

В.М. ГУСЯТИН, канд. техн. наук, ХНУРЭ (г. Харьков),
А.Е. ГРОМЕНКО, ХНУРЭ (г. Харьков)

ПОДХОД К ВИЗУАЛИЗАЦИИ ТРИАНГУЛИРОВАННЫХ ПОВЕРХНОСТЕЙ МЕТОДОМ ОБРАТНОГО ТРАССИРОВАНИЯ ЛУЧЕЙ

У статті введено поняття гранично триангульованої поверхні, розглянуті способи її отримання і завдання, приведено алгоритм пошуку точок перетинання проєкційного променя з трикутником цієї поверхні. Для запропонованого підходу розглянуті особливості реалізації технології субдискретизації екрану, виконана оцінка необхідної пропускну здатності шини даних, проведене порівняння по обчислювальних витратах з архітектурою SaarCOR, запропоновано структуру спецобчислювача, який оптимізований для паралельно-конвексної архітектури з безперервним потоком вхідних даних.

In the article it is introduced conception of ultimate triangulated surface, obtaining method and representation such surface is considered. Intersection algorithm projection ray with triangle is described. For proposed approach is analyzed features of supersampling realization, rough estimate for capacity of data bus is performed, computing costs comparison with SaarCOR architecture is introduced, special numerator architecture is proposed.

Актуальность и постановка проблемы. В настоящее время изображения трехмерных сцен для систем визуализации (СВ) синтезируются в основном двумя методами: прямым и обратным трассированием лучей [1]. На современном этапе развития компьютерной графики наиболее развит метод прямого трассирования. Метод обратного трассирования, хотя и требует больших вычислительных и аппаратных затрат для своей реализации, но обладает рядом существенных преимуществ: обработка динамических источников света и теней, расчет вторичного освещения, преломление и отражение лучей света [1, 2]. Все это позволяет данным методом синтезировать более реалистичное изображение. Поэтому разработка подсистем и узлов СВ, работающей по обратному методу, актуальна. Сейчас это направление активно исследуется как в нашей стране, так и за рубежом. Имеются разработки в области синтеза изображений облаков [3], рельефа [4], объектов, состоящих из совокупности поверхностей второго порядка и плоскостей [5]; рассматриваются вопросы визуализации источников света [6]. Разработкой спецпроцессоров, работающих по методу обратного трассирования, занимаются коллективы из Саарлендского университета (Saarland University) [7] и университета Торонто (University of Toronto) [8]. Существует коммерческая реализация аппаратного ускорителя метода обратного трассирования под названием PURE, разработанного компанией Advanced Rendering Technology (www.art-render.com).

В современной компьютерной графике из-за преобладания метода прямого трассирования представление 3D-объектов в виде триангулированной поверхности (ТП) является стандартом. К преимуществам ТП относятся: простота и универсальность её математического представления, удобство построения, трансформации и преобразования; возможность получения в триангулированном виде геометрии многих объектов реального мира. Существует множество библиотек ТП для свободного использования, например, в Стэнфордском университете (The Stanford 3D Scanning Repository). Решение задачи эффективного синтеза изображений ТП методом обратного трассирования является актуальным, так как позволит сохранить для этого метода преемственность ранее полученных результатов, уменьшить время на подготовку или изменение 3D сцены и моделей объектов в ней.

Классический подход к синтезу изображений ТП обратным методом подразумевает поиск точки пересечения проекционного луча (ПЛ) с треугольником и интерполяцию освещенности. Широко известен алгоритм поиска пересечения проекционного луча с треугольником [9], а также его модификации для оптимальной обработки ТП [10]. Данные алгоритмы считаются наиболее приемлемыми для аппаратной реализации [8] по сравнению с другими классическими методами [11, 12], но не позволяют достичь быстродействия соизмеримого с прямым методом. При этом синтезированные изображения наравне с достоинствами обратного метода имеют все недостатки изображений, синтезированных прямым методом. Основным недостатком следует считать отсутствие взаимосвязи угловой погрешности СВ и геометрических искажений, возникающих в процессе синтеза изображения поверхности. В этом случае с определенных ракурсов видны плоские грани треугольников поверхности, возникают неточности в процессе расчета освещенности. Очевидно, что при классическом подходе к синтезу изображений увеличение количества треугольников для более точного представления геометрии поверхности ведет к существенному росту объема вычислений, особенно при высоких разрешениях экрана. Введение различных классификационных описаний геометрии трехмерных объектов [13], построение восьмеричных деревьев (octree) [1] уменьшают зависимость времени синтеза изображения от количества треугольников, но резко усложняют аппаратную реализацию. Варианты спецвычислителей, основанные на классическом подходе к синтезу ТП, предложены коллективами из Саарленда [14] и Торонто [8].

Таким образом, поиск подходов, которые устраняют недостатки, присущие классическим методам синтеза изображений ТП, работают в реальном времени и аппаратно реализуются в виде простой параллельно-конвейерной архитектуры, является актуальной задачей.

Целью статьи является описание разработанного подхода к эффективному синтезу высокореалистичных изображений ТП методом обратного трассирования и особенностей его реализации.

Предельно триангулированная поверхность. Определим, что ТП является предельно триангулированной (ПТП) для расстояния $D_{\min}$, если она состоит из треугольников, максимальная длина $L_{\max}$ стороны которых удовлетворяет условию:

$$L_{\max} < 2 \cdot D_{\min} \operatorname{tg} \frac{\gamma_{\text{доп}}}{2}, \quad (1)$$

где $\gamma_{\text{доп}}$ – допустимая угловая погрешность СВ [15].

Из (1) видно, что для любого расстояния больше $D_{\min}$ угловой размер треугольника всегда меньше $\gamma_{\text{доп}}$, т.е. длина проекции любой стороны треугольника на плоскость экрана меньше линейного размера пикселя. В таком случае при синтезе изображения ПТП выполнение интерполяции нормали или освещенности в пределах треугольника является избыточной операцией. Это позволяет упростить процесс визуализации, а также избежать искажений, присущих методам интерполяции [16].

Значение расстояния $D_{\min}$ в системном плане обычно получают из технических характеристик на проект СВ.

Получение ПТП с заданной длиной $L_{\max}$ стороны треугольника возможно двумя способами:

- сканирование геометрии объекта реального мира;
- уменьшение размеров треугольников ТП (subdivision), в дальнейшем, просто разбиение.

Первый способ более точный, но его не всегда возможно применить из-за отсутствия необходимого объекта реального мира, аппаратных или программных средств для сканирования, а также из-за трудоемкости процесса.

Второй способ в качестве исходных данных использует ТП, полученные по первому способу или сконструированные в графических пакетах (3Dsmax, Maya и т.п.). Для уменьшения длины стороны треугольников исходной ТП выполняется разбиение треугольников согласно некоторому методу. В этом направлении выполнен ряд исследований [17 – 20]. Условно их можно разделить на методы, основывающиеся на нормалях в вершинах ТП [17, 18], и методы, основывающиеся на топологии соседних треугольников [19, 20]. Эти методы имеют свои достоинства и области применения, но для подготовки ПТП предпочтительнее методы [17, 18], так как позволяют более точно приблизиться к исходной поверхности, в частности, на границах треугольников ТП. Вычислительная сложность алгоритма разбиения не имеет большого значения в силу того, что подготовка ПТП выполняется на препроцессорном этапе. Для уменьшения размеров треугольников ТП наиболее подходит предложенный в [18] метод, при котором нормали плоскостей двух соседних треугольников в процессе увеличения числа итераций разбиения приближаются друг к другу.

Для хранения ПТП из-за существенного увеличения количества треугольников необходимо больше оперативной памяти, чем для обычной ТП. Вследствие этого подход целесообразно использовать для синтеза изображения поверхностей объектов, которые имеют относительно всей сцены небольшие линейные размеры и требуют высокой детализации. Например, такие объекты как человек, животные, строения, ЛА и прочая техника. Для уменьшения объема памяти предлагается использовать относительный способ задания вершин всех треугольников относительно некоторого начального:

$$T_{\text{начало}} = \{V_1, V_2, V_3\} \text{ и } T_{\text{след}} = \{\vec{v}_N\}, \quad (2)$$

где V_1, V_2, V_3 – координаты вершин в системе координат (СК) объекта; $\vec{v}_N$ – вектор смещения в СК объекта вершины нового треугольника относительно одной из общих вершин.

Важно отметить, что в СК объекта координаты вершин и вектора смещения (2) могут быть представлены меньшим количеством двоичных разрядов по сравнению с координатами в базовой системе координат. В дополнение к этому, максимальное значение модуля вектора смещения v_N ограничено условием (1).

Алгоритм поиска точки пересечения ПЛ с треугольниками ПТП.

Считаем, что ПЛ располагаются по столбцам и строкам и проходят через центры пикселей плоского экрана, центр проекций (ЦП) равноудален от углов экрана, количество пикселей по высоте и ширине чётное. Тогда с учетом условия (1) треугольник может иметь общую точку только с одним ПЛ. Такая ситуация показана на рис. 1, где ЦП обозначен точкой h ; ξ – плоскость экрана; $V_1V_2V_3$ – обрабатываемый треугольник, hP_1, hP_2 – проекционные лучи, и β – плоскость, определяемая условиями:

$$\begin{cases} \beta \parallel OY, \\ \beta \perp OZ, \\ h \in \beta, \end{cases} \quad (3)$$

где OYZ – система координат экрана.

Этап 1. Определяются указатели положения вершин треугольника. Выполняемые действия аналогичны для всех вершин треугольника, в качестве примера на рис. 1 приведена вершина V_1 .

Указатель положения N_{Zvi} вершины V_i относительно столбцов ПЛ вычисляется согласно соотношению (для вершины V_1 , см. рис. 1):

$$N_{Zvi} = \left\lceil \frac{d_{zvi} + 0,5 \cdot l_{vi}}{l_{vi}} \right\rceil, \quad (4)$$

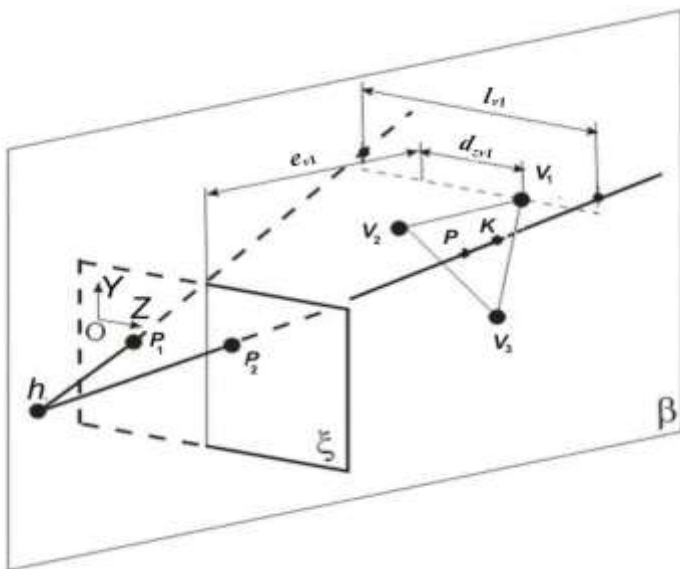


Рис. 1. Геометрические элементы задачи

где номер вершины $i = 1, 2, 3$; d_{vi} – расстояние от вершины V_i до плоскости β ; l_{vi} – линейный размер пикселя на расстоянии e_{vi} (расстояние от вершины V_i до плоскости экрана ξ); $\lceil x \rceil$ – округление вверх до ближайшего целого числа.

Введем плоскость α , определяемую условиями:

$$\begin{cases} \alpha \parallel OZ, \\ \alpha \perp OY, \\ h \in \alpha. \end{cases} \quad (5)$$

Указатель положения N_{Yvi} относительно ближайшей к вершине V_i строки ПЛ вычисляется согласно соотношению:

$$N_{Yvi} = \left\lceil \frac{d_{yvi} + 0,5 \cdot l_{vi}}{l_{vi}} \right\rceil, \quad (6)$$

где d_{yvi} – расстояние от вершины V_i до плоскости α .

В результате имеем положение вершин треугольника относительно некоторого ПЛ с индексами строки и столбца ($N_{ZПЛ}, N_{YПЛ}$).

Этап 2. Определяется факт пересечения треугольника ПЛ. Считается, что треугольник имеет точку пересечения с ПЛ, если выполняется условие:

$$(N_{Zv1} \neq N_{Zv2} \text{ OR } N_{Zv1} \neq N_{Zv3}) \text{ AND } (N_{Yv1} \neq N_{Yv2} \text{ OR } N_{Yv1} \neq N_{Yv3}). \quad (7)$$

Индексы ($N_{ZПЛ}, N_{YПЛ}$) ПЛ, который пересекает треугольник равны:

$$\begin{aligned} N_{ZПЛ} &= \min(N_{Zv1}, N_{Z2}, N_{Zv3}); \\ N_{YПЛ} &= \min(N_{Yv1}, N_{Yv2}, N_{Yv3}). \end{aligned} \quad (8)$$

Этап 3. Поиск точки пересечения. Вычисление координат реальной точки K пересечения ПЛ, имеющего индексы $(N_{ZПЛ}, N_{YПЛ})$, и обрабатываемого треугольника не выполняется. В качестве условной точки пересечения P принимается точка на ПЛ такая, что (см. рис. 1):

$$hP = hV_m, \quad (9)$$

где V_m – вершина на расстоянии $e_{vm} = \min(e_{vi})$, где $i = 1, 2, 3$, т.е. ближайшая к экрану вершина треугольника.

На данном этапе координаты точки P вычисляются через параметр t , такой что $P = h + \sqrt{t} \cdot \vec{O}$, где $\vec{O}$ является направляющим вектором ПЛ. Параметр t равен:

$$t = e_{vm}^2 + d_{zvm}^2 + d_{yvm}^2. \quad (10)$$

Из анализа возможных положений треугольника относительно ПЛ угловая ошибка точки пересечения γ_{PK} удовлетворяет условию:

$$\gamma_{PK} \leq 0,5 \cdot \gamma_{доп}. \quad (11)$$

Нормалью поверхности $\mathbf{n}$ в точке пересечения P является нормаль плоскости треугольника. По известным координатам вершин треугольника нормаль $\mathbf{n}$ вычисляется тривиально.

Этап 4. Полученная точка пересечения P и нормаль $\mathbf{n}$ помещаются в кадровый буфер, который обозначим следующим образом:

$$(N_{ZПЛ}, N_{YПЛ}) \rightarrow (P, \mathbf{n}). \quad (12)$$

Этапы 1 – 4 выполняются для всех треугольников ПТП.

Субдискретизация экрана. Необходимо отметить, что при нанесении текстуры для повышения реалистичности (устранения ступенчатости (antialiasing)) применяется технология субдискретизации экрана (supersampling). В этом случае при классическом подходе увеличение степени субдискретизации экрана приводит к увеличению объема вычислений пропорционально увеличению количества ПЛ. В рассмотренном алгоритме вычислительные затраты практически не зависят от количества ПЛ, так как выполняется обработка множества треугольников, а не ПЛ. В тоже время, при субдискретизации экрана для работы алгоритма согласно условию (1) необходимо уменьшить длину $L_{\max}$ стороны треугольника. Для этого в реальном времени выполняется разбиение исходного треугольника ПТП в его же плоскости на дочерние треугольники согласно Серпинскому [21]. Разбиение выполняется при условии:

$$\Delta d_{v\max} < l_{vm}, \quad (13)$$

где l_{vm} – линейный размер пикселя на расстоянии e_{vm} ;

$$\Delta d_{\text{vmax}} = \max \left(\begin{array}{l} |d_{zv1} - d_{zv2}|, |d_{zv1} - d_{zv3}|, |d_{zv3} - d_{zv2}|, \\ |d_{yv1} - d_{yv2}|, |d_{yv1} - d_{yv3}|, |d_{yv3} - d_{yv2}| \end{array} \right).$$

Из (13) следует, что количество разбиений и, соответственно, обрабатываемых дочерних треугольников уменьшается при увеличении расстояния между ПТП и центром проекций. Также на количество разбиений по Серпинскому влияет расположение исходного треугольника относительно плоскости экрана.

Для вершины V_k дочернего треугольника, расположенной посередине стороны V_iV_j , вычисление линейного размера пикселя l_{vk} , расстояний $d_{zv k}$ и $d_{yv k}$ существенно упрощается, а именно:

$$\begin{aligned} l_{vk} &= 0,5 \cdot (l_{vi} + l_{vj}); \\ d_{zv k} &= 0,5 \cdot (d_{zvi} + d_{zvj}); \\ d_{yv k} &= 0,5 \cdot (d_{yvi} + d_{yvj}). \end{aligned} \tag{14}$$

Дальнейшая обработка дочернего треугольника выполняется согласно этапам 1 – 4.

Как следует из алгоритма поиска точки пересечения, его особенностью является небольшая вычислительная сложность: отсутствуют операции с плавающей точкой, используются только арифметические функции. Для большинства обрабатываемых треугольников выполняются только этапы 1, 2, количество треугольников, для которых выполняются этапы 3, 4, пропорционально количеству ПЛ. Кроме того, количество матричных преобразований координат минимально за один кадр. Это связано с тем, что параметр t (выражение 10) инвариантен к СК; поиск точек пересечения для всех треугольников ПТП осуществляется в её СК, для чего достаточно выполнить матричные преобразования только координат ЦП и нормалей плоскостей α и β в ту же систему координат.

С точки зрения аппаратной реализации предложенный алгоритм хорошо подходит для параллельно-конвейерной архитектуры. Действительно, обработка соседних треугольников и вершин не взаимосвязана, этапы 1 и 2 алгоритма можно выполнять параллельно с этапом 3 (расчетом параметра t точки пересечения и нормали к поверхности), в вычислениях используется целочисленная арифметика, отсутствуют циклы, условные или безусловные переходы.

Оценка необходимой пропускной способности шины данных.

Произведем оценку максимального количества треугольников, которые возможно обработать на современных аппаратных средствах за время одного кадра. Очевидно, что узким местом подхода является необходимость передачи большого числа структур данных (2) на обработку. Основная нагрузка ложится на пропускную способность шины данных между оперативной памятью, хранящей ТП, и спецвычислителем, синтезирующим изображение.

Определим размер памяти необходимый для хранения (2). Поля структуры V_1, V_2, V_3 содержат координаты вершин треугольника, поэтому их разрядность зависит от линейных размеров поверхности. Пусть $l_{\text{ТП}}$ – максимальный линейный размер ПТП в произвольном направлении, α – минимальный угол обзора экрана. Определим, что расстояние $D_{\min}$ до поверхности удовлетворяет условию

$$D_{\min} \geq \frac{l_{\text{ТП}}}{2 \cdot \text{tg}(0,5\alpha)}, \quad (15)$$

т.е. на таком расстоянии ТП практически полностью помещается на экране.

Количество двоичных разрядов для представления координаты вершины треугольника в СК поверхности вычисляется согласно соотношению

$$n_{\text{разрядов}}^{\text{вершина}} = \left\lceil \log_2 \frac{l_{\text{ТП}}}{\Delta l} \right\rceil, \quad (16)$$

где Δl – является весом младшего двоичного разряда координаты.

Вес младшего разряда предлагается вычислять согласно соотношению

$$\Delta l = 2 \cdot D_{\min} \cdot \text{tg} \frac{\gamma_{\text{длп}}}{2 \cdot M}, \quad (17)$$

где M – коэффициент позиционирования вершин в СК объекта.

Например, если принять $M = 2^5$, то это предполагает задание длины стороны треугольника с шагом не хуже $0,03 \cdot l_{\text{max}}$.

При относительном задании ПТП максимальная длина вектора смещения u из (3) определяется условием (1). Количество разрядов для его представления определяется из соотношения

$$n_{\text{разрядов}}^{\text{смещение}} = \left\lceil \log_2 \frac{L_{\text{max}}}{\Delta l} \right\rceil. \quad (18)$$

В предельном случае условия (15) согласно (16) имеем $n_{\text{разрядов}}^{\text{вершина}} = 15$ и с учетом (1), (17) для рассмотренного примера согласно (18) – $n_{\text{разрядов}}^{\text{смещение}} = 5$.

Длина стороны треугольника ТП определяется согласно условию (1). Положим, что ТП состоят из правильных треугольников. Оценивать количество треугольников предлагается согласно соотношению

$$N_t \approx \frac{S_{\text{поверх}}}{S_t}, \quad (19)$$

где $S_{\text{поверх}}$ – площадь поверхности; S_t – площадь треугольника ПТП.

Необходимая пропускная способность B_{data} рассчитывается согласно соотношению

$$B_{\text{data}} \approx \frac{V_T \cdot N_t}{T_{\text{кадра}}}, \quad (20)$$

где $T_{\text{кадра}} = 33 \text{ мс}$ – является максимальным периодом обновления кадра, который обусловлен психофизическими особенностями зрения человека; $V_T = 16 \text{ бит}$ – размер вектора смещения $\vec{v}_N$ из (2).

По известной максимальной пропускной способности современных шин данных [22] и соотношению (20) выполнен расчет максимального количества треугольников N_t , данные о которых можно передать за время обновления кадра (см. табл. 1).

Таблица 1

№	Шина данных	Пропускная способность, МБ/с	V_T , байт	N_t , шт.
1.	AGP 8x	2000	2	$3,3 \cdot 10^7$
2.	PCIExpress 16	4000	2	$6,6 \cdot 10^7$
3.	PCIExpress 32	8000	2	$1,32 \cdot 10^8$
4.	DDR553 двухкан.	8533	2	$1,4 \cdot 10^8$
5.	DDR2 800 двухкан.	12800	2	$2,11 \cdot 10^8$
6.	FB-DIMM 566/667	21504	2	$3,54 \cdot 10^8$

Данные по необходимой пропускной способности B_{data} для различных поверхностей сведены в табл. 2, в которой обозначено: D – расстояние от ЦП до поверхности; $\rho_{\text{пов}}$ – занимаемая поверхностью площадь экрана из расчета, что видимой является $0,25 \cdot S_{\text{пов}}$; $N_{\text{суб}}$ – степень субдискретизации экрана; N_t^{max} – максимальное число обработанных треугольников с учетом их разбиения по Серпинскому и условия (13).

Таблица 2

№	Название	$S_{\text{пов}}, \text{м}^2$	$D_{\text{min}}, \text{м}$	$D, \text{м}$	$\rho_{\text{пов}}, \%$	N_t , шт	B_{data} , МБ/с	$N_{\text{суб}}$	N_t^{max} , шт
1.	Человек	1,8	2	2	21	$1,1 \cdot 10^6$	127,16	1	$1,1 \cdot 10^6$
				2	21			4	$4,4 \cdot 10^6$
				5	4			4	$1,1 \cdot 10^6$
2.	Самолет	700	20	20	83	$4,2 \cdot 10^6$	485,51	1	$4,2 \cdot 10^6$
				20	83			4	$16,4 \cdot 10^6$
				45	16			4	$4,2 \cdot 10^6$
				90	4			16	$4,2 \cdot 10^6$

Из табл. 1 и 2 следует, что при существующей пропускной способности возможно передать в спецвычислитель данные о геометрии порядка 100 объектов. В перспективных разработках компании IBM предполагается введение стандартов пропускной способности шины данных с памятью до 75 ГБ/с. Кроме того, из табл. 2 следует, что при заданной степени

субдискретизации экрана и удалении объекта количество фактически обрабатываемых треугольников $N_t^{\max}$ уменьшается.

Оценка вычислительных затрат. Предложенный подход сравнивается по вычислительным затратам с разработанным научным коллективом (J. Schmittler, I. Wald, P. Slusallek и др.) из Саарлендского (Saarland, Germany) университета методом, заложенным в архитектуру для обратного трассирования лучей SaarCOR. Для сравнения выбран этап поиска точки пересечения как имеющий наибольший удельный вес в вычислительных затратах. Сведения по реализации архитектуры SaarCore взяты из [7, 14, 23]. Данные о количестве операций по сравниваемым архитектурам приведены в табл. 3. В скобках для предложенного подхода указано количество операций для вершины дочернего треугольника.

Таблица 3

№	Блок	Add	Mul	Div	Cmp	Shift
Предложенный подход						
1	Количество операций на вершину $N_{\text{верш}}^{\text{опер}} (N'_{\text{верш}}^{\text{опер}})$					
	Размер пикселя $l_{vi} (l_{vk})$	3(1)	3(0)	0(0)	0(0)	2(2)
	Индекс строки $N_{zvi} (N_{zvi})$	4(2)	3(0)	1(1)	0(0)	0(1)
	Индекс столбца $N_{yvi} (N_{yvi})$	4(2)	3(0)	1(1)	0(0)	0(1)
2	Определение факта пересечения, $N_{\text{перес}}^{\text{опер}}$	0	0	0	4	0
3	Определение параметра t , $N_t^{\text{опер}}$	2	3	0	0	0
Архитектура SaarCore						
1	Traversal, $N_{\text{trav}}^{\text{опер}}$	4	4	0	13	0
2	Transformation, $N_{\text{trans}}^{\text{опер}}$	9	9	0	0	0
3	Intersection, $N_{\text{inter}}^{\text{опер}}$	3	2	1	0	0

Общее число операций для синтеза изображения сцены для предложенного подхода $N_{\text{предл}}^{\text{общее}}$ и архитектуры SaarCOR $N_{\text{SaarCOR}}^{\text{общее}}$ вычисляются согласно соотношениям:

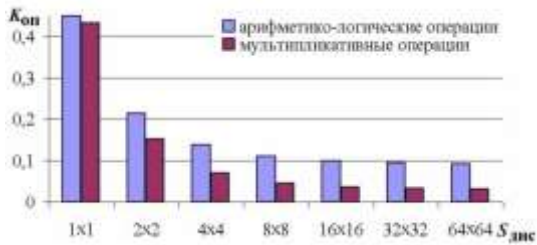
$$N_{\text{предл}}^{\text{общее}} = T_{\Delta} (3 \cdot N_{\text{верш}}^{\text{опер}} + N'_{\text{верш}}^{\text{опер}} (V - 3) + N_{\text{перес}}^{\text{опер}} K) + 2^{S_d} N_t^{\text{опер}} T_{\text{ПЛ}} R_{\text{ПЛ}}; \quad (21a)$$

$$N_{\text{SaarCOR}}^{\text{общее}} = R_{\text{ПЛ}} \cdot 2^{S_d} T_{\text{ПЛ}} \cdot (2 \cdot N_{\text{trans}}^{\text{опер}} + N_{\text{inter}}^{\text{опер}} + 4 \cdot N_{\text{trav}}^{\text{опер}}), \quad (21б)$$

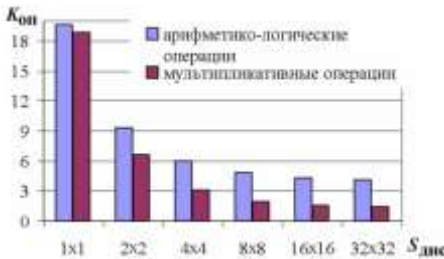
где S_d – степень субдискретизации экрана; $V = 0,5 \cdot (2 + 2^{S_d}) \cdot (1 + 2^{S_d})$ – количество вершин после разбиений по Серпинскому [21]; $K = 4^{S_d}$ – количество треугольников после разбиений [21]; T_Δ – количество треугольников в сцене; $R_{ПЛ}$ – количество ПЛ; $T_{ПЛ}$ – среднее количество треугольников на ПЛ.

Соотношение (21a) не позволяет учесть расстояние от треугольника до ЦП в сцене, поэтому количество разбиений исходного треугольника рассчитывается для расстояния $D_{\min}$, что в большинстве случаев приводит к избыточному количеству обрабатываемых треугольников (см. табл. 2). Также необходимо отметить, что в отличие от предложенного подхода, в архитектуре SaarCOR используются операции с плавающей точкой.

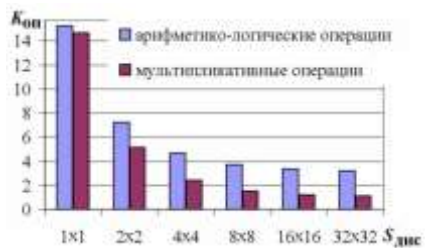
Для рассматриваемых подходов на рис. 2 приведено сравнение количества арифметико-логических и мультипликативных операций в зависимости от степени субдискретизации экрана $S_{\text{дис}}$. Рассматриваются идеальный случай (рис. 2a), сцены "cruiser" (рис. 2б) и "office" (рис. 2в), приведенные в качестве примера в [14].



а) идеальный случай



б) сцена "cruiser"



в) сцена "office"

Рис.2. Сравнение вычислительных затрат

Для сравнения вычислительных затрат вводится коэффициент $K_{\text{оп}}$, который вычисляется с учетом (21) согласно соотношению:

$$K_{\text{оп}} = \frac{N_{\text{предл}}^{\text{общее}}}{N_{\text{SaarCOR}}^{\text{общее}}}. \quad (22)$$

Под идеальным случаем понимается соответствие одного треугольника проекционному лучу; для сцен принималось $T_{\Delta}^{\text{office}} = 4 \cdot 10^8$, $T_{\Delta}^{\text{cruiser}} = 2,4 \cdot 10^8$, $T_{\text{ПЛ}}^{\text{office}} = 10$, $T_{\text{ПЛ}}^{\text{cruiser}} = 10$ для предложенного подхода и $T_{\text{ПЛ}}^{\text{office}} = 15$, $T_{\text{ПЛ}}^{\text{cruiser}} = 7$ для архитектуры SaarCOR, $R_{\text{ПЛ}} = 786432$.

Из рис. 2а видно, что в идеальном случае предложенный метод эффективнее SaarCOR по вычислительным затратам, особенно при увеличении степени субдискретизации экрана. Согласно рис. 2б и 2в в реальных сценах наблюдается проигрыш по количеству операций, который уменьшается при увеличении степени субдискретизации экрана. Важно отметить:

- синтезированное согласно предложенному подходу изображение имеет допустимую угловую погрешность геометрии поверхностей, которая при данных условиях не выдерживается в архитектуре SaarCOR;

- дальнейшее увеличение детализации объектов сцен не приведет к существенному росту вычислительных затрат для предложенного подхода, так как поверхности уже являются ПТП;

- для предложенного подхода на этом этапе заканчивается обработка треугольника, а в архитектуре SaarCOR необходимо еще выполнить интерполяцию освещенности по Фонгу [14].

Аппаратная реализация. На рис. 3 представлена структурная схема спецвычислителя, работающего по предложенному подходу.

Спецвычислитель выполняет обработку одного треугольника ПТП. Как видно из рис. 3, вычисление координат нормали и параметра точки пересечения выполняется параллельно с обработкой вершин треугольника. Конвейер предложенного спецвычислителя состоит из 5 этапов и, согласно алгоритму поиска точки пересечения поток обрабатываемых данных не прерывается. Минимальное время обработки одного треугольника ПТП достигается при аппаратной реализации и параллельном выполнении этапов 3 и 4 конвейера для каждого дочернего треугольника. Естественно, что в таком случае необходимо в блоке разбиения (этап 2) одновременно формировать все дочерние треугольники. Для ускорения обработки всей ПТП возможно установить необходимое число параллельно работающих спецвычислителей.

Выводы. В статье предложен метод синтеза триангулированных поверхностей, основанный на методе обратного трассирования и имеющий следующие отличительные особенности:

- оптимизирован для параллельно-конвейерной архитектуры: алгоритм вычислений линейный с непрерывным потоком входных данных, существенную часть операций он позволяет выполнять параллельно, в

вычислениях используется только целочисленная арифметика, количество матричных преобразований минимально;

- существует взаимосвязь угловой погрешности СВ и геометрических искажений визуализируемой поверхности;

- отсутствует необходимость интерполяции освещенности в плоскости треугольника в соответствии с Гуро или Фонгом;

- возможен синтез изображений динамически изменяемых ТП, так как не требуются дополнительные описания для сцены или объектов (различные классификации, двоичные деревья и пр.);

- с увеличением степени субдискретизации экрана вычислительные затраты уменьшаются по сравнению с классическим подходом.

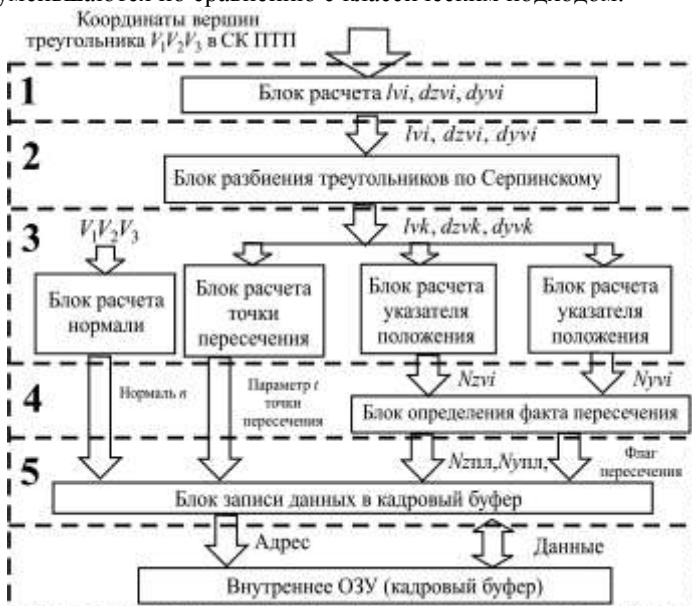


Рис. 3. Структурная схема спецвычислителя

Метод целесообразно использовать для синтеза изображений детализированных или динамически изменяющихся объектов с относительно небольшими линейными размерами. Например, человек, животные, ЛА и прочая техника, которую сложно достаточно детально представить в виде совокупности аналитически заданных поверхностей.

В дальнейшем предполагается исследовать особенности наложения текстуры на предельно триангулированную поверхность и способы устранения её ступенчатости: билинейная фильтрация, анизотропная фильтрация и т.д.

Таким образом, для предельно триангулированной поверхности будут рассмотрены все этапы синтеза изображения. Это позволит сравнить объем вычислений, необходимый для полного синтеза изображения предложенным

методом с существующими подходами, основанными как на методе обратного трассирования лучей, так и прямого.

Список литературы: 1. Computer graphics: principles and practice / J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes. – Addison-Wesley publishing company. – 1996, 2-nd ed. – 1174 p. 2. Realtime Ray Tracing For Current And Future Games / J. Schmittler, D. Pohl, T. Dahmen and etc. – Proceedings of 34. Jahrestagung der Gesellschaft für Informatik. – 2004. 3. *Остроушко А.П.* Модели и структуры вычислительных устройств обработки цветовой информации и создания спецэффектов в системах визуализации: Дис. канд. техн. наук: 05.13.13. – Харьков, 2000. – 138 с. 4. *Чаговец Я.В.* Модели, алгоритмы и структуры спецпроцессоров для формирования изображений рельефа в системах визуализации реального времени: Дис. канд. техн. наук: 05.13.13. – Харьков, 2002. – 209 с. 5. *Филимончук М.А.* Модели, алгоритмы и структуры спецпроцессоров формирования изображений движущихся объектов в системах визуализации реального времени: Дис. канд. техн. наук: 05.13.13. – Харьков, 2004. – 141 с. 6. *Бугрий А.Н.* Модели, алгоритмы и структуры спецпроцессоров формирования изображений световых эффектов в системах визуализации реального времени: Дис. канд. техн. наук: 05.13.13. – Харьков, 2004. – 141 с. 7. Schmittler J. SaarCore A Hardware-Architecture for Realtime Ray Tracing, PhD thesis. Computer Graphics Group, Saarland University, 2006. 8. Fender J., Rose J. A High-Speed Ray Tracing Engine Built on a Field-Programmable System // In IEEE International Conf. on Field-Programmable Technology. – 2003. – P. 188–195. 9. Möller T., Trumbore B. Fast, Minimum Storage Ray Triangle Intersection // The Journal of Graphics Tools, A. K. Peters. – 1997. – Vol 2. – № 1. – P. 21–28. 10. Galin E. Fast Ray Triangle Fan Intersection. Rapport de recherche RR-LIRIS-2004-030, Soumis à Journal of Graphic Tools. – 2004. 11. Badouel D. An Efficient Ray-Polygon Intersection // *Graphics Gems*, Academic Press. – 1990. – P. 390–393. 12. Stolfi J. Oriented Projective Geometry. – Boston: Academic Press. – 1991. – 237 p. 13. *Филимончук М.А., Громенко А.Е.* Уменьшение времени синтеза изображений динамических объектов методом обратного трассирования // Вісник Харківського Національного університету, № 551. Серія: "Актуальні проблеми сучасної науки в дослідженнях молодих вчених м. Харкова". Частина 1. – Харків: ХНУ, 2002. – С. 179–181. 14. Realtime Ray Tracing of Dynamic Scenes on an FPGA Chip. / J. Schmittler, S. Woop, D. Wagner, and etc. // In Proceedings of Graphics Hardware. – 2004. – P. 95–106. 15. *Гусятин В.М.* Оценка точности геометрических преобразований в спецпроцессоре растровой графики // Радиоэлектроника и информатика. – 1998. – № 2. – С. 118–120. 16. Olano T.M., Yoo T.S. Precision Normals (Beyond Phong). UNC Computer Science Technical Report 93-021, Department of Computer Science, University of North Carolina, Chapel Hill, North Carolina, 1993. 17. Curved PN Triangles // A. Vlachos, J. Peters., C. Boyd, J.L. Mitchell // ACM Symposium on Interactive 3D Graphics. – 2001. – P. 159–166. 18. *Гусятин В.М., Громенко А.Е., Сорокин Р.В.* Рекурсивное разбиение триангулированных поверхностей в задачах синтеза изображений для систем визуализации // Материалы VII международной научно-технической конференции "Авиа-2006", 2006. – Т. 1. – К.: НАУ. – С. 13.113–13.116. 19. Zorin D, Schroder P, Sweldens W. Interpolating subdivision for meshes with arbitrary topology // ACM siggraph proceedings. – 1996. – P. 189–192. 20. Labsik U., Greiner G. Interpolatory $\sqrt{3}$ -Subdivision // In Proceedings of Eurographics Computer Graphics Forum, Bd. 19. – 2000. – Vol. 19. – P. 131–38. 21. *Морозов А.Д.* Введение в теорию фракталов: учебное пособие. – Нижний Новгород: Изд-во Нижегородского университета, 2004. – 160 с. 22. *Гуж М.Ю.* Аппаратные средства IBM PC. Энциклопедия. – С-Пб: Питер, 2006. – 1072 с. 23. Schmittler J., Wald I., Slusallek P. SaarCOR – A Hardware Architecture for Ray Tracing // In Proceedings of the ACM SIGGRAPH / Eurographics Conference on Graphics Hardware 2002. – 2002. – P. 27–36.

Поступила в редакцию 30.10.2006