

# ГАММА-КОРРЕКЦИЯ ИЗОБРАЖЕНИЙ ПРИ РЕНДЕРИНГЕ ОБЪЕКТОВ МЕТОДОМ ТРАССИРОВКИ ЛУЧЕЙ

*Рассматривается программная реализация гамма-коррекции реалистичного изображения при рендеринге методом трассировки лучей.*

## ВВЕДЕНИЕ

Гамма-коэффициент или просто гамма определяет отношение между численным значением пикселя и его действительной светимостью. Без коррекции гаммы тёмные тона, снятые цифровыми камерами, не выглядели бы так, как их видят наши глаза. Однако гамма присутствует не только камерам — собственную гамму имеют файлы изображений, экраны и практически любое другое устройство отображения. Говоря о коррекции гаммы, кодировании гаммы или компрессии (сжатии) гаммы, подразумевают одно и то же понятие. Понимание того, что собой представляет гамма, может помочь улучшить технику экспозиции, а также извлечь максимум из обработки изображений.

### I. ЗАЧЕМ НУЖНА КОРРЕКЦИЯ ГАММЫ

Наши глаза видят иначе, чем камеры. В цифровой камере удвоенное количество фотонов, попадающих на сенсор, означает удвоение сигнала (зависимость «линейна»). Вполне логично, да? Однако наши глаза устроены иначе. Для нас увеличение освещённости вдвое означает, что свет стал слегка ярче (зависимость «нелинейна»).

По сравнению с камерами мы более чувствительны к малейшим изменениям тёмных оттенков и менее чувствительны к достаточно большим изменениям в ярких тонах. Для такой странности есть свои биологические причины: это позволяет нашему зрению работать в более широком диапазоне освещённости. В противном случае типичный диапазон яркостей, с которым мы сталкиваемся на улице, был бы невыносим.

В данном случае гаммой мы называем преобразование к светочувствительности наших глаз показаний камеры. Когда сохраняется цифровое изображение, оно подвергается «гамма-кодированию» — так чтобы удвоение значения в файле ближе соответствовало тому, что мы воспринимаем как удвоение яркости.

В качестве передаточной функции при гамма-коррекции чаще всего используется степенная в виде (1):

$$V_{out} = AV_{in}^{\gamma} \quad (1)$$

где  $A$  служит коэффициентом, а входные  $V_{out}$  и выходные  $V_{in}$  значения — неотрицатель-

ные вещественные числа. В общем случае, если  $A = 1$ , то входные и выходные значения находятся в пределах от 0 до 1. При равенстве  $\gamma$  единице характеристика передачи полутонов линейна и перепады освещённости объекта в светах и тенях отображаются одинаково.

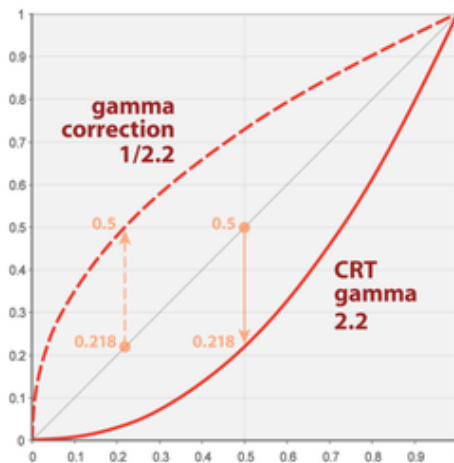


Рис. 1 – Пример гамма-коррекции изображения на ЭЛТ. Средняя линия из точек является результирующей функцией передачи полутонов; сплошная кривая отражает характеристику монитора, а пунктирная кривая — гамма-коррекцию

В случае, когда этот параметр меньше единицы, улучшается распознавание деталей на слабо освещённых участках. Такое соотношение, называемое «гаммой кодирования», используется при преобразовании оптического изображения в электрический сигнал или цифровой файл в передающих камерах и цифровых фотоаппаратах. При воспроизведении полученного сигнала на кинескопе, за счёт особенностей его световых характеристик происходит обратное преобразование, в результате которого результирующая гамма всей системы приближается к единице, обеспечивая пропорциональную передачу полутонов во всём диапазоне[4]. Аналогичный процесс происходит при воспроизведении изображения на жидкокристаллических дисплеях за счёт цепей обратной коррекции видеокарт[5].

### II. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

В грубом приближении была использована «гамма 2», что означает возведение цвета в сте-

пень  $1/\gamma$ , или, в нашем простом случае,  $1/2$ , что является просто квадратным корнем:

```
void write_color(std::ostream &out, color
pixel_color, int samples_per_pixel) {
    auto r = pixel_color.x();
    auto g = pixel_color.y();
    auto b = pixel_color.z();
    // Divide the color by the number of samples
and gamma-correct for gamma=2.0.
    auto scale = 1.0 / samples_per_pixel;
    r = sqrt(scale * r);
    g = sqrt(scale * g);
    b = sqrt(scale * b);
    // Write the translated [0,255] value of each
color component.
    out << static_cast<int>(256 * clamp(r, 0.0,
0.999)) << ' '
    << static_cast<int>(256 * clamp(g, 0.0,
0.999)) << ' '
    << static_cast<int>(256 * clamp(b, 0.0,
0.999)) << '\n';
}
```

После обработки изображения (рисунок 2) с помощью данной функции, получим скорректированное изображение (рисунок 3):

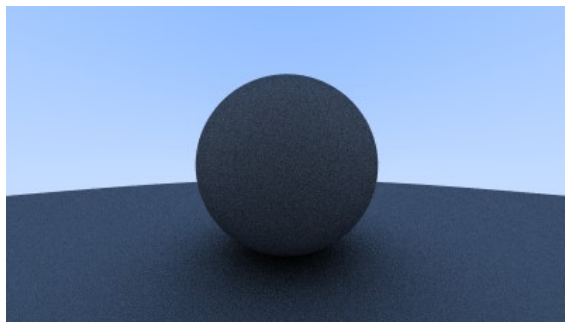


Рис. 2 – Нескорректированное изображение, полученное в результате рендеринга методом трассировки лучей [1-3]

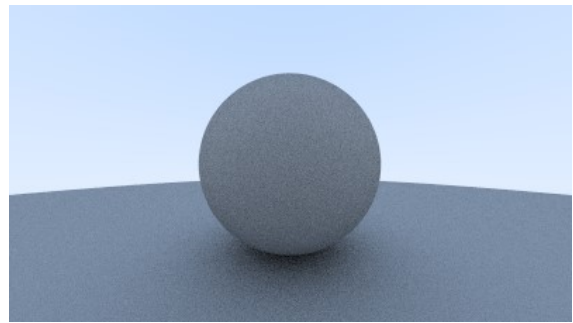


Рис. 3 – Изображение, полученное в результате рендеринга методом трассировки лучей, после применения гамма-коррекции

### III. ВЫВОДЫ

Результаты полученные в результате обработки исходного изображения являются применимыми, хотя в качестве параметра была выбрана 2, вместо 2.2. На визуальную составляющую это повлияло слабо, зато был получен выигрыш в скорости рендеринга.

#### *Список литературы*

1. Roth, S. D. Ray casting for modeling solids // Computer Graphics and Image Processing. – 1982. – №18. – P. 109-144.
2. Appel, A. Some techniques for shading machine renderings of solids // AFIPS spring joint computer conference. IBM Research Center, Yorktown Heights, N.Y. – 1968. – P. 37-45.
3. Рак, Т. А., Реалистичный рендеринг на основе метода трассировки лучей / Т. А. Рак // 56-я научная конференция аспирантов, магистрантов и студентов учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» – 2020. – С. 147–148.
4. Джакония, В. Е. Глава 4. Искажения телевизионного изображения // Телевидение. М.: «Горячая линия — Телеком» –2002. – С. 59–61.
5. Johnson, C. Chapter 10. The Zone System and Digital Photography // The Practical Zone System for Film and Digital Photography Focal Print –2007. — 285 p.

*Рак Татьяна Александровна*, магистрант кафедры информационных технологий автоматизированных систем БГУИР, tatianarak@bsuir.com.

*Научный руководитель: Алёхина Алина Энодиевна*, доцент кафедры экономической информатики БГУИР, кандидат технических наук, доцент, ae.alekhina@gmail.com.