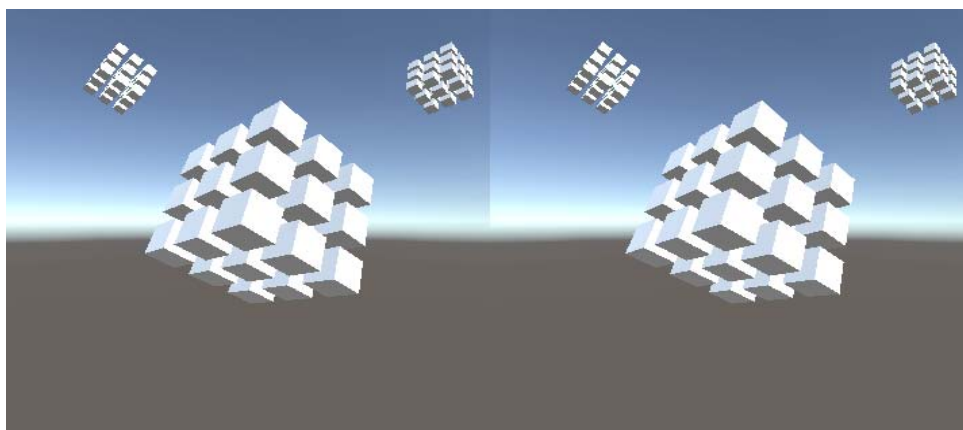


Рис. 3. Оригинальное изображение при отсутствии антиалиасинга

Затем однопроходным шейдером была извлечена карта глубины (рис. 4).

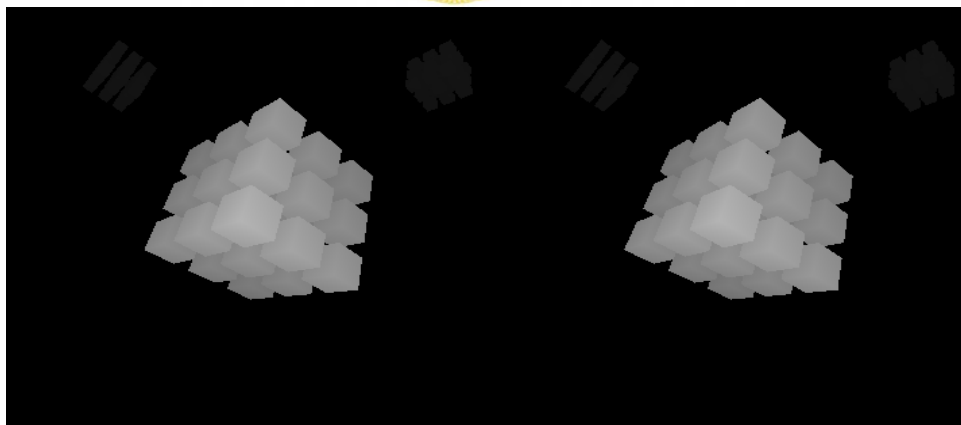


Рис. 4. Карта глубины

Далее с помощью формулы (1) сформирована усеченная маска смешивания (рис. 5). Участки с $A = 0$ будут смешаны 1:1.

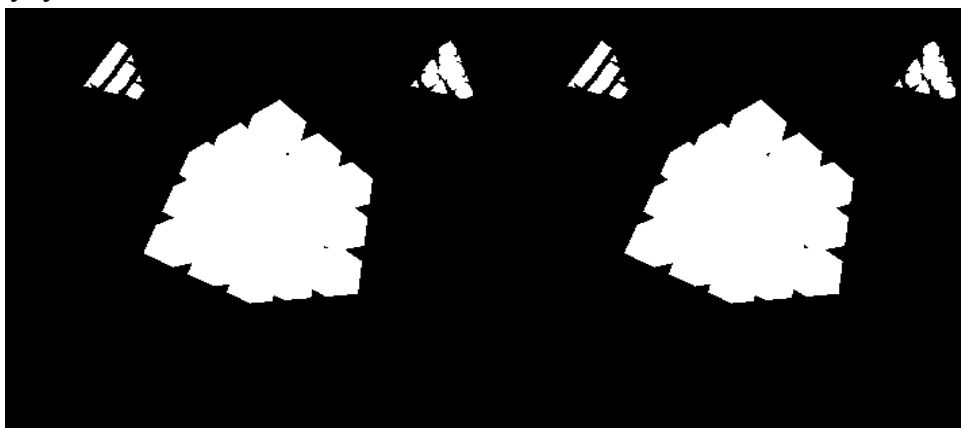


Рис. 5. Маска смешивания

Третьим шагом алгоритма было выполнение преобразования (2) и получение карты мировой позиции (рис. 6). Каналы R , G , B отвечают положению пикселя по оси X , Y , Z соответственно.

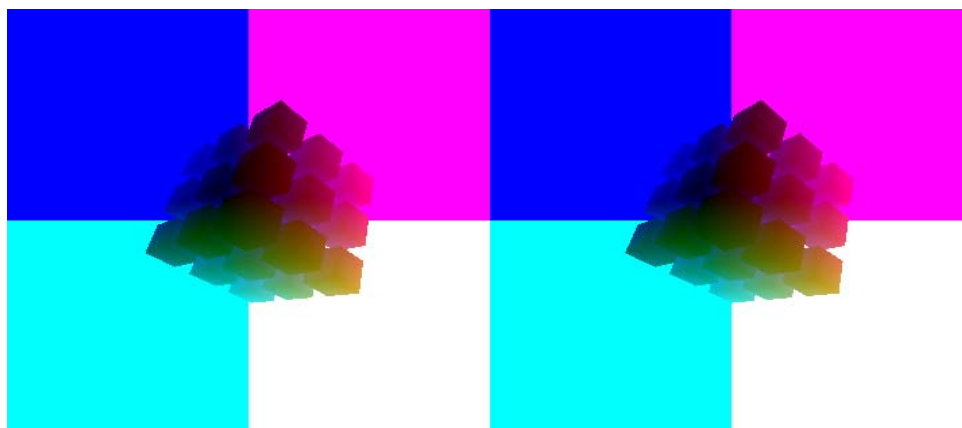


Рис. 6. Карта мировой позиции

Затем полученные текстуры были загружены в скрипт на *Python*, выполнивший смешивание по формуле (3) с параметром $\varepsilon=0.01$. В результате было получено сглаженное изображение (рис. 7). При сравнении рис. 7 и рис. 8, полученного при применении алгоритма смешивания с $\varepsilon=1$, заметно, что выбор ε является критическим для качества получаемой картинки.

При выборе слишком большого ϵ изображение начинает «раздваиваться», а при выборе слишком маленького значения эффект антиалиасинга пропадает.

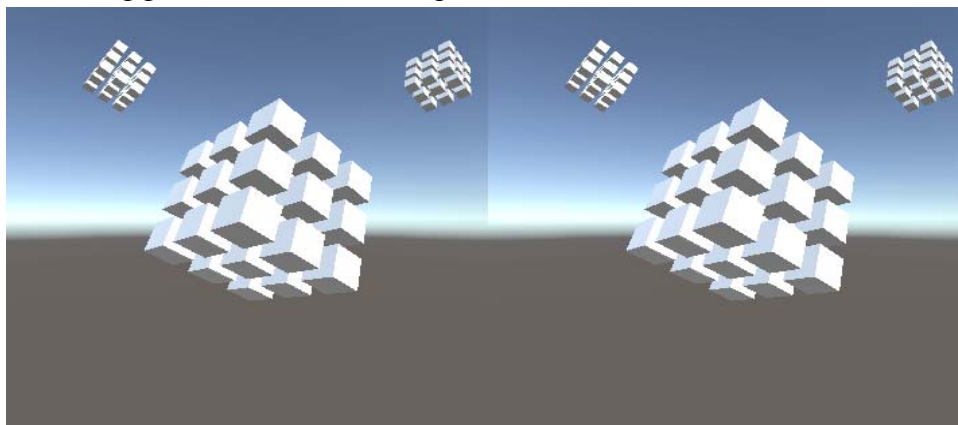


Рис. 7. Изображение с примененным антиалиасингом ($\epsilon=0.01$)

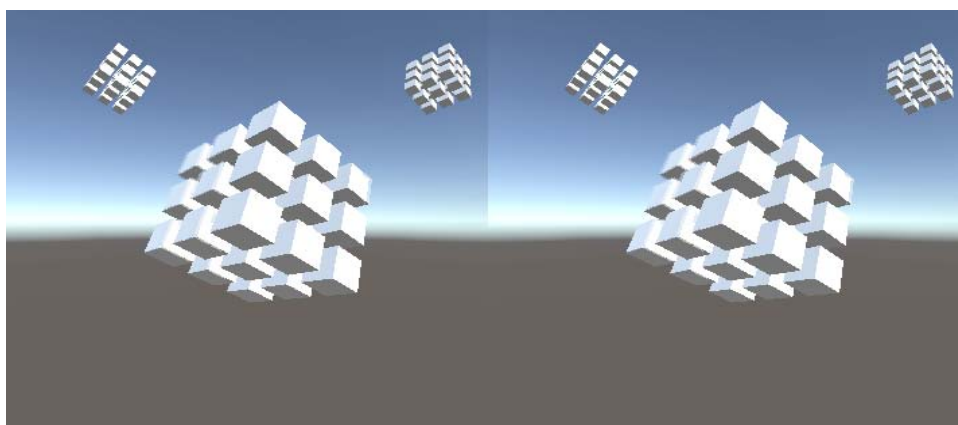


Рис. 8. Изображение с неправильно примененным алгоритмом при $\epsilon=1$

Заключение. Автором был предложен и программно реализован новый метод антиалиасинга, использующий особенности конвейера визуализации приложений виртуальной реальности. Перспективное направление улучшения алгоритма состоит в исключении областей, заведомо не совпадающих, путем обрезания по ближней маске. Отметим, возможно качественное улучшение эффекта при использовании камеры как начала координат.

Библиографический список.

1. Franklin, C. C. The aliasing problem in computer-generated shaded images / Franklin C Crow // *Communications of the ACM*, — Nov. 1977. — С. 57–63.
2. Fernando, N. Practical Morphological Anti-Aliasing / Fernando Navarro, Diego Gutierrez // *GPU Pro 2*, 2011. — С. 110.
3. Chu, A. VR Design: Transitioning from a 2D to a 3D Design Paradigm / Alex Chu // *Samsung Developer Conference*, 2014 — С. 13.