

# Матричные преобразования для эффективной реализации алгоритма излучательности на графических процессорах

А.С. ЩЕРБАКОВ<sup>1</sup>, В.А. ФРОЛОВ<sup>1, 2</sup>

<sup>1</sup>Московский государственный университет им. М.В. Ломоносова, <sup>2</sup>Институт прикладной математики им. М.В. Келдыша РАН, Москва  
E-mail: alex.shcherbakov@graphics.cs.msu.ru, vladimir.frolov@graphics.cs.msu.ru

## Аннотация

Предложен метод преобразования матрицы форм-факторов, позволяющий ускорить расчёт вторичного освещения методом излучательности. Рассмотрена адаптация этого метода для графических процессоров (*graphics processing unit, GPU*). В частности, предложено использовать *DXT*-текстуры для хранения матрицы форм-факторов и переупорядочивать столбцы и строки матрицы для уменьшения потерь при сжатии. Предложенные оптимизации повышают скорость работы алгоритма излучательности до 10 раз и уменьшают до 3 раз объём занимаемой памяти *GPU*.

**Ключевые слова:** излучательность, глобальное освещение, *GPU*.

## 1. Введение

Глобальное освещение 3D-сцен складывается из первичного освещения источниками света (ИС) и вторично-многократно отражённым от поверхностей сцены светом. Наибольшую сложность представляет расчёт вторичного освещения, так как размерность интеграла освещённости увеличивается с каждым отражением. Поэтому в приложениях реального времени используют ряд методов приближённого расчёта.

## 2. Обзор существующих методов

• **Метод «Instant Radiosity»** [1, 2]. Является одним из самых популярных благодаря своей простоте. Для расчёта вторичного освещения используются «вторичные» ИС, создаваемые путём трассировки лучей из первичных ИС. Вторичное освещение, таким образом, может рассматриваться как первичное от вторичных ИС. Развитием метода «Instant Radiosity» для *GPU* служит алгоритм «Reflective Shadow Maps» («RSM») [3]. Вместо трассировки лу-

чей в «RSM» для создания вторичных ИС используются карты теней (*shadow maps*). Основным недостатком данного метода – низкая точность.

• **Метод «Light Propagation Volumes»** [4]. Создаёт вторичные ИС так же, как это делает «Instance Radiosity», но расчёт вторичного освещения производится посредством распространения света по трёхмерной сетке. Основным недостатком данного метода – высокий расход памяти и низкая эффективность при расчётах по распространению света через пустые пространства.

• **Метод «Voxel Cone Tracing»** [5]. Производит «сбор освещения» для каждого пикселя путём трассировки нескольких конусов из заданной точки на поверхности, имитируя Монте-Карло-трассировку лучей по полусфере. В этот метод (так же как в предыдущем) используется воксельная сетка для представления упрощённой геометрии, а сама трассировка конусов аналогична шаганию по лучу (*ray marching*). Отличие в том, что с увеличением расстояния выборка производится из более грубых мип-уровней воксельной сетки, за счёт чего и получается геометрическая аппроксимация конуса. Недостатком алгоритма является высокая вычислительная сложность и зависимость скорости вычисления от разрешения.

• **Метод сферических гармоник** [6]. Основывается на разложении сложных функций освещённости в сумму более простых для вычисления величин. Для некоторых точек поверхности (как правило, вершин) вычисляются коэффициенты разложения их функций освещения по базису. Функции освещения из ИС также раскладываются по базису. В итоге вычисление освещения в точке с разложенной в ней функцией освещённости сводится к скалярному произведению векторов, состоящих из коэффициентов функции освещённости в данной

точке и функции освещения из ИС. Данный метод широко используется при визуализации открытых пространств, но уступает в точности для закрытых помещений методу излучательности.

• **Метод излучательности** [7]. Позволяет получать качественные изображения для закрытых помещений с диффузными поверхностями, во многом не уступая более современным методам. Однако, время выполнения и требуемые ресурсы очень сильно зависят от сцены. Для сцен, содержащих сотни тысяч треугольников, прямое применение излучательности затруднено из-за квадратичной сложности и затрат памяти в зависимости от количества примитивов. Поэтому на практике алгоритм излучательности выполняется для упрощённой сцены (содержащей меньшее количество площадок) и результат расчёта переносится на исходную сцену [8]. Следует подчеркнуть, что наряду со сферическими гармониками алгоритм излучательности переносит основную вычислительную сложность на этап предрасчёта, за счёт чего и достигается хороший баланс точности и скорости по сравнению с остальными методами.

## 3. Ключевые термины и определения.

В классическом алгоритме излучательности используется матрица форм-факторов  $F$  размера  $n \times n$ , где  $n$  – число площадок сцены. Для вычисления освещённости  $m_i$ , после отражения данная матрица умножается на  $n$ -компонентный вектор начальной светимости  $m_e^{(0)}$ , содержащий светимость площадок:

$$\bar{m}_i^{(1)} = F \cdot \bar{m}_e^{(0)}$$

Умножая полученный вектор на коэффициент отражения площадок  $\rho$ , на которые пришёл свет, вычисляют светимость площадок после отражения:

$$\bar{m}_e^{(1)} = \bar{m}_i^{(1)} \cdot \rho. \quad (1)$$

Элементами векторов в данных формулах являются 3-компонентные векторы, содержащие информацию по каждому цветовому каналу, а элементами матрицы форм-факторов – вещественные числа от 0 до 1. Приведённые вычисления можно повто-

рять, используя векторы  $m_e^{(n)}$  вместо изначальной светимости площадок, для получения света, пришедшего на поверхности сцены, после произвольного отражения:

$$\bar{m}_i^{(n)} = F \cdot \bar{m}_e^{(n-1)}. \quad (2)$$

Полное освещение сцены после  $k$  отражений получается путём суммирования векторов  $m_i^{(n)}$ :

$$\bar{I} = \sum_{n=1}^k \bar{m}_i^{(n)}. \quad (3)$$

#### 4. Предлагаемый метод

##### 4.1. Предрасчёт нескольких отражений сцены

Предлагаемая модификация заключается в использовании преобразованной матрицы форм-факторов. В начале введём «цветную» матрицу форм-факторов:

$$F_{ij}^C = F_{ij} \cdot \rho_j,$$

где  $\rho_j$  – цвет  $j$ -ой площадки. Эта матрица содержит информацию о переносе света между площадками сцены по каждому каналу в отдельности. Таким образом, для её хранения требуется втрое больше памяти. В данной работе рассматривается использование алгоритма излучательности для вторичного освещения сцены. Поэтому вычисление излучательности начинается не с получения вектора  $m_i^{(1)}$ , а с вычисления светимости после первого отражения  $m_e^{(1)}$ . Можно считать, что вектор  $m_e^{(1)}$  уже посчитан. Мы можем преобразовать выражение (2), используя формулу (1) для произвольного индекса.

$$\begin{aligned} m_i^{(n)} &= F \cdot (\rho \cdot m_i^{(n-1)}) = \\ &= F^C \cdot m_i^{(n-1)} = (F^C)^{-1} \cdot m_i^{(n)}. \end{aligned} \quad (4)$$

Преобразуем (3) с использованием (4):

$$\bar{I} = \left( \sum_{n=1}^k (F^C)^{n-1} \right) \cdot m_i^{(n)}.$$

Матричный полином в скобках не зависит от первичного освещения сцены,

Рис. 1. Распределение значений в матрице форм-факторов (логарифмическая и линейная шкалы)

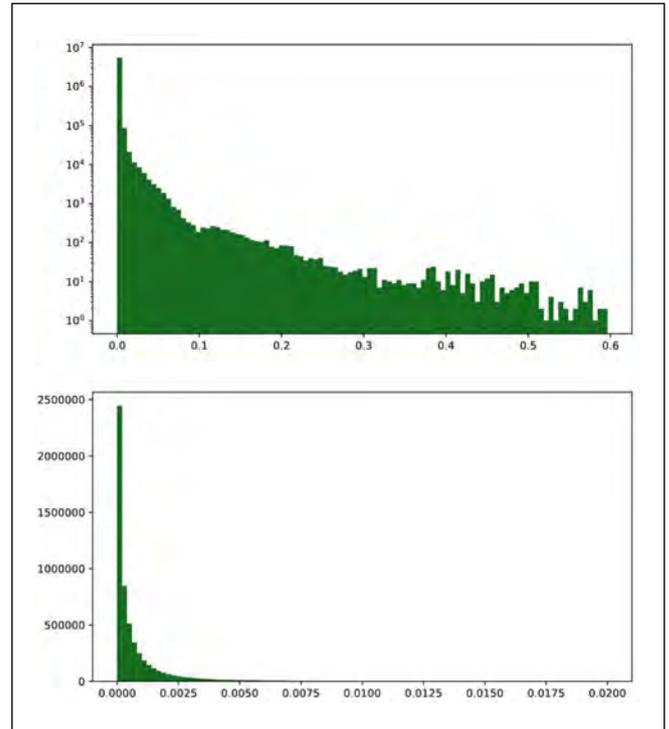


Рис. 2. Фрагмент матрицы форм-факторов до (слева) и после (справа) сортировки строк и столбцов. Увеличенный фрагмент матрицы форм-факторов

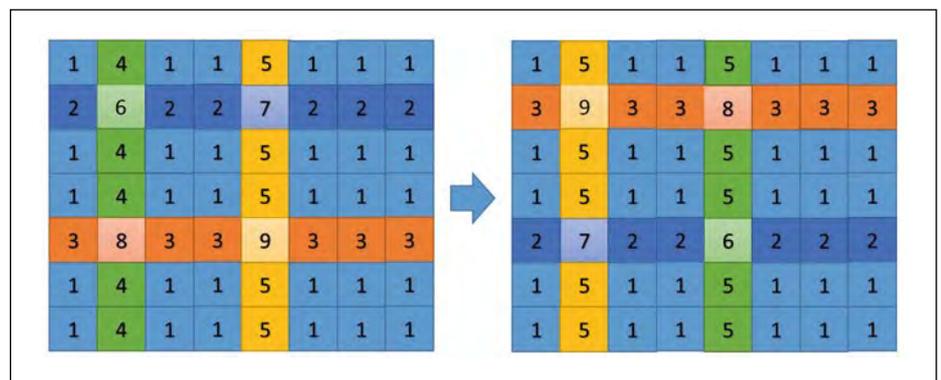
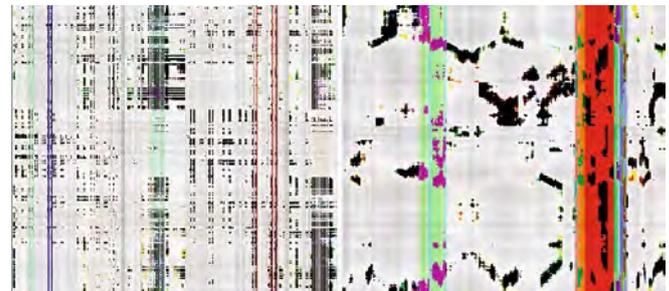


Рис. 3. Схема перестановки строк и столбцов, используемая авторами при сортировке

ности, а зависит лишь от геометрии сцены и материалов поверхности. Поэтому он может быть вычислен на этапе предрасчёта:

$$S = \left( \sum_{i=1}^k (F^C)^{i-1} \right).$$

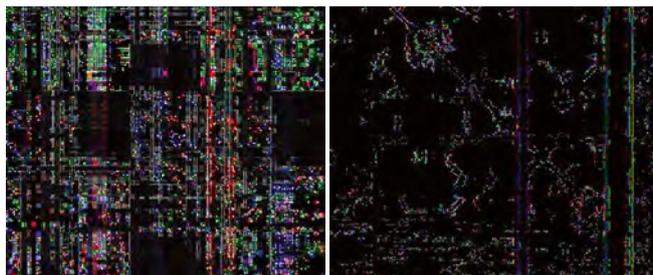
Таким образом, вычисление вторичного освещения методом излуча-

тельности после  $k$  отражений сводится к одному умножению вектора на матрицу:

$$m_i^{(n)} = S \cdot m_i^{(1)}.$$

Использование матрицы такого вида позволяет ускорять вычисления в  $k$  раз (где  $k$  – число переотражений), однако это требует втрое большей памяти.

Рис. 4. Визуализированная разница между сжатой и несжатой текстурами (увеличенный фрагмент) до (слева) и после (справа) переупорядочивания строк и столбцов



ния в матрице необходимо приводить к диапазону 0–255. На рис. 1 показано распределение значений в матрице форм-факторов. При масштабировании этих чисел в диапазоне 0–255 большая часть чисел обнуляется. Поэтому большие числа, которые вносят основной вклад в вычисление излучательности сохраняются отдельно от остальной матрицы, а к остальным применяется следующее преобразование:

$$value'_i = \frac{\max \left( \log \left( \frac{value_i}{\max_j value_j} \right) + shift, 0 \right)}{shift} \cdot 255.$$

Текстура, которая получается в результате такого преобразования, показана на рис. 2, слева.

Однако *DXT*-сжатие допускает потери. Чтобы минимизировать их при сжатии строки и столбцы матрицы сортируются так, чтобы уменьшить разницу между соседними значениями (рис. 2, справа). Так как матрица является отношением для пар площадок сцены, то строки и столбцы должны меняться местами одновременно (рис. 3). В результате среднеквадратичная ошибка при сжатии уменьшается до 5 раз (рис. 4).

### 4.3. Детали реализации

Так как современные *3D*-сцены содержат сотни тысяч треугольников и время выполнения алгоритма излучательности является неприемлемым при таком количестве площадок в сцене, для вычисления вторичного освещения была выбрана упрощённая версия той же самой сцены, используя метод упрощения на основе воксельной сетки из работы [9].

Таким образом, предобработка сцены осуществляется по следующей схеме:

1. Строится упрощённый аналог сцены на основе вокселизации.
2. Вычисляются форм-факторы для площадок упрощённой сцены.
3. Вычисляется матрица форм-факторов, учитывающая несколько отражений света.
4. Числа матрицы больше порогового значения сохраняются в отдельном файле, а на их место ставится 0.

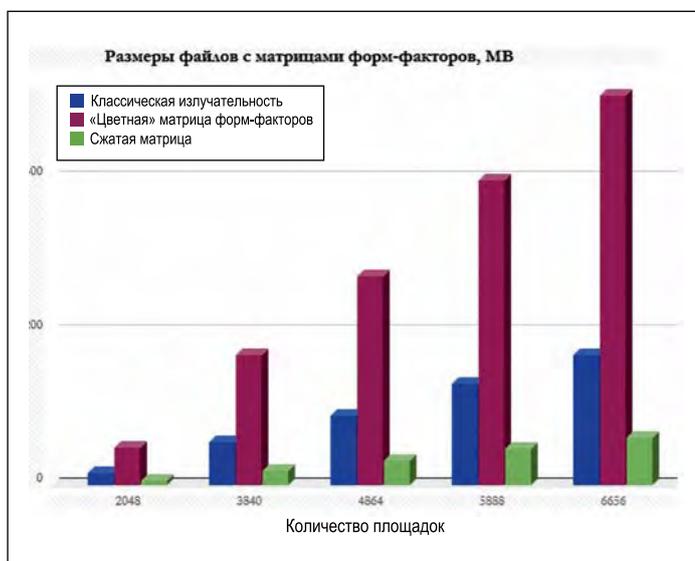


Рис. 5. Сравнение размеров файлов с матрицами форм-факторов, МВ

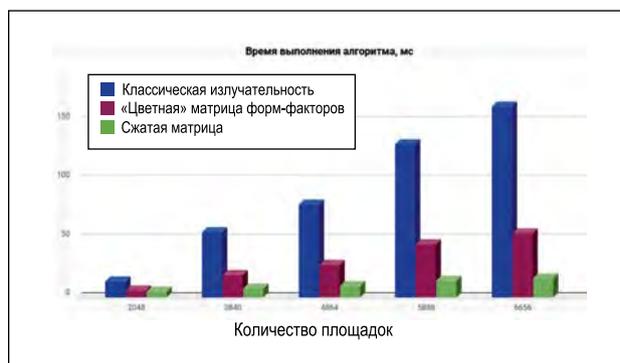


Рис. 6. Сравнение скоростей вычисления излучательности. (Сжатие увеличивает скорость, т.к. алгоритм ограничен доступом к памяти, а не вычислениями.)

Для учёта всех возможных отражений света на сцене необходимо вычислять бесконечный ряд

$$S = \left( \sum_{i=0}^{\infty} (F^C)^i \right).$$

Переход к пределу. Так как исходная матрица положительно определена, все числа в ней меньше 1 и, более того, сумма чисел в одной строке меньше 1, то к этому ряду применима формула для вычисления суммы геометрической прогрессии

$$S = F^C \cdot (I - F^C)^{-1}.$$

Однако вычисление обратной матрицы возможно не всегда из-за неустойчивости решения, в то время как вычисление частичной суммы ряда не обладает подобным недостатком и позволяет учитывать большое количество отражений на этапе прерасчёта.

### 4.2. *DXT*-сжатие

Так как предлагаемая модификация алгоритма требует большего объёма памяти, был разработан метод хранения матрицы форм-факторов в сжатом виде. В качестве формата хранения был выбран формат *DXT*-текстур, поддерживаемый многими *GPU* (имеется ввиду аппаратная декомпрессия при чтении). Для этого формата все значе-

Рис. 7. Изображения, полученные методами «Light Propagation Volumes» (а) и предлагаемым (б), а также – трассировкой путей /эталон/ (в). «Light Propagation Volumes» и предлагаемый метод работают с частотой 40 кадров/с (период 25 мс), а эталон получен за 5 мин

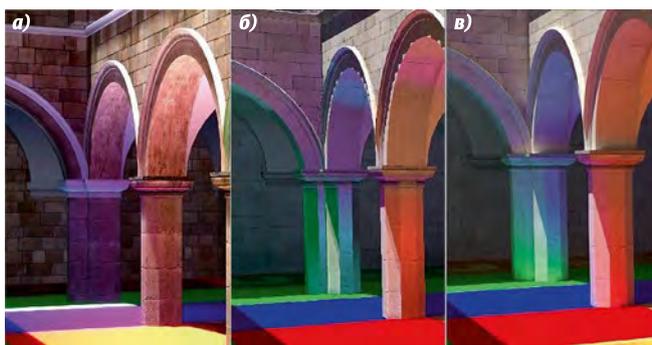


Рис. 8. Изображение, полученное методом «Light Propagation Volumes» («Unreal Engine 4»)



Рис. 9. Изображение, полученное предлагаемым методом



Рис. 10. Изображение, полученное трассировкой путей (в качестве эталона)



5. Значения в матрице приводятся к диапазону 0–255.

6. Осуществляется перестановка строк и столбцов.

7. Полученная матрица сохраняется в виде DXT-текстуры.

Визуализация происходит по следующему алгоритму:

1. Создаётся карта теней.

2. Рассчитывается освещение площадок упрощённой сцены, производимое ИС.

3. Рассчитывается вторичное освещение методом излучательности с помощью матрицы форм-факторов, сохранённой в виде текстуры.

4. Вторичное освещение переносится с упрощённой сцены на исходную.

#### 4.4. Сравнение результатов

Предложенные модификации алгоритма позволяют ускорять алгоритм излучательности в сумме до 10 раз (рис. 5–7). Использование матрицы форм-факторов, учитывающей несколько отражений, увеличивает размер файла с матрицей в 3 раза, однако использование DXT-сжатия позволяет уменьшать требуемую память до 3 раз по сравнению с изначальной матрицей форм-факторов (и классическим алгоритмом излучательности (рис. 5)). Мы провели сравнение с изображениями, полученными методами «Light Propagation Volumes» из «Unreal Engine 4», классической излучательности и трассировки путей (эталон). Предложенный метод демонстрирует результат, сравнимый по точности с методом классической излучательности. При этом он ближе к эталону, чем изображение, получаемое методом «Light Propagation Volumes» при одинаковой частоте кадров (рис. 7–10).

#### 5. Заключение

В отличие от других распространённых методов решения уравнения излучательности предложенный метод позволяет вычислять глобальное освещение за  $n^2 + O(n)$  арифметических операций и чтений из памяти (где, напомним,  $n$  – число площадок сцены), так как сводится к единственному умножению матрицы на вектор.

Аналогичного результата можно было бы добиться, решая систему линейных алгебраических уравнений с помощью LU-разложения. Однако, эффективная реализация LU-разложения на GPU нетривиальна, а при использовании сторонних библиотек (например, «CUBLAS») отсутствует возможность использования сжатия. Последнее, как отмечено выше, критично для алгоритма излучательности (рис. 6 и 7).

Работа поддержана грантом РФФИ 16–31–60048 мол\_а\_дк.

#### СПИСОК ЛИТЕРАТУРЫ

1. Keller A. Instant radiosity / Proc. 24th annual conf. on Computer graphics and interactive techniques (SIGGRAPH'97). – New York, NY, USA: ACM Press/Addison-Wesley

Publishing Co. – P. 49–56. DOI=http://dx.doi.org/10.1145/258734.258769

2. Будак В.П., Желтов В.С., Калакуцкий Т.К. Локальные оценки метода Монте-Карло в решении уравнения глобального освещения с учётом спектрального представления объектов // Компьютерные исследования и моделирование. – 2012. – Т. 4, № 1. – С. 75–84.

3. Dachsbacher C., Stamminger M. Reflective shadow maps / Proc. 2005 symp. on Interactive 3D graphics and games (I3D '05). – New York, NY, USA: ACM – P. 203–231.

4. Kaplanyan A., Dachsbacher C. Cascaded light propagation volumes for real time indirect illumination / Proc. 2010 ACM SIGGRAPH symp. on Interactive 3D Graphics and Games (I3D'10). – New York, NY, USA: ACM. – P. 99–107. DOI=http://dx.doi.org/10.1145/1730804.1730821

5. Crassin C., Neyret F., Sainz M., Green S., Eiseman E. Interactive indirect illumination using voxel-based cone tracing: an insight / ACM SIGGRAPH 2011 Talks (SIGGRAPH '11). – New York, NY, USA:

ACM. – Article 20, 1 pages. DOI=http://dx.doi.org/10.1145/2037826.2037853.

6. Lisle I.G., Tracy Huang S.-L. Algorithms for spherical harmonic lighting / Proc. 5th int. conf. on Computer graphics and interactive techniques in Australia and Southeast Asia (GRAPHITE '07). – New York, NY, USA: ACM. – P. 235–238.

7. Goral C.M., Torrance K.E., Greenberg D.P., Battaile B. Radiosity on graphics hardware modeling the interaction of light between diffuse surfaces / Proc. 11th annual conf. on Computer graphics and interactive techniques (SIGGRAPH'84), Hank Christiansen (Ed.).

8. Martin S., Einarsson P. A Real Time Radiosity Architecture for Video Games / Siggraph 2010. URL: http://www.geomerics.com/wpcontent/uploads/2014/03/radiosity\_architecture.pdf (дата обращения: 01.2018).

9. Щербаков А., Фролов В. Автоматическое упрощение геометрии для расчёта вторичной освещённости методом излучательности / Тр. 26 Межд. конф. по компьют. графике и зрению. 19–23 сентября 2016, Нижний Новгород, с. 34–38.



**Щербаков Александр Станиславович**, студент-магистр факультета ВМК МГУ им. М.В. Ломоносова



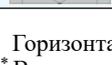
**Фролов Владимир Александрович**, кандидат физ.-мат. наук. Окончил факультет ВМК МГУ им. М.В. Ломоносова. Научный сотрудник МГУ им. М.В. Ломоносова и ИПМ им. М.В. Келдыша РАН

## От редакции

В статье К. Брожио, С. Вёлкера, М. Кнооп и М. Нидлинга «О влиянии различных зон поля зрения на не связанное с формированием изображения воздействие света», опубликованной в № 2 журнала за 2018 г. (стр. 28-32), допущена ошибка: в таблице отсутствует последняя строка. Опубликованную таблицу следует заменить на:

Таблица

### Значения освещённости

Зоны		Освещаемая сцена							
									
		<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
Освещённость	$E_v$ , лк	506	511	504	500	497	500	507	496
	$E_h^*$ , лк	657	694	619	640	263	276	329	338
	$E_v^{**}$ , %	2,8	3,2	2,0	2,8	9,5	10,6	7,5	8,5
	$E_v^{**}$ , %	15,1	19,2	5,6	17,3	24,3	29,2	28,6	19,1
	$E_v^{**}$ , %	2,8	3,1	1,9	2,6	9,9	11,1	7,7	7,3
	$E_v^{**}$ , %	20,7	25,5	9,5	22,6	43,7	50,8	43,8	34,9
	$E_v^{**}$ , %	1,5	1,7	1,0	1,3	1,2	1,3	1,5	2,1
	$E_v^{**}$ , %	14,0	11,0	18,6	18,4	4,2	4,2	9,0	10,5
	$E_v^{**}$ , %	1,5	1,7	0,8	1,2	1,6	1,6	1,8	5,4
	$E_v^{**}$ , %	37,7	39,9	30,0	43,6	50,7	58,0	56,1	52,9

\* Горизонтальная освещённость на высоте 0,85 м.

\*\* Вертикальная освещённость в процентах от измеренного люксметром значения.

Редакция приносит свои извинения авторам статьи и читателям журнала.