# Гамма-коррекция

Рассмотрим сначала частичное решение (работаем на черном фоне), когда вы уже вычислили интенсивность пикселя, который нужно нарисовать. Самое время отобразить их на мониторе. На заре цифровой обработки изображений большинство мониторов имели электронно-лучевые трубки (ЭЛТ). Этот тип мониторов имел физическую особенность: повышение входного напряжение в два раза не означало двукратного увеличения яркости. Зависимость между входным напряжением и яркостью выражалась степенной функцией, с показателем примерно 2.2, также известным как **гамма** монитора.

Эта особенность мониторов (по случайному совпадению) очень напоминает то, как люди воспринимают яркость: с подобной же (но обратной) степенной зависимостью. Чтобы лучше это понять, взгляните на следующее изображение:



Vs показывает как воспринимается яркость человеческим глазом: при увеличении яркости в 2 раза (например, от 0.1 до 0.2) картинка действительно выглядит так, будто она в два раза ярче: изменения видны довольно отчетливо. Однако, когда мы говорим о физической яркости света, как, например, о количестве фотонов, выходящих из источника света, верную картину дает нижняя шкала (I). На ней удвоение значения дает правильную с физической точки зрения яркость, но поскольку наши глаза более восприимчивы к изменениям темных цветов, это кажется несколько странным.

Поскольку для человеческого глаза более привычен верхний вариант, мониторы и по сей день используют степенную зависимость при выводе цветов, так что исходные, в физическом смысле, значения яркости преобразуются в нелинейные значения яркости, изображенные на верхней шкале. В основном это сделано потому, что так выглядит лучше.

Эта особенность мониторов действительно делает картинку лучше для наших глаз, но когда дело доходит до рендеринга графики появляется одна проблема: все параметры цвета и яркости, которые мы устанавливаем в наших приложениях, основаны на том, что мы видим на мониторе. А это означает что все эти параметры на самом деле являются нелинейными.

|  |  |
| --- | --- |
|  | * Серая линия соответствует значениям цвета в линейном пространстве; * Сплошная красная линия представляет собой цветовое пространство отображаемое монитором. * Важное примечание: дальше разговор будет вестись про яркость в диапазоне [0..1], а не [0..255], как мы привыкли (т.е. используются нормированные значения яркости) |

Когда мы хотим получить в 2 раза более яркий цвет в линейном пространстве, мы просто берем и удваиваем его значение. Например, возьмем яркость 0.5. Если бы мы удвоили его значение в линейном пространстве, он стал бы равным 1. С другой стороны, при выводе на дисплей, он будет преобразован в цветовое пространство монитора как 0.218, как видно из графика. Вот здесь и возникает проблема: удваивая цвет в линейном пространстве, мы фактически делаем его более чем в 4.5 раза ярче на мониторе.

До этого момента мы предполагали, что работали в линейном пространстве, но на самом деле мы работали в цветовом пространстве, определяемом монитором, поэтому все установленные нами цвета были физически не корректны, а всего лишь выглядели правильными конкретно на нашем мониторе. Руководствуясь данным предположением принято устанавливать значения освещения ярче, чем они должны быть (т.к. монитор затемняет их), что в результате делает большинство последующих вычислений в линейном пространстве неверными. Также обратите внимание, что оба графика начинаются и заканчиваются в одних и тех же точках, затемнению на дисплее подвержены только промежуточные цвета.

**Идея гамма-коррекции** заключается в том, чтобы применить инверсию гаммы монитора к окончательному цвету перед выводом на монитор (записью в файл). Снова посмотрим на график гамма-кривой, обратив внимание на еще одну линию, обозначенную штрихами, которая является обратной для гамма-кривой монитора. Мы умножаем выводимые значения цветов в линейном пространстве на эту обратную гамма-кривую ( делаем их ярче), и как только они будут выведены на монитор, к ним применится гамма-кривая монитора, и результирующие цвета снова станут линейными. По сути мы делаем промежуточные цвета ярче, чтобы сбалансировать их затенение монитором.

Приведем еще один **пример**. Допустим, у нас опять есть наш серый цвет 0.5. Перед отображением этого цвета на монитор мы сперва применяем кривую гамма-коррекции к его компонентам. Значения цвета в линейном пространстве при отображении на мониторе возводятся в степень, приблизительно равную 2.2, поэтому инверсия требует от нас возведения значений в степень 1/2.2. Таким образом, наш цвет с гамма-коррекцией становится 0.5 ^ 1/2.2 = 0.73. Затем этот скорректированные цвет выводится на монитор, и в результате он отображается как 0.73 ^ 2.2 = 0.5.

Как видите, когда мы используем гамма-коррекцию монитор отображает цвета, точно такими, какими мы задаем их в линейном пространстве в нашем приложении.

*Гамма равная 2.2 это дефолтное значение, которое приблизительно выражает среднюю гамму большинства дисплеев. Цветовое пространство в результате применения этой гаммы называется цветовым пространством sRGB. Каждый монитор имеет свои собственные гамма-кривые, но значение 2.2 дает хорошие результаты на большинстве мониторов. Из-за этих небольших отличий многие игры позволяют игрокам изменять настройку гаммы.*

|  |  |
| --- | --- |
|  | Пример на цветных градиентах:   * сверху - с применением гамма коррекции * снизу - без применения гамма коррекции |

## sRGB

**sRGB** является стандартом представления цветового спектра с использованием модели [RGB](https://ru.wikipedia.org/wiki/RGB). sRGB создан для унификации использования модели RGB в мониторах, принтерах и Интернет-сайтах.

sRGB использует основные цвета, описанные стандартом [BT.709](https://ru.wikipedia.org/w/index.php?title=Rec._709&action=edit&redlink=1), аналогично студийным мониторам и HD-телевидению, а также [гамма-коррекцию](https://ru.wikipedia.org/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B0-%D0%BA%D0%BE%D1%80%D1%80%D0%B5%D0%BA%D1%86%D0%B8%D1%8F), аналогично мониторам с электронно-лучевой трубкой. Такая спецификация позволила sRGB в точности отображаться на обычных CRT-мониторах и телевизорах, что стало в своё время основным фактором, повлиявшим на принятие sRGB в качестве стандарта.

В отличие от большинства других цветовых пространств RGB, [гамма](https://ru.wikipedia.org/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B0-%D0%BA%D0%BE%D1%80%D1%80%D0%B5%D0%BA%D1%86%D0%B8%D1%8F) в sRGB не может быть выражена одним числовым значением, так как функция коррекции состоит из линейной части около чёрного цвета, где гамма равна 1.0, и нелинейной части до значения 2.4 включительно. Приблизительно можно считать, что гамма равна 2.2. Гамма может изменяться от 1.0 до 2.3.

# Альфа смешивание

***Альфа-значение*** определяет степень прозрачности конкретного пикселя. Объекты могут иметь различную прозрачность, например 8-битный альфа-канал может представлять 256 уровней прозрачности: от 0 (всё растровое изображение прозрачное) до 255 (всё растровое изображение непрозрачное).

***Альфа-смешение*** есть процесс комбинирования двух объектов на экране с учётом их альфа-каналов. Альфа-смешение используется для: [антиалиасинга](https://ru.wikipedia.org/wiki/%D0%A1%D0%B3%D0%BB%D0%B0%D0%B6%D0%B8%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5); создания прозрачности, теней, зеркал, тумана.

Расчёт яркости результирующего пикселя после наложения двух пикселей друг на друга выполняется по формуле:

**R = B ⋅ ( 1 − A ) + F ⋅ A**  или в иной записи: **R = B + ( F − B ) ⋅ A**

Обозначения:

* B — яркость фонового пикселя;
* F — яркость накладываемого пикселя;
* A ∈ [ 0..1 ] — непрозрачность накладываемого пикселя;
* R ― результат.

Вторая запись отображает следующий смысл: значение A указывает относительное положение на отрезке [B..F].

|  |
| --- |
|  |
| Эффекты гамма-некорректного альфа-смешения. Для каждой пары вертикальных столбцов верхняя рисуется с непрозрачностью 100%, а нижняя с 50%.  Левое изображение - гамма-правильное. |

# **Источники (полезные ссылки)**:

1. [What every coder should know about gamma](http://blog.johnnovak.net/2016/09/21/what-every-coder-should-know-about-gamma/)
2. <https://habr.com/en/post/353632/>
3. [Gamma correction](https://en.wikipedia.org/wiki/Gamma_correction)
4. [Гамма-коррекция](https://ru.wikipedia.org/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B0-%D0%BA%D0%BE%D1%80%D1%80%D0%B5%D0%BA%D1%86%D0%B8%D1%8F)
5. [sRGB](https://en.wikipedia.org/wiki/SRGB)
6. [Understanding Gamma Correction](https://www.cambridgeincolour.com/tutorials/gamma-correction.htm)
7. [Альфа-канал](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D1%8C%D1%84%D0%B0-%D0%BA%D0%B0%D0%BD%D0%B0%D0%BB)