

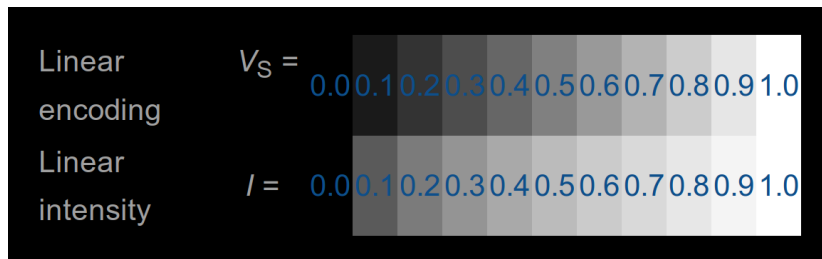
Компьютерная графика

Лекция 10: Gamma correction, sRGB, color banding, dithering, deferred shading, tiled/clustered shading

2021

- ▶ Обычно, интенсивность света I , излучаемого монитором, нелинейно зависит от значения V , записанного в пикселе

Линейное значение пикселя vs линейная интенсивность излучения



Гамма

- ▶ Обычно, интенсивность света I , излучаемого монитором, нелинейно зависит от значения V , записанного в пикселе
- ▶ Это лучше соответствует восприятию света человеком

Гамма

- ▶ Обычно, интенсивность света I , излучаемого монитором, нелинейно зависит от значения V , записанного в пикселе
- ▶ Это лучше соответствует восприятию света человеком
- ▶ Почти всегда используется показательная функция:

$$I \sim V^\gamma \quad (1)$$

Гамма

- ▶ Обычно, интенсивность света I , излучаемого монитором, нелинейно зависит от значения V , записанного в пикселе
- ▶ Это лучше соответствует восприятию света человеком
- ▶ Почти всегда используется показательная функция:

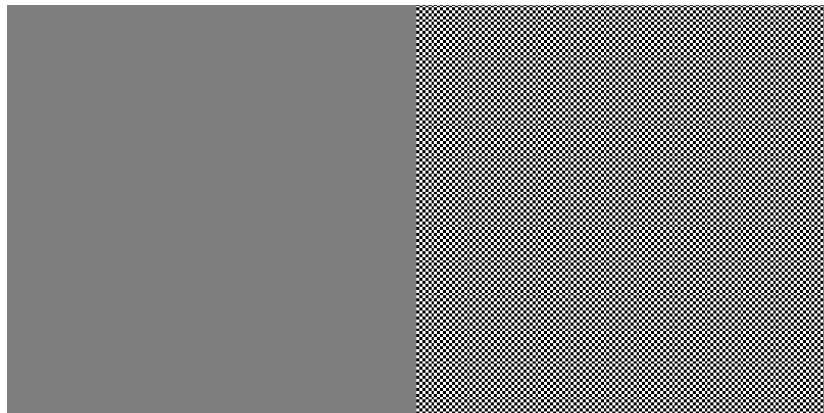
$$I \sim V^\gamma \quad (1)$$

- ▶ γ обычно равна 2.2 (некоторые компьютеры Macintosh использовали 1.8)

Проблемы гаммы

- ▶ Картинка может издалека выглядеть ярче, чем её усреднённый вариант (e.g. mipmap)

Серый (цвет=0.5) квадрат и квадрат с мелкой шахматной раскраской



Проблемы гаммы

- ▶ Картинка может издалека выглядеть ярче, чем её усреднённый вариант (e.g. mipmap)
- ▶ Искажается восприятие относительных яркостей, особенно при реалистичном рендеринге (e.g. объект в два раза ярче не будет выглядеть в два раза ярче)

Проблемы гаммы

- ▶ Картинка может издалека выглядеть ярче, чем её усреднённый вариант (e.g. mipmap)
- ▶ Искажается восприятие относительных яркостей, особенно при реалистичном рендеринге (e.g. объект в два раза ярче не будет выглядеть в два раза ярче)
- ▶ Неправильно выглядит освещение, наложение источников света, и т.д.

Коррекция гаммы (gamma-correction)

- ▶ Коррекция гаммы - общий термин для применения любых нелинейных преобразований над интенсивностью пикселя

Коррекция гаммы (gamma-correction)

- ▶ Коррекция гаммы - общий термин для применения любых нелинейных преобразований над интенсивностью пикселя
- ▶ В рендеринге под гамма-коррекцией обычно подразумевают применение обратного к $V^{2.2}$ преобразования, чтобы получить линейную зависимость выходящего излучения от значения пикселя

Коррекция гаммы (gamma-correction)

- ▶ Коррекция гаммы - общий термин для применения любых нелинейных преобразований над интенсивностью пикселя
- ▶ В рендеринге под гамма-коррекцией обычно подразумевают применение обратного к $V^{2.2}$ преобразования, чтобы получить линейную зависимость выходящего излучения от значения пикселя
- ▶ Делает картинку ярче и часто более реалистичной

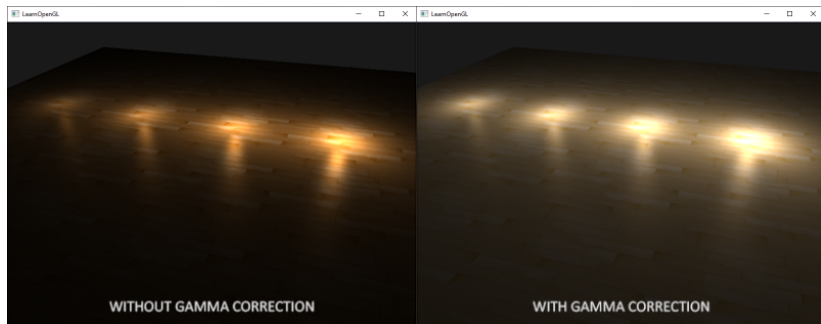
Коррекция гаммы (gamma-correction)

```
// Вычислили цвет пикселя  
// с учётом освещения  
vec3 color = ...;  
  
color = pow(color, vec3(1.0 / 2.2));
```

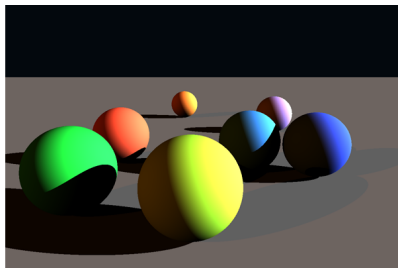
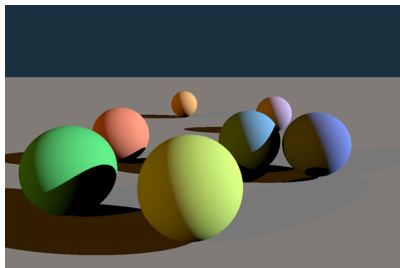
Эффекты коррекции гаммы



Эффекты коррекции гаммы



Эффекты коррекции гаммы



Коррекция гаммы: ссылки

- ▶ en.wikipedia.org/wiki/Gamma_correction
- ▶ What every coder should know about gamma
- ▶ Linear-space lighting (i.e. gamma)
- ▶ Тutorial на learnopengl.com
- ▶ Пример с гамма-коррекцией

- ▶ По тем же причинам, по которым существует нелинейная гамма у мониторов (отдать больше бит под тёмные цвета), хочется хранить текстуры не в сыром формате, а в гамма-преобразованном: значение пикселя V зависит от желаемой интенсивности I как $V \sim I^{\frac{1}{\gamma}}$

- ▶ По тем же причинам, по которым существует нелинейная гамма у мониторов (отдать больше бит под тёмные цвета), хочется хранить текстуры не в сыром формате, а в гамма-преобразованном: значение пикселя V зависит от желаемой интенсивности I как $V \sim I^{\frac{1}{\gamma}}$
- ▶ Такой формат хранения цвета называется sRGB
 - ▶ N.B. Точная формула преобразования sRGB отличается от $I^{\frac{1}{2.2}}$, но очень близка к ней

- ▶ По тем же причинам, по которым существует нелинейная гамма у мониторов (отдать больше бит под тёмные цвета), хочется хранить текстуры не в сыром формате, а в гамма-преобразованном: значение пикселя V зависит от желаемой интенсивности I как $V \sim I^{\frac{1}{\gamma}}$
- ▶ Такой формат хранения цвета называется sRGB
 - ▶ N.B. Точная формула преобразования sRGB отличается от $I^{\frac{1}{2.2}}$, но очень близка к ней
- ▶ Обычно изображения хранятся именно в таком формате

- ▶ По тем же причинам, по которым существует нелинейная гамма у мониторов (отдать больше бит под тёмные цвета), хочется хранить текстуры не в сыром формате, а в гамма-преобразованном: значение пикселя V зависит от желаемой интенсивности I как $V \sim I^{\frac{1}{\gamma}}$
- ▶ Такой формат хранения цвета называется sRGB
 - ▶ N.B. Точная формула преобразования sRGB отличается от $I^{\frac{1}{2.2}}$, но очень близка к ней
- ▶ Обычно изображения хранятся именно в таком формате
- ▶ Проверить формат можно любым редактором изображений или программой `identify` пакета `imagemagick`

- ▶ При чтении из sRGB-текстуры нужно возводить прочитанное значение в степень 2.2

- ▶ При чтении из sRGB-текстуры нужно возводить прочитанное значение в степень 2.2
- ▶ При записи в sRGB-текстуру нужно возводить записываемое значение в степень $1/2.2$

sRGB

- ▶ При чтении из sRGB-текстуры нужно возводить прочитанное значение в степень 2.2
- ▶ При записи в sRGB-текстуру нужно возводить записываемое значение в степень $1/2.2$
- ▶ В OpenGL есть поддержка sRGB для текстур и фреймбуферов

sRGB-текстуры

- ▶ Специальное значение `internal format` для текстуры (`GL_SRGB8` или `GL_SRGB8_ALPHA8`) означает, что текстура хранит sRGB-значения - они будут *автоматически* переведены в линейные значения (т.е. возведены в степень 2.2) при чтении из шейдера

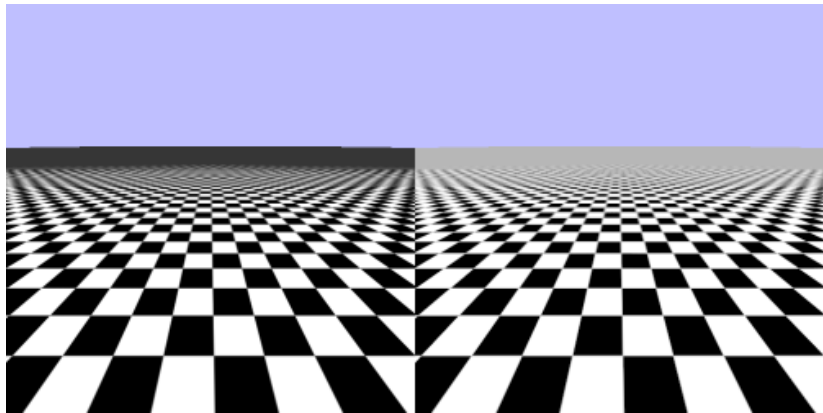
sRGB-текстуры

- ▶ Специальное значение `internal format` для текстуры (`GL_SRGB8` или `GL_SRGB8_ALPHA8`) означает, что текстура хранит sRGB-значения - они будут *автоматически* переведены в линейные значения (т.е. возведены в степень 2.2) при чтении из шейдера
- ▶ `glEnable(GL_FRAMEBUFFER_SRGB)` включит автоматическое обратное преобразование (возведение в степень $1/2.2$) при рисовании в sRGB-текстуру

sRGB-текстуры

- ▶ Специальное значение internal format для текстуры (GL_SRGB8 или GL_SRGB8_ALPHA8) означает, что текстура хранит sRGB-значения - они будут *автоматически* переведены в линейные значения (т.е. возведены в степень 2.2) при чтении из шейдера
- ▶ glEnable(GL_FRAMEBUFFER_SRGB) включит автоматическое обратное преобразование (возведение в степень 1/2.2) при рисовании в sRGB-текстуру
- ▶ **N.B.:** glGenerateMipmap *не обязан* правильно обрабатывать sRGB-текстуры!

sRGB-correct mipmap



sRGB: ссылки

- ▶ en.wikipedia.org/wiki/SRGB
- ▶ Подробный текст про sRGB на stackoverflow
- ▶ sRGB framebuffers