

УДК 004.42

В.М. Гусятин, Я.В. Чаговец, Д.Г. Кожушко

Харьковский национальный университет радиоэлектроники, Харьков

МЕТОД УСТРАНЕНИЯ КРАЕВОГО АЛИАСИНГА ПРИ СИНТЕЗЕ ИЗОБРАЖЕНИЙ ОБРАТНЫМ ТРАССИРОВАНИЕМ

Предлагается метод уменьшения количества дополнительно трассируемых лучей при устранении краевого алиасинга. В отличие от существующих методов введен критерий выбора количества уровней разбиения. Использование предложенного метода позволяет уменьшить вычислительную сложность при синтезе изображений методом обратного трассирования.

Ключевые слова: алиасинг, проекционный луч.

Введение

Актуальность и постановка задачи. Метод обратного трассирования – это один из наиболее перспективных методов синтеза изображений 3D сцен. Он позволяет фотореалистично визуализировать аналитически заданные поверхности, теневые эффекты, моделировать рассеяние и преломление световых лучей.

Сущность получения проекции 3D сцены на плоскость экрана методом обратного трассирования заключается в трассировании лучей из центра проекции через центры пикселей экрана. Данный подход получения проекции 3D сцены на плоскость экрана может привести к проявлению эффекта алиасинга (aliasing). Рассмотрим результат проекции 3D сцены на плоскость экрана как непрерывную функцию двух аргументов (пространственный сигнал). Данная функция устанавливает соответствие между каждой точкой плоскости экрана и значением интенсивности цвета в данной точке. Применим к данной функции метод точечной выборки, т.е. двумерную дискретизацию:

$$\text{sample}_{2D}(f(x, z)) = F, \quad (1)$$

где: $f(x, z)$ – исходная функция, задающая цвет в каждой точке плоскости проецирования; F – массив значений функции $f(x, z)$ в выбранных точках.

В этом случае при дискретизации часть информации может быть потеряна, если пространственная частота сигнала превышает половину частоты дискретизации [1]. Данный эффект, возникающий при синтезе изображения, приводит к неправильному отображению непрерывного сигнала проекции 3D объектов с высокой пространственной частотой на дискретной сетке экрана и называется алиасинг (aliasing). Эффект алиасинга проявляется в таких основных формах: краевой алиасинг (эффект блочных границ 3D объектов и теней); пропуск мелких деталей объектов или их мерцание при синтезе динамического изображения; алиасинг, возникающий при нанесении текстур (эффект муара, пропуск мел-

ких деталей текстур или их мерцание при синтезе динамического изображения).

Анализ проблемы

Существует ряд известных методов устранения алиасинга при синтезе изображения методом обратного трассирования:

Увеличение количества трассируемых лучей (supersampling) [2, 3]. Данный метод устранения алиасинга заключается в разбиении пикселя регулярной решеткой на совокупность субпикселей и трассировании лучей через центр каждого субпикселя. Реализация данного метода достаточно проста, однако приводит к значительному увеличению вычислительных затрат. Стохастическое трассирование увеличенного количества лучей (stochastic sampling) [4]. Суть данного метода заключается в стохастическом разбиении пикселя на субпиксели и трассировании лучей через центр каждого субпикселя. При этом артефакты алиасинга заменяются другими артефактами (“стохастический шум”), к которому человеческий глаз менее чувствителен. Данный метод позволяет уменьшить количество дополнительно трассируемых лучей, однако требует дополнительных вычислительных затрат на вычисление стохастического расположения дополнительных субпикселей. Вышеописанные методы устранения алиасинга позволяют устранить эффект краевого алиасинга и значительно уменьшить эффект муара, однако требуют значительного увеличения количества дополнительно трассируемых лучей даже в тех случаях, когда в этом нет необходимости. С целью устранения алиасинга и повышения эффективности метода обратного трассирования предлагается рассматривать пирамиду, образованную лучами, проходящими через вершины пикселя. Для экрана с разрешением N_z на N_y пикселей будет трассироваться $(N_z + 1) \times (N_y + 1)$ лучей.

При этом возможны такие варианты пересечения лучей пирамиды с 3D сценами: ни один из лучей не пересекает объекты сцены и ни один объект не

попал внутрь пирамиды пикселя (рис. 1, а); все четыре луча пересекают один объект (рис. 1, б); существует пересечение хотя бы одного луча пирамиды

пикселя с объектом, но количество лучей, попавших в этот объект, меньше четырех (рис. 1, в); объект находится внутри пирамиды пикселя (рис. 1, г).

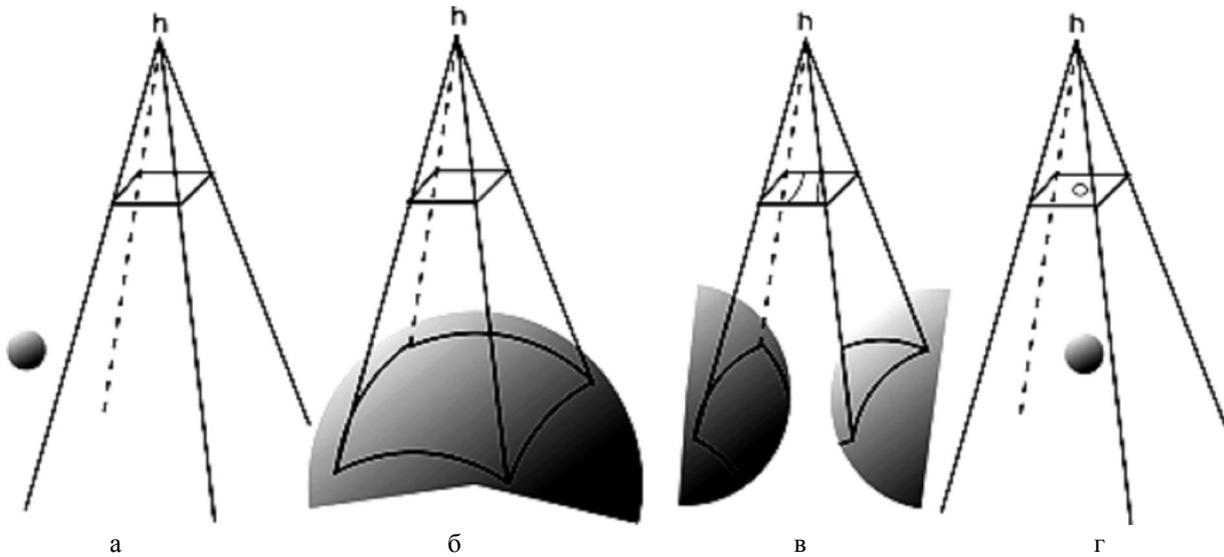


Рис. 1. Варианты пересечения пирамиды пикселя со сценой

В первом случае анализ пирамиды показывает отсутствие объектов в данной области сцены и данная пирамида больше не рассматривается. Во втором случае производится вычисление интегрального цвета текстур в формате, описанном в [5], и присвоение данного цвета соответствующему пикселю экрана. Четвертый вариант в данной работе не рассматривается.

Рассмотрим третий вариант, который приводит к появлению краевого алиасинга.

Метод адаптивного трассирования лучей

Для устранения данного артефакта предлагается алгоритм адаптивного увеличения количества дополнительно трассируемых лучей.

При таком подходе исходный пиксель рекурсивно разбивается на субпиксели в результате трассирования дополнительных проекционных лучей (ПЛ) через середины сторон и через центр пикселя (рис. 2).

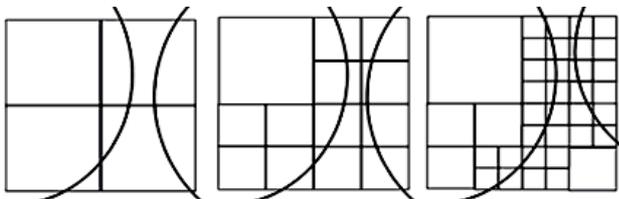


Рис. 2. Рекурсивное разбиение пикселя на субпиксели

Для каждого субпикселя определяется вариант пересечения его пирамиды с 3D сценой (возможен: первый, второй или третий вариант). Разбиение субпикселя на некотором уровне рекурсии (в дальней-

шем уровне разбиения) прекращается в случае отнесения его к первому или второму варианту, а также при достижении максимального предопределенного уровня разбиения, иначе разбиение продолжается. Запишем данный алгоритм в виде псевдокода:

```
void render(in центр_проекции, in экран-
ные_координаты_пикселя, out резуль-
тат_пересечения){
    рассчитываем 4 вектора наблюдения для данного
    пикселя;
    трассируем 4 ПЛ и определяем вари-
    ант_пересечения пирамиды со сценой;
    switch(вариант_пересечения){
    case 1:{ помечаем пиксель как пустой; }
    case 2:{ возвращаем результат пересечения и инте-
    гральный цвет; }
    case 3:{ pixel_subdivision(номер_уровня, координаты
    _пикселя, результат_пересечения_пирамид
    _дерево_трассирования);
        calc_result(дерево_трассирования, резуль-
    тат_пересечения); }
    case 4:{ необходимо произвести захват объекта
    (алгоритм не рассматривается; )} }
void pixel_subdivision(in номер_уровня, in координат
ы_пикселя, in результат_пересечения_пирамид, out
дерево_трассирования){
    расчет дополнительных 5-ти векторов на-
    блюдения и ПЛ;
    трассирование дополнительных ПЛ;
    из результат_пересечения_пирамиды и ре-
    зультата трассирования 5-ти дополнительных ПЛ
    пересечения формируются субпиксели;
    subpixel_check(номер_уровня, субпиксели,
    дерево_трассирования); }
```

```

void subpixel_check(in номер_уровня, in субпиксели,
out дерево_трассирования){
    for_each(субпиксель){
        определить вариант_пересечения пирами-
ды субпикселя со сценой;
        switch(вариант_пересечения){
            case 1,4:{ помечаем субпиксель как пустой;
continue; }
            case 2:{ помечаем соответствующий узел
дерева как терминальный;
            add_terminal_node(номер_уровня, субпик-
сель, дерево_трассирования); continue; } }
            case 3: {if(номер_уровня + 1 <
MAX_LEVEL){
                помечаем соответствующий узел дерева
текущего уровня как не терминальный;
                pixel_subdivision(номер_уровня + 1, координ
аты субпикселя,
                результат_пересечения субпикселя, дере-
во_трассирования); } }
    }
}
    
```

В функции `render` генерируются вектора наблюдения (ВН) [6], направленные в вершины пикселя экрана. Из центра проекции [6] в направлении этих векторов строятся ПЛ. В результате трассирования четырех ПЛ определяется вариант пересечения пирамиды пикселя со сценой. Функция `pixel_subdivision` работает в два этапа. На первом этапе производится расчет 5-ти дополнительных ВН и ПЛ. На втором этапе находится пересечения этих ПЛ со сценой. Запишем формулу для расчета этих дополнительных ВН.

Обозначим через \bar{v}_h вектор, идущий в верхний левый угол экрана. Обозначим через $\bar{v}_{\mu,v}^{i,j}$ ВН, идущий из центра проекции в вершину субпикселя, с учетом экранных координат его родительского пикселя (μ, v) (рис. 3). Экранными координатами (μ, v) пикселя называется соответственно номер строки μ и столбца v , в которых находится пиксель. Нумерация выполняется слева направо и сверху вниз, начиная с нуля.

Каждый ВН задается следующим образом:

$$\bar{v}_{\mu,v}^{i,j} = \bar{v}_h + (\mu + i \cdot d_L) \cdot \bar{v}_\mu + (v + j \cdot d_L) \cdot \bar{v}_v, \quad (2)$$

где \bar{v}_h – начальный вектор наблюдения; \bar{v}_μ и \bar{v}_v – векторы приращения; i, j – индексы дополнительных ВН в пределах пикселя; d_L – отношение размера субпикселя к размеру пикселя;

$$d_L = 2^{-L}, \quad (3)$$

где L – номер уровня разбиения.

Векторы приращения в системе координат наблюдателя:

$$\bar{v}_\mu = [0 \quad -dy \quad 0]^T; \quad \bar{v}_v = [0 \quad 0 \quad dz]^T, \quad (4)$$

где dy и dz – размеры пикселя.

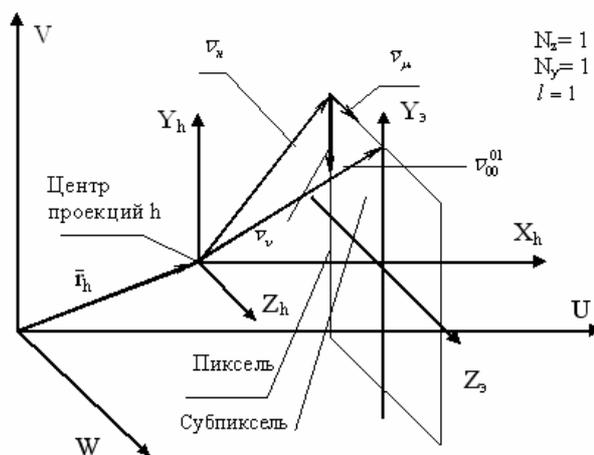


Рис. 3. Геометрические элементы задачи

Начальный ВН в системе координат наблюдателя:

$$\bar{v}_h = \left[dx \quad \frac{N_y}{2} dy \quad -\frac{N_z}{2} dz \right]^T, \quad (5)$$

где dx – расстояние от центра проекции до экрана.

Функция `subpixel_check` производит анализ вариантов пересечения пирамид субпикселей со сценой и формирует дерево трассирования для каждого пикселя (рис. 2). Дерево трассирования в каждой своей вершине содержит информацию о пересечении с объектами сцены. В зависимости от варианта пересечения пирамиды со сценой вершины могут быть терминальными или нетерминальными.

Максимальное количество уровней разбиения выбирается исходя из углового разрешения глаза человека, разрешения экрана отображающего устройства, а также расстояния, с которого оператор просматривает изображение на экране. В существующих мониторах обеспечивается угловое разрешение порядка 2^{-10} . В статье [7] показано, что угловое разрешение глаза человека порядка 2^{-13} . Чтобы полностью устранить эффект краевого алиасинга, необходимо уменьшить размеры пикселей современных мониторов (dy, dz) в 2^3 раз. Исходя из формулы (3) и углового разрешения современных устройств отображения максимальное количество уровней разбиения было выбрано $L = 3$.

Результаты

На рис. 4, а,б представлено изображение части киля самолета, для синтеза которого использовался метод обратного трассирования. При синтезе изображения, приведенного на рис. 4, а, использовался метод точечной выборки. Вследствие ограниченного количества трассируемых лучей в данном изобра-

жени легко заметить артефакты (эффект алиасинга), в том числе наличие блочных границ. На рис. 4, б представлено изображение, при синтезе которого использовался предложенный метод уст-

ранения алиасинга. Легко заметить, что применение предложенного метода позволяет устранить краевой алиасинг и повысить качество синтезируемого изображения методом обратного трассирования.

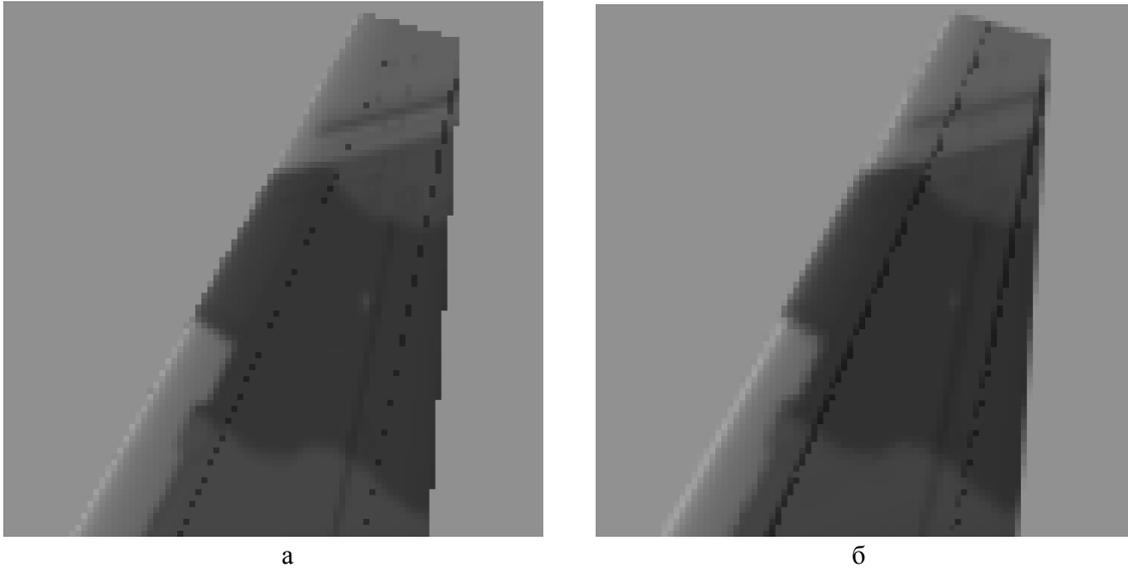


Рис. 4. Результаты моделирования

Выводы

Предложенный метод в отличие от [2 – 4] позволяет формировать дополнительно трассируемые лучи только по границе поверхности объекта, что позволяет повысить производительность алгоритма синтеза изображений методом обратного трассирования и полностью устранить эффект краевого алиасинга. С целью ограничения количества дополнительно трассируемых лучей предложена методика расчета максимального уровня разбиения исходя из углового разрешения глаза человека и устройства отображения.

Список литературы

1. Гольденберг Л.М. Цифровая обработка сигналов / Л.М. Гольденберг, Б.Д. Матюшкин, М.Н. Поляк. – М.: Радио и связь, 1990. – 256 с.
2. Molnar S. Efficient Supersampling Antialiasing for High-performance Architectures / S. Molnar // Technical Report 91-023. – University of North Carolina. – 1991.

3. Computer Graphics: Principles and Practice / J.D. Foley, A. van Dam, S.K. Feiner, J. Hughes. – 2nd ed. – Addison-Wesley, Reading, Mass. 1990.

4. Cook R.L. Stochastic Sampling in Computer Graphics / R.L. Cook // ACM Transactions on Graphics. – January 1986. – Vol. 5 (1). – P. 51-72.

5. Гусятин В.М. Упаковка векторных текстур в задачах синтеза изображений для систем визуализации / В.М. Гусятин, Я.В. Чаговец, Д.Г. Кожушко // Вестник НТУ «ХПИ». – 2005. – С. 4-10.

6. Гусятин В.М. Математическая модель геометрических преобразований для специализированных процессоров растровой графики / В.М. Гусятин // Радиоэлектроника и информатика. – 1997. – № 1. – С. 86-87.

7. James A. Ferwerda. Elements of Early Vision for Computer Graphics / James A. Ferwerda // IEEE Computer Graphics and Applications. – Sept/Oct. 2001. – Vol. 21, no. 5. – P. 22-33.

Поступила в редколлегию 18.03.2010

Рецензент: д-р техн. наук, проф. С.Г. Удовенко, Харьковский национальный университет радиоэлектроники, Харьков.

МЕТОД УСУНЕННЯ КРАЙОВОГО АЛІАСІНГА ПРИ СИНТЕЗІ ЗОБРАЖЕНЬ ЗВОРОТНИМ ТРАСУВАННЯМ

В.М. Гусятин, Я.В. Чаговец, Д.Г. Кожушко

Пропонується метод зменшення кількості додатково трасованих променів при усуненні крайового аліасінгу. На відміну від існуючих методів введений критерій вибору кількості рівнів розбиття. Використання запропонованого методу дозволяє зменшити обчислювальну складність при синтезі зображень методом зворотного трасування.

Ключові слова: аліасинг, проєкційний промінь.

THE METHOD OF REMOVING THE EDGE ALIASING IN THE 3D IMAGE RENDERING TASKS USING RAY-TRACING

V.M. Gusyatın, Ya.V. Chagovets, D.G. Kozhushko

We propose the method for removing the edge aliasing in the 3D image rendering tasks using ray-tracing, which allows to increase the quality of 3D images. Unlike existent methods the criterion of choice of amount of levels of breaking up is entered. The use of the offered method allows to decrease calculable complication at the synthesis of images the method of the reverse tracing.

Keywords: aliasing, projection ray.