

# Ускорение метода световых сеток за счет использования графического процессора

Виктор Дебелов, Илья Новиков  
Институт вычислительной математики и математической геофизики СО РАН,  
Новосибирск, Россия  
[debelov@oapmg.sgcc.ru](mailto:debelov@oapmg.sgcc.ru), [ilya.novikov@gmail.com](mailto:ilya.novikov@gmail.com)

## Аннотация

Метод световых сеток (МСС) является модификацией классической обратной рекурсивной лучевой трассировки Виттеда (ОРЛТ), использующейся для рендеринга пространственных сцен. Изображение, рассчитанное по МСС, является аппроксимацией изображения, рассчитанного алгоритмом Виттеда, при этом МСС генерирует мягкие тени. Алгоритмы теневых карт (АТК) и теневых объемов (АТО) разработаны для определения теней точечных источников света. Их основная привлекательность – быстрый расчет изображений при использовании операций z-буфер и стенси-буфер графического процессора (ГП).

В работе предложены гибридные алгоритмы МСС-АТК и МСС-АТО, в которых часть теневых расчетов осуществляется на ГП с использованием подходов АТК или АТО, соответственно. Приводятся данные численных экспериментов, которые показывают, что ускорение МСС достигает в среднем 20-34% для МСС-АТК и 5-28% для МСС-АТО в зависимости от типа сцены.

**Ключевые слова:** реалистический рендеринг, трассировка лучей Виттеда, теневые карты, теневые объемы, метод световых сеток, мягкие тени.

## 1. ВВЕДЕНИЕ

МСС является модификацией трассировки Виттеда [1], использующейся для рендеринга пространственных сцен. Первая версия МСС была введена в работах [2, 3]. В работах [4, 5] сформировался его современный вид, который рассматривается в данной работе. В работе [5] дается обширное исследование МСС и его сравнение с известными алгоритмами генерации мягких теней в рамках ОРЛТ как по скорости, так и по качеству получаемых изображений.

АТК [6] и АТО [7] – два алгоритма генерации четких теней, получившие широкое распространение благодаря аппаратной поддержке расчета изображения. Современные ГП позволяют рассчитывать изображения для сцен, состоящих из сотен тысяч и миллионов треугольников, меньше чем за секунду. С другой стороны, у этих алгоритмов много недостатков: они страдают от артефактов, не применяются для рекурсивных расчетов, не работают для полупрозрачных объектов.

В данной работе ставится и решается задача ускорения МСС за счет применения ГП, используя подходы упомянутых алгоритмов. Предполагается, что читатель в основных чертах знаком с алгоритмами ОРЛТ, АТК, АТО.

## 1.1 Обратная рекурсивная лучевая трассировка

Пространственная сцена – это кусочно-непрерывная поверхность, представленная набором из  $nO$  объектов или примитивов, обладающих следующими свойствами:

- В каждой точке поверхности определены свойства отражения, которые задаются коэффициентами:  $k_d$  – коэффициент диффузного отражения,  $k_s$  – коэффициент зеркального отражения,  $k_t$  – прозрачность.
- Для поверхности определена операция пересечения с лучом.
- В каждой точке поверхности  $P$  определена нормаль  $\vec{n}(P)$ .
- Все объекты сцены задаются в декартовой мировой системе координат.

Сцена освещается  $nL$  точечными источниками освещения, специфицированными интенсивностями излучения  $I_i$  и

позициями в пространстве  $LP_i$ ,  $i=1, \dots, nL$ . Камера – это набор параметров, характеризующих наблюдателя: позиция и ориентация в пространстве, высота и ширина экрана.

Пусть луч из камеры имеет ближайшее пересечение с поверхностью сцены в точке  $P$ . Базовую формулу расчета значения интенсивности в точке  $P$  по ОРЛТ можно кратко записать, как (следуя [8]):

$$I_{RT}(P) = A + \sum_{i=1}^{nL} V(LP_i, P) \Omega(L_i, P) + k_s I_r + k_t I_t \quad (1)$$
$$= A + U_{RT}(P) + k_s I_r + k_t I_t,$$

здесь  $A$  – интенсивность рассеянного света,  $\Omega(L_i, P)$  характеризует конкретную локальную модель освещенности,  $I_r$  – интенсивность, пришедшая с направления отраженного вектора;  $I_t$  – интенсивность, пришедшая через поверхность из-за прозрачности.  $V(LP_i, P)$  – булева функция видимости  $i$ -го точечного источника из точки  $P$ .

## 1.2 Алгоритм теневых карт

Буфер глубины  $DB[N, M]$  – прямоугольник на плоскости изображения, совпадающий с прямоугольником изображения, разрешение буфера глубины  $N \times M$  пикселей. Пиксель буфера глубины хранит расстояние от камеры до ближайшего пересечения луча из камеры, проходящего через центр этого пикселя, с объектами сцены.

В АТК для определения видимости изображаемой точки источником используются теневые карты. Теневая карта – это буфер глубины, рассчитанный для камеры, находящейся в

позиции источника освещения. Чтобы определить видимость изображаемой точки, расстояние от точки до источника сравнивается со значением глубины из теневой карты. Если значение из теневой карты меньше, источник не виден – изображаемая точка находится в тени источника. Подробное исследование АТК можно найти в работах [5, 6].

### 1.3 Алгоритм теневых объемов

АТО для определения видимости изображаемой точки использует дополнительные геометрические построения, называемые теневыми объемами. Теневые объемы – это полубесконечные поверхности, получаемые вытягиванием геометрии силуэтов объектов от источника на бесконечность. Если изображаемая точка попадает внутрь хотя бы одного теневого объема, значит она в тени источника.

Стенсил-буфер  $SB[N, M]$  – еще один буфер на прямоугольнике изображения. Каждый пиксель стенсил-буфера можно интерпретировать как счетчик, который увеличивается, когда луч из камеры через центр этого пикселя входит в теневой объем, и уменьшается, когда луч выходит из теневого объема. Более подробно см. [5, 7].

### 1.4 Метод световых сеток

Световая сетка  $LM = \{x_k\}_{k=1}^{N^{LM}}$  – это равномерная сетка с шагом  $h$  и размером  $N^{LM} = N_x^{LM} \times N_y^{LM} \times N_z^{LM}$  в пространстве сцены, каждая точка световой сетки  $x_k$  (световая точка) хранит шкалу видимости данной точки пространства сцены для каждого источника. Для расчета видимости в точке сцены  $P$  строится интерполяционное множество  $D_{LM}(P, r)$  по следующему правилу

$$D_{LM}(P, r) = \{x : x \in LM, \|P - x\| < r, (\vec{n}(P), x - P) > 0, V(P, x) = 1\} \quad (2)$$

Аналогично формуле (1) базовую формулу расчета значения интенсивности в точке  $P$  по МСС представим как

$$U_{LM}(P) = \sum_{i=1}^{m_L} \left[ \frac{1}{m} \sum_{k=1}^m V(LP_i, x_k) \chi(P, L_i) \right] \Omega(L_i, P), \quad (3)$$

$$I_{LM}(P) = A + U_{LM}(P) + k_s I_r + k_t I_t$$

здесь  $\chi(P, L_i) = \begin{cases} 1, & (\vec{n}(P), (LP_i - P)) \geq 0 \\ 0, & \text{иначе} \end{cases}$  – функция, которая

говорит о том, освещает источник точку спереди (значение 1) или сзади (значение 0).  $m$  – число световых точек в интерполяционном множестве. Если  $m = 0$ , тогда значение интенсивности в точке  $P$  по МСС рассчитывается по формуле (1), т.е. в этом случае  $U_{LM}(P) = U_{RT}(P)$ .

В формуле (3) используется функция значения видимости источника световой точкой  $V(LP_i, x_k)$ , которая требует значительных вычислительных затрат. Для уменьшения этих затрат рассматриваются два алгоритма: 1) комплексирование МСС с АТК; 2) комплексирование МСС с АТО. Основная идея: заимствовать для МСС положительное качество АТК и АТО – быстрый тест видимости источника.

## 2. КОМПЛЕКСИРОВАНИЕ МСС С АТК

Рассмотрим этот алгоритм по шагам для сцены с одним источником  $L$ : интенсивность  $I$ , позиция в сцене  $LP$ .

1. Строим световую сетку  $LM = \{x_k\}_{k=1}^{N^{LM}}$ .
2. Строим камеру зрителя  $Cam$ . Оператор перевода координат точки сцены из системы камеры в мировую систему координат  $Tr_{Cam}: P(u, v, w) \Rightarrow P(x, y, z)$ , т.е.  $Tr_{Cam}(u, v, w) = (x, y, z)$ . Порт вывода  $\Pi_{Cam}$ . Рассчитываем буфер глубины  $DB_{Cam}$ .
3. Для источника  $L$  строим камеру  $LCam$ . Порт вывода  $\Pi_L$ . Оператор перевода координат  $Tr_L$ . Строим теневую карту  $DB_L$ .
4. Строим изображение сцены  $FB_{Cam}$  в камере  $Cam$ , опираясь на обе карты глубины. Если глубина изображаемой точки  $w = DB_{Cam}(u, v) = -1$ , полагаем  $FB_{Cam}(u, v) = -1$ , т.е. цвет фона. Выход.
5. Иначе вычисляем  $P = (x, y, z) = Tr_{Cam}(u, v, w)$  – координаты изображаемой точки в мировой системе.
6. Для точки  $P$  строим интерполяционное множество  $D_{LM}(P, r)$  по правилу (2), размер множества  $m$ .
7. Если  $m > 0$ , т.е.  $D_{LM}(P, r) \neq \emptyset$ , определяем видимость источника  $V(LP, x_k)$  для каждой световой точки  $x_k \in D_{LM}(P, r) = \{x_k\}_{k=1}^m$ :
  - 7.1. Если видимость для световой точки уже известна, т.е. была вычислена ранее, переходим к следующей световой точке интерполяционного множества.
  - 7.2. Иначе вычисляем расстояние  $d = \|x_k - LP\|$ .
  - 7.3. Переводим координаты световой точки  $x_k = (x_k^x, x_k^y, x_k^z)$  в систему координат порта вывода камеры источника  $LCam$ :  $(uL, vL, wL) = Tr_L^{-1}(x_k^x, x_k^y, x_k^z)$ .
  - 7.4. Выбираем из карты глубины камеры источника значение  $d_L = DB_L(uL, vL)$ .
  - 7.5. Если  $d_L < d$ , то источник заслонен от световой точки  $x_k$ , т.е.  $V(LP, x_k) = 0$ . Иначе световая точка освещена источником, т.е.  $V(LP, x_k) = 1$ .
8. Рассчитываем интенсивность изображаемой точки  $P$  по формуле (3) и добавляем в  $FB_{Cam}(u, v)$ .
9. Если  $m = 0$ , т.е.  $D_{LM}(P, r) = \emptyset$ , используем стандартный ОРЛТ: рассчитываем интенсивность точки  $P$  по формуле (1) и добавляем в  $FB_{Cam}(u, v)$ .

Шаги 2 и 3 выполняются на ГП. Остальные шаги выполняются на центральном процессоре.

## 3. КОМПЛЕКСИРОВАНИЕ МСС С АТО

По шагам алгоритм можно записать следующим образом:

1. Рассмотрим световую сетку  $LM$  размера  $N_x^{LM} \times N_y^{LM} \times N_z^{LM}$ , как набор прямоугольных слоев световых точек вдоль одной из координатных осей. Пусть это будет ось  $Z$ , тогда каждый слой имеет размеры  $N_x^{LM} \times N_y^{LM}$ .

2. Введем новый порт вывода  $\Pi_{LM}$ , на котором построим растр размером  $N_x^{LM} \times N_y^{LM}$ . Вводим буфер глубины  $DB_{LM}$  и стенил-буфер  $SB_{LM}$  на  $\Pi_{LM}$ .
3. Для каждого слоя световых точек  $LM_z^k$  строим камеру  $CamLM$  так, что луч из верхушки  $O_{LM}$  камеры в каждую световую точку  $LM_z^k[i, j]$  проходит через центр пикселя  $[i, j]$  прямоугольника  $\Pi_{LM}$  (рис. 1).
4. Рассчитываем буфер глубины  $DB_{LM}$  в камере  $CamLM$  для сцены, состоящей из световых точек  $LM_z^k$ .
5. Строим теневой объем сцены  $SV(L)$  относительно источника  $L$ .
6. В точках  $(u, v) \in \Pi_{LM}$ , где  $DB_{LM}(u, v) = -1$  полагаем  $SB_{LM}(u, v) = -1$ . Иначе  $SB_{LM}(u, v) = c$ , где  $c$  – некоторый счетчик, вычисляемый по алгоритму:
  - 6.1. Находим точку  $P(u, v, w)$  в системе координат камеры, где  $w = DB_{LM}(u, v) \neq -1$  (не фон), переводим ее координаты в мировую систему  $(x, y, z) = Tr_{LM}(u, v, w)$ . Пускаем луч  $\overrightarrow{O_{LM}P}$  в пространстве сцены и подсчитываем число пересечений этого луча с поверхностями из теневого объема  $SV(L)$ . Вначале  $c = 0$ .
  - 6.2. После нахождения очередного пересечения луча с тенью объемом в некоторой точке  $Q$  проверяем по карте глубины: если  $\|\overrightarrow{O_{LM}Q}\| > DB_{LM}(u, v) = \|\overrightarrow{O_{LM}P}\|$ , то переходим к поиску следующего пересечения, поскольку найденная точка не может заслонять тестируемую точку  $P$ , она дальше от камеры.
  - 6.3. Иначе анализируем скалярное произведение  $t = (\overrightarrow{O_{LM}P}, \vec{n}(Q))$ .
  - 6.4. Если  $t = 0$ , то счетчик не изменяется. Если  $t < 0$ , то увеличиваем счетчик  $c = c + 1$ . Это означает, что луч вошел в очередной тенью объем очередного примитива.
  - 6.5. Если  $t > 0$ , то уменьшаем счетчик  $c = c - 1$ . Это означает, что луч вышел из очередного теневого объема очередного примитива.
7. Заполняем значения видимости для световых точек слоя  $LM_z^k$ . Если  $SB_{LM}[i, j] > 0$ , эта световая точка в тени источника  $L$ , т.е.  $V(LP, LM_z^k[i, j]) = 0$ . Иначе световая точка освещена источником, т.е.  $V(LP, LM_z^k[i, j]) = 1$ .
8. Выполняем шаги алгоритма 3 – 7 для каждого слоя  $LM_z^k$ ,  $k = 1 \dots N_z^{LM}$ . Получаем заполненную световую сетку  $LM$ .
9. Строим камеру зрителя  $Cam$ . Оператор перевода координат  $Tr_{Cam}$ . Порт вывода  $\Pi_{Cam}$ . Рассчитываем буфер глубины  $DB_{Cam}$ .

10. Строим изображение сцены  $FB_{Cam}$  в камере  $Cam$ . Если глубина изображаемой точки  $w = DB_{Cam}(u, v) = -1$ , тогда  $FB_{Cam}(u, v) = -1$ , т.е. цвет фона. Выход.
11. Иначе вычисляем  $P = (x, y, z) = Tr_{Cam}(u, v, w)$  – координаты изображаемой точки в мировой системе.
12. Для точки  $P$  строим интерполяционное множество  $D_{LM}(P, r)$  по правилу (2), размер множества  $m$ .
13. Если  $m > 0$ , рассчитываем интенсивность изображаемой точки  $P$  по формуле (3) и добавляем в  $FB_{Cam}(u, v)$ .
14. Если  $m = 0$ , используем стандартный ОРЛТ: рассчитываем интенсивность точки  $P$  по формуле (1) и добавляем в  $FB_{Cam}(u, v)$ .

Шаги 3 – 9 выполняются на ГП, остальные шаги выполняются на центральном процессоре.

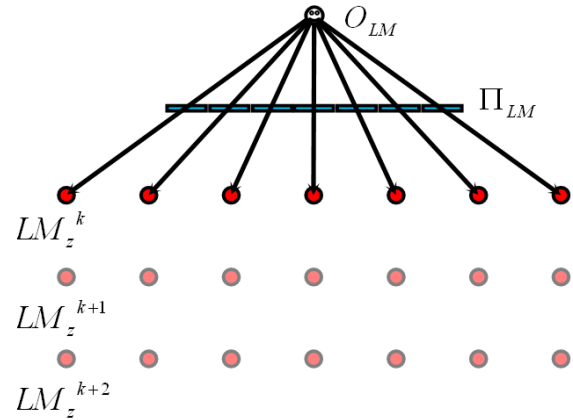


Рис. 1. Камера  $CamLM$ , прямоугольник  $\Pi_{LM}$ , слой световых точек

#### 4. ЧИСЛЕННЫЕ ЭКСПЕРИМЕНТЫ

В табл. 1 – 3 представлены времена расчета для сцен: “Gazebo” – 300 треугольников, “Child park” – 27000 треугольников, “Garden” – 234000 треугольников. В ходе расчетов варьировались: разрешение изображения  $N_{Cam} \times M_{Cam}$ , шаг световой сетки  $h$  и радиус интерполяционной сферы  $r$ . Представлены времена расчета изображений по базовому МСС, гибричному алгоритму МСС-АТК, гибричному алгоритму МСС-АТО, как и обозначены соответствующие столбцы. Для сцены “Gazebo” комплексирование МСС с АТК позволяет сократить время расчета в среднем на 20%, комплексирование МСС с АТО – на 5%.

Таблица 1. Времена расчета сцены “Gazebo” (с)

$N_{Cam} \times M_{Cam}$	$h$	$r/h$	МСС	МСС-АТК	МСС-АТО
1024 x 1024	2.	2.1	13,2	11,0	12,3
	0	3.1	20,5	16,7	18,0
	1.	2.1	18,4	11,6	19,8
	0	3.1	28,9	17,7	25,9
2048 x 2048	2.	2.1	45,4	41,9	43,9
	0	3.1	69,9	63,9	65,5
	1.	2.1	51,1	43,0	51,6
	0	3.1	79,1	66,1	74,3

Для сцены “Child park” комплексирование МСС с АТК позволяет сократить время расчета в среднем на 21%, комплексирование МСС с АТО – на 11%.

**Таблица 2.** Времена расчета сцены “Child park” (с)

$N_{Cam} \times M_{Cam}$	$h$	$r/h$	МСС	МСС-АТК	МСС-АТО
1024 x 1024	2.	2.1	27,0	22,3	24,0
		0	3,1	34,0	26,7
	1.	2.1	37,0	23,0	31,0
		0	3,1	50,8	28,0
2048 x 2048	2.	2.1	90,6	84,4	96,6
		0	3,1	109,6	101,0
	1.	2.1	101,8	87,0	94,1
		0	3,1	127,5	103,5

Для сцены “Garden” комплексирование МСС с АТК позволяет сократить время расчета в среднем на 34%, комплексирование МСС с АТО – на 28%.

**Таблица 3.** Времена расчета сцены “Garden” (с)

$N_{Cam} \times M_{Cam}$	$h$	$r/h$	МСС	МСС-АТК	МСС-АТО
1024 x 1024	2.	2.1	24,1	15,3	16,9
		0	3,1	31,4	21,0
	1.	2.1	37,2	16,4	21,3
		0	3,1	55,3	22,8
2048 x 2048	2.	2.1	65,4	56,2	58,1
		0	3,1	89,0	76,8
	1.	2.1	81,4	57,8	63,2
		0	3,1	115,0	79,2

## 5. ЗАКЛЮЧЕНИЕ

Гибрид МСС-АТК позволяет рассчитывать изображения в среднем на 20 – 34 % (в зависимости от типа сцены) быстрее в сравнении с базовым МСС. Для гибрида МСС-АТО среднее ускорение в сравнении с базовым МСС составляет 5 – 28 % в зависимости от типа сцены. С увеличением сложности сцены гибридные алгоритмы дают больший выигрыш по времени в сравнении с базовым МСС.

## Благодарности

Работа выполнялась при финансовой поддержке РФФИ по грантам 06-07-89216\_а, 08-07-12094\_офи.

## 6. REFERENCES

- [1] Whitted T. An Improved Illumination Model for Shaded Display / Commun. ACM. – 1980. – Vol. 23, № 6. – P. 343-349.
- [2] Дебелов В.А., Севастьянов И.М. *Оригинальный подход к имитации мягких теней и учету диффузных переотражений в лучевой трассировке // Тр. 11-й междунар. конфер. по компьютерной графике и машинному зрению Графикон'2001. – Нижний Новгород, 2001. – С. 18–24. 21.*
- [3] Debelov V.A., Sevastyanov I.M. *Soft shadows as interpolation of visibility // Future Generation Computer Systems. - 2004. - Vol. 20, № 8. - P. 1299-1315.*
- [4] Дебелов В.А., Васильева Л.Ф., Новиков И.Е. *Развитие метода световых сеток для алгоритма лучевой трассировки: аппроксимация решения, реализация на графическом акселераторе // Тр. 15-й междунар. конфер. по*

*компьютерной графике и ее приложениям Графикон'2005. – Новосибирск, 2005. – С. 355–359.*

- [5] Дебелов В.А., Новиков И.Е. *Генерация мягких теней при использовании алгоритма трассировки лучей // Вестник НГУ. Серия: Информационные Технологии. – 2009. – Т. 7, № 2. – С. 18–41.*
- [6] Williams L., *Casting curved shadows on curved surfaces // Computer Graphics. – 1978. – Vol. 10, № 2. – P. 270-274.*
- [7] Crow F. *Shadow Algorithms for Computer Graphics // Computer Graphics. – 1977. – Vol. 11, № 2. – P. 242-247.*
- [8] Foley J., Van Dam A., a.o. *Computer Graphics Principles and Practice / 2nd Edition. – Addison Wesley. – 1990.*

## Об авторах

**Дебелов Виктор Алексеевич** – доктор технических наук, ведущий научный сотрудник лаборатории численного анализа и машинной графики Института вычислительной математики и математической геофизики СО РАН, профессор Новосибирского государственного университета.

E-mail: [debelov@oapmg.sccc.ru](mailto:debelov@oapmg.sccc.ru).

**Новиков Илья Евгеньевич** – аспирант Института вычислительной математики и математической геофизики СО РАН.

E-mail: [iliya.novikov@gmail.com](mailto:iliya.novikov@gmail.com).

# Light Meshes Method Acceleration Using GPU

## Abstract

This work is devoted to Light meshes method (LMM) which is a modification of Whitted's ray tracing algorithm. LMM allows simulating soft shadows from point light sources that increases realism of synthesized images, but requires additional computational costs. Shadow maps (SM) and Shadow volumes (SV) are two popular algorithms for hard shadows generation. The main advantage of SM and SV is hardware acceleration by GPU. The idea of this work is to use fast light visibility detection by SM and SV for LMM, i.e. to construct hybrid algorithms based on LMM that fulfill visibility tests on GPU giving us an image generation speed-up while the same image quality.

**Keywords:** *photorealistic rendering, ray tracing, shadow map, shadow volume, light meshes method, soft shadows.*

## About the authors

**Victor A. Debelov** is a doctor of technical sciences and holds position of leading researcher at Computer Graphics Lab of the Institute of Computational Mathematics and Mathematical Geophysics (<http://www.sccc.ru>). Also he is a professor at Novosibirsk State University.

E-mail: [debelov@oapmg.sccc.ru](mailto:debelov@oapmg.sccc.ru).

**Iliya Novikov** is a Ph.D. student of the Institute of Computational Mathematics and Mathematical Geophysics (<http://www.sccc.ru>).

E-mail: [iliya.novikov@gmail.com](mailto:iliya.novikov@gmail.com).