



Спецкурс ОСФИ

Лекция 5

23 марта 2011

Гамма-коррекция

Алексей Игнатенко, к.ф.-м.н.

Лаборатория компьютерной графики и
мультимедиа ВМК МГУ

На лекции

- Что такое гамма-коррекция
- Зачем необходимо корректировать яркость
- Гамма-коррекция в разных операционных системах
- Применении гамма-коррекции в синтезе изображений

Светлота (lightness)

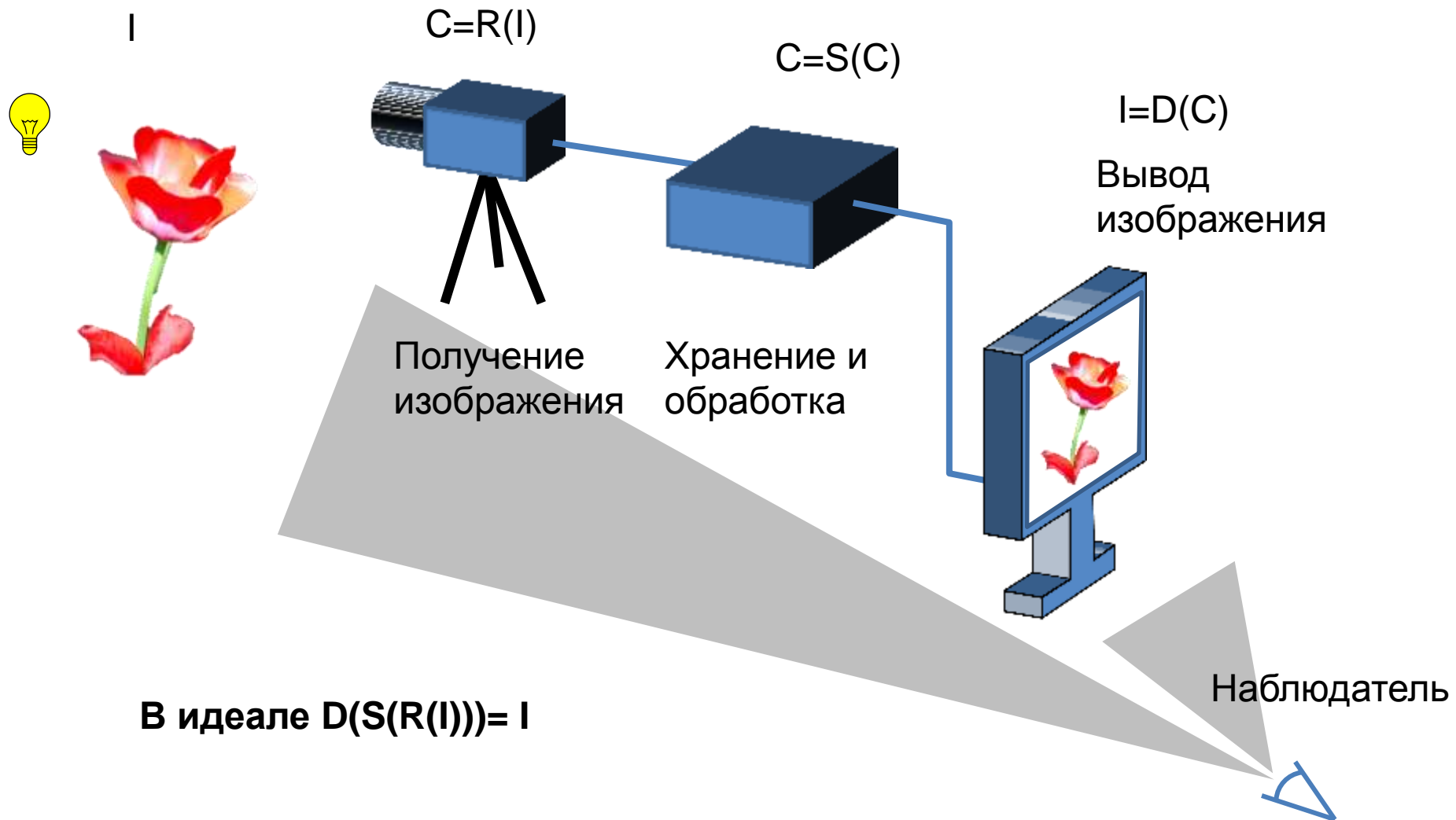
- Человеческое зрение имеет нелинейный отклик на яркость.
- Источник света яркостью 18% по сравнению с базовым будет казаться вполовину менее ярким.
- Восприятие света человеком описывается светлотой источника
 - Яркость (luminance) описывает спектральную составляющую восприятия
 - Светлота описывает относительную мощностную характеристику восприятия

$$L^* = 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16; \quad 0.008856 < \frac{Y}{Y_n}$$

Что хранит пиксель изображения?

- Если убрать цветовую составляющую
- Желательно кодировать энергетическую яркость (radiance)
- Ну или хотя бы световую яркость (luminance)
 - Легко получить с помощью преобразования в хуз и рассмотрения компоненты Y
- Вопрос в линейности!

Тракт передачи изображений и передающие функции



Проблемы с передающими функциями

- Особенности передающих функцией обусловлены физическим устройством приемника и дисплея
- При хранении информации в цифровом виде неизбежна дискретизация сигнала
 - Потеря информации!
- Передающие функции дисплея и камеры

Почему необходимо корректировать яркость?

Причина 1: Нелинейность передающей функции CRT-дисплеев

Причина 2: Необходимость нелинейного кодирования яркости для более полного использования ограниченного диапазона представления яркости в ЭВМ

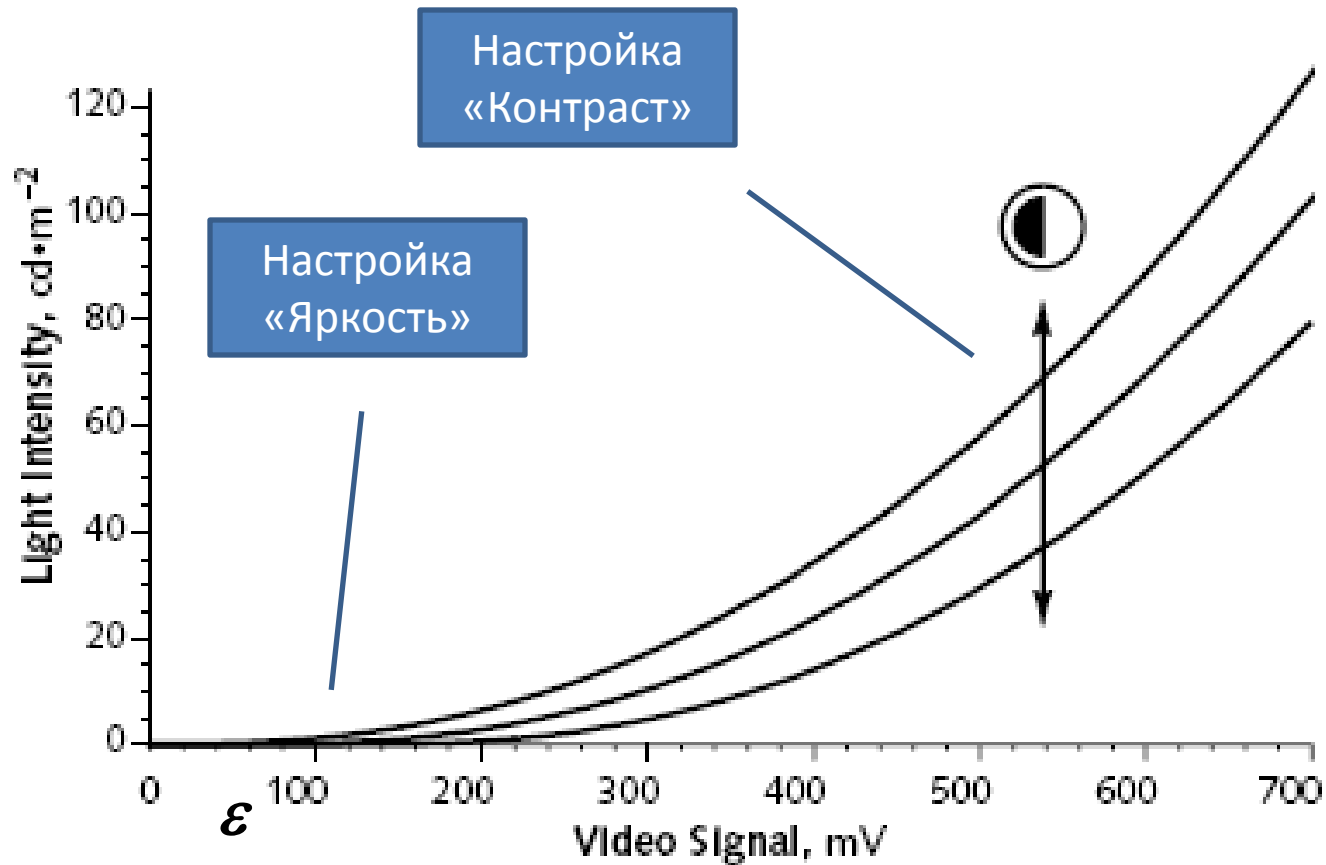
Причина 3: Особенности восприятия интенсивностей человеческим зрительным аппаратом

Причина 1: Передающая функция монитора

- Интенсивность света, генерируемого физическим устройством не является линейной функцией входящего сигнала
- CRT-устройства (телевизоры, мониторы) имеют степенную зависимость интенсивности излучения от входящего напряжения:

$$D(C) = k(C + \varepsilon)^\gamma$$

Причина 1: Передающая функция монитора



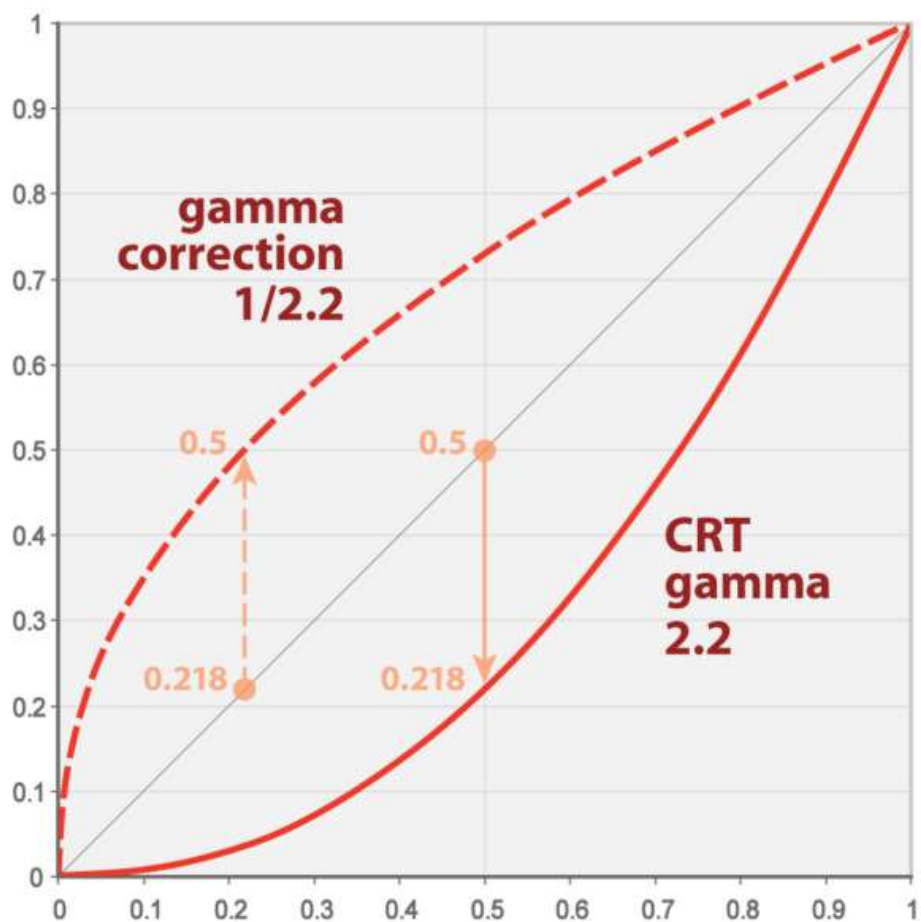
Что такое гамма?

- Гамма – характеристика нелинейности интенсивности сигнала, выдаваемого физическим устройством вывода.
- Обычно 2.35 – 2.55
- Гамма-преобразование – нелинейное преобразование формы $k(V)^\gamma$
- Гамма-коррекция – процесс компенсации нелинейного преобразования устройства вывода. Преобразование формы $kV^{(1/\gamma)}$
- Гамма коррекция необходима для более точной передачи интенсивностей монитором
- Но не только!

Gamma compression & Gamma expansion

- Гамма-сжатие = гамма-коррекция
- Гамма-расширение = гамма-преобразование

Функция гамма-коррекции



Применяется к относительным яркостям!

CRT и LCD-мониторы

- LDC-мониторы аппаратно полностью линейны по передаче интенсивности
- Но ведут себя как CRT, делая гамма-преобразование перед выводом изображения
- Почему?
 - Для совместимости
 - Для лучшего соответствия «логарифмическому» восприятию яркости человеком (см. далее)

Причина 2: Кодирование яркости с учетом восприятия яркости человеком

- Имеем на входе произвольный диапазон яркости
- Хотим закодировать его в компьютере с заданным количеством бит на пиксель (целочисленно)
- При различном кодировании шаг между соседними значениями цвета будет разным

Какой должен быть шаг, чтобы при просмотре изображения человек видел плавные градации?

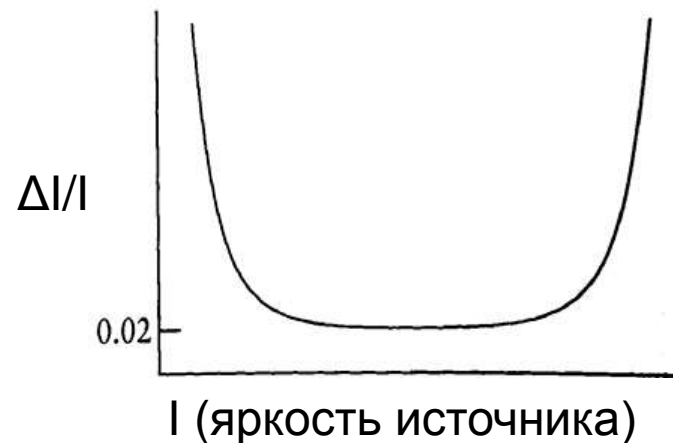
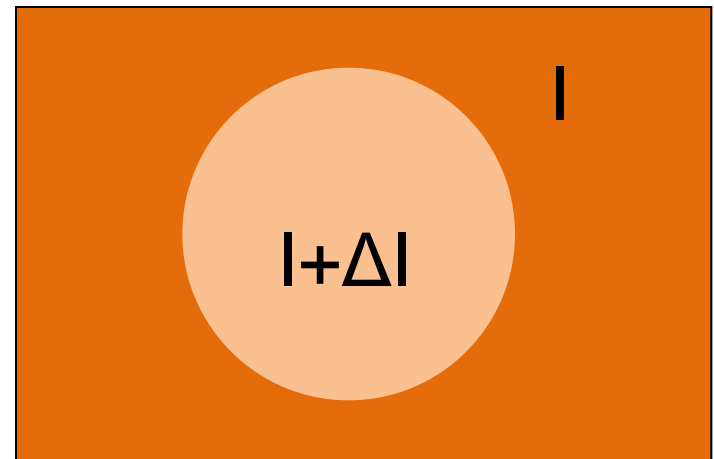
Причина 2: Кодирование яркости с учетом восприятия яркости человеком

- Человеческое зрение имеет логарифмическую характеристику восприятия интенсивности света
- Приближаем функцией степени

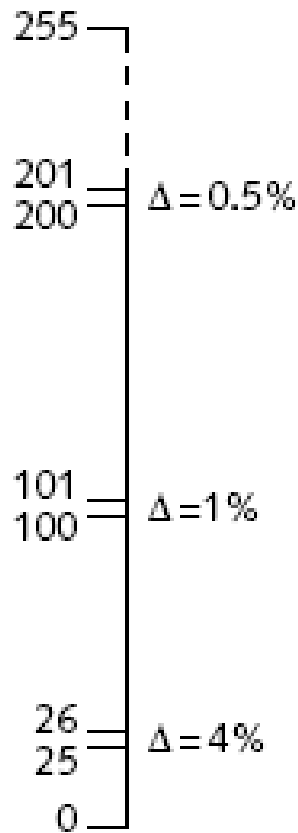
$$L^* = 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16; \quad 0.008856 < \frac{Y}{Y_n}$$

Чувствительность к контрасту: минимальная различимая разница (закон Вебера)

- Задача – найти минимальную различимую разницу интенсивностей ΔI
 - JND – just noticeable difference
- Экспериментально получена кривая чувствительности к контрасту (закон Вебера)
- $\Delta I/I \sim \text{const} \sim 0.02$
- Вывод: глаз реагирует на относительные интенсивности!
- Вывод: меньше 2% шаг человек не различит



Причина 2: Линейное кодирование яркости



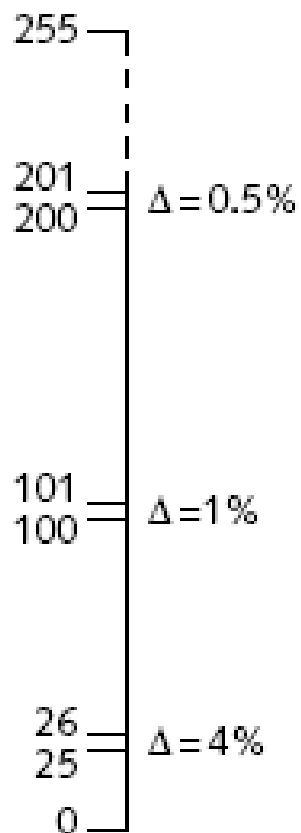
На яркостях меньше 100 будут видимые границы или ступени

Яркости больше 200 идут излишне «часто», т.к. неразличимы.

В 8-бит системе максимальный контраст 2.55:1

(человек различает контраст до 10000:1 !)

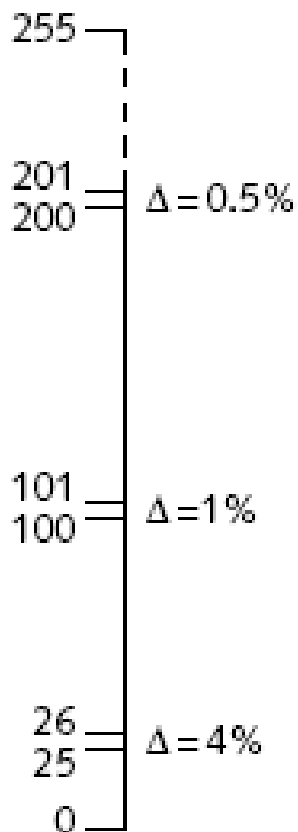
Причина 2: Кодирование яркости



- Гамма-коррекция позволяет выделить больше бит на яркость там, где это лучше всего видно



Причина 2: Кодирование яркости



Без гамма-коррекции:

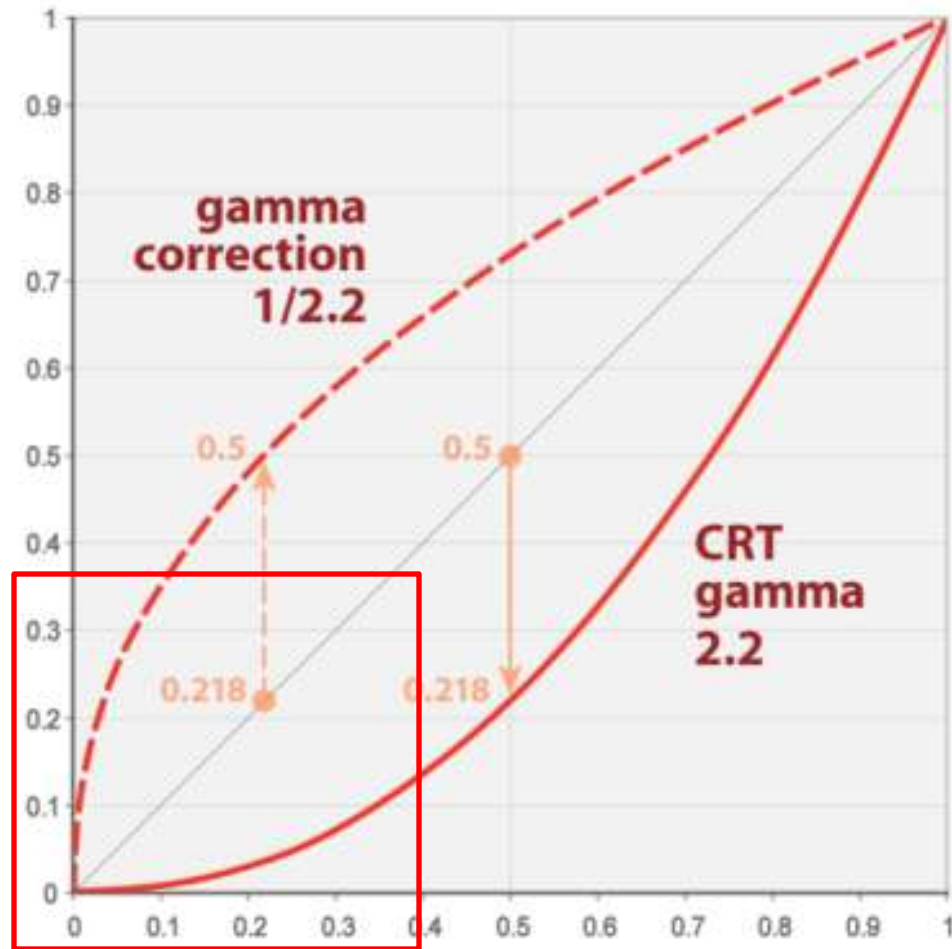
26/25 - разница 4%

После гамма-коррекции

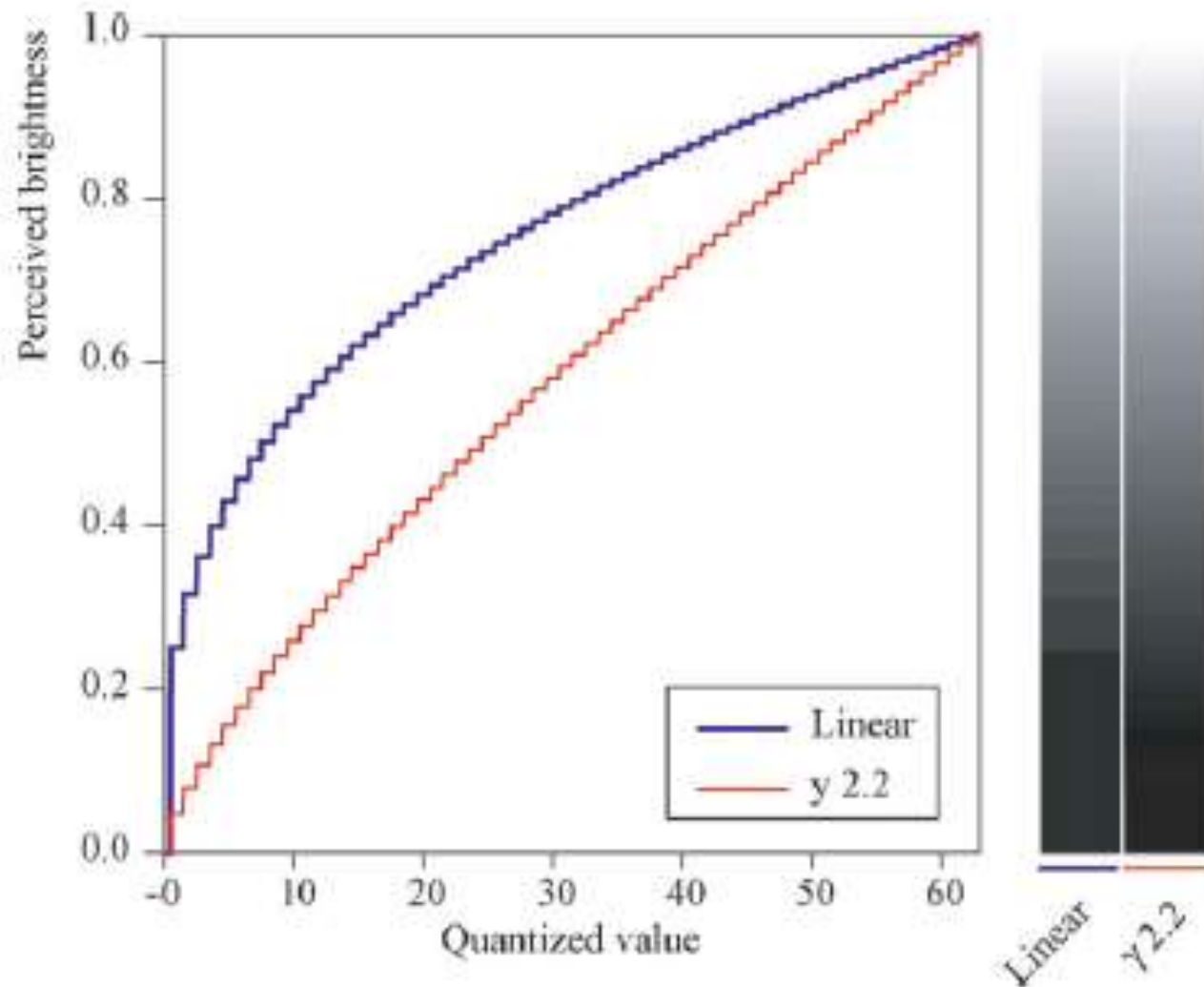
88/90 - разница 1,8%

Причина 2: Кодирование яркости: Функция монитора

- Монитор выполняет «аналоговое» расширение диапазона яркостей, сжимая темные области
- Тем самым увеличивается «плотность» градаций, оставляя разницу в пределах 1-2%



Сравнение линейного и гамма-кодирования



Кодирование яркости: примеры

На входе – неквантованное изображение



Задача: сохранить его в формате: 24 бит на пиксель

Кодирование яркости: примеры



Кодирование яркости: примеры

Вариант 1: линейная дискретизация



(показано БЕЗ гамма-коррекции!)

Кодирование яркости: примеры

Вариант 1: линейная дискретизация



(показано с гамма-коррекцией!)

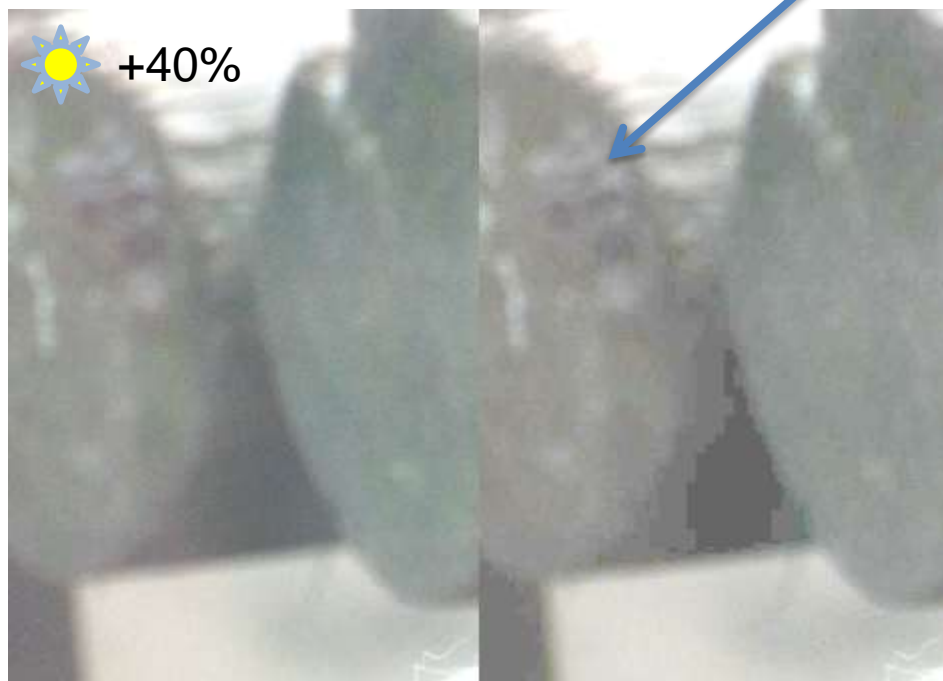
Кодирование яркости: примеры

Вариант 2: нелинейная дискретизация



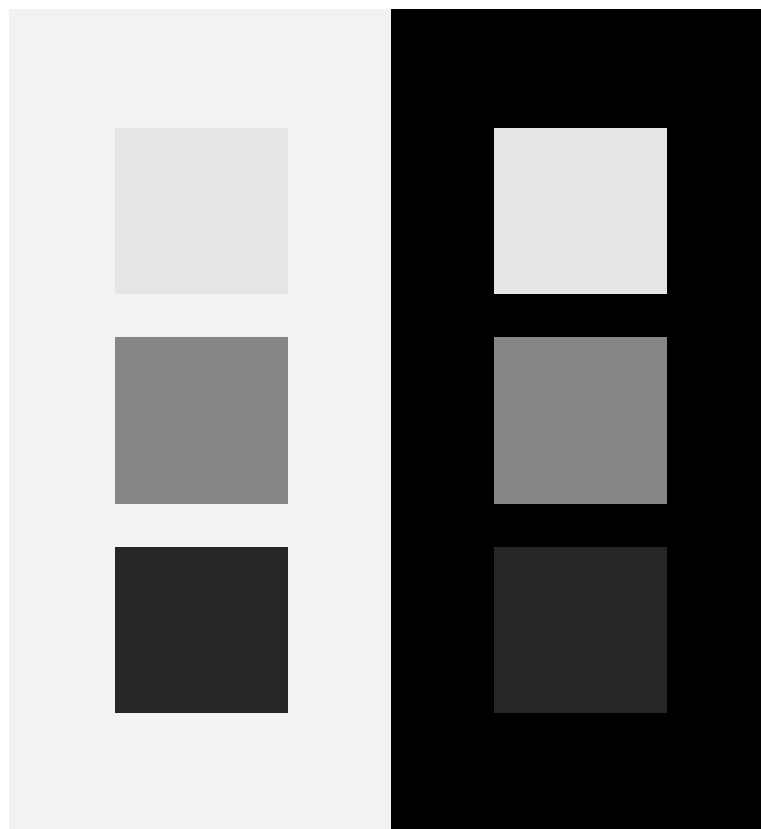
(Гамма-коррекция сразу в файле)

Кодирование яркости: сравнение



Причина 3: Одновременный контраст

- Зрительная система человека адаптируется к уровню окружающего освещения
- При ярком свете контраст повышается
 - черное кажется контрастно черным
- В темноте контраст понижается
 - черное становится серым



Одновременный контраст

- Можно использовать системы с суммарной передающей функцией с $\gamma \neq 1.0$, чтобы компенсировать разные освещения при создании и при просмотре изображения!
- $\gamma > 1.0$ больше контраст
- $\gamma < 1.0$ меньше контраст

Одновременный контраст: применение в видео

- Используется в видео
- «Недокоррекция»: предполагается $\gamma=2.5$, а корректируется 2.2
- Результирующая $\gamma \approx 1.1$
 - изображение более контрастно, что правильно при просмотре в темном окружении

$\gamma=2.2$ (результатирующая 1.0)



Предполагаем $\gamma=2.5$
(результатирующая 1.1)



Предполагаем $\gamma=2.8$
(результатирующая 1.27)



Гамма в мониторах и операционных системах

- Win32
- Apple (Mac)
- Варианты работы с гаммой:
 - Нет коррекции
 - должно делать приложение
 - Полная коррекция
 - приложение отдает линейное изображение
 - Частичная коррекция
 - частично корректирует драйвер, частично – приложение

Гамма в Windows

- Гамма не корректируется в драйвере!
- Для того, чтобы корректно вывести изображение на монитор, оно должно быть в нелинейном пространстве, с полной гамма-коррекцией
- С камеры уже приходят такие изображения, так что с ними ничего делать не нужно
 - При условии совпадения гаммы!
 - Гамма может быть в профиле (например, sRGB)
- Если ничего не сказано, используйте гамму 2.2
 - Прописано в sRGB

Примеры изображений!

Без гамма-коррекции



Гамма 1.8



Примеры изображений!

Без гамма-коррекции



Гамма 2.2



Примеры изображений!

Без гамма-коррекции



Гамма 2.5



Примеры изображений



Еще пример

Гамма-коррекция 1.0 (нет)



Гамма-коррекция 1.7



Гамма в Apple до Mac OS X 10.6 Snow Leopard в 2009

- Мониторы имеют те же ~ 2.5 гамма, что в Windows
- Графическая карта выполняет «недокоррекцию» гаммы самостоятельно
 - $C = C^1 / 1.45$
- Суммарно считается, что гамма на платформе Apple равна 1.8 (1.4 / 2.5)
- Изображения, сделанные на «старой» платформе Apple и не содержащие профиля, на платформе Windows будут выглядеть темными!

Гамма в Apple сейчас

- 2.2
- Как и в Windows

Как узнать, какая гамма у монитора???

- 1) померять вручную
- 2) считать ее равной 2.2

- Обычно гамма у CRT-монитора лежит в пределах 2.35-2.55

- У LCD должно быть так же, но в современных стараются делать 2.2

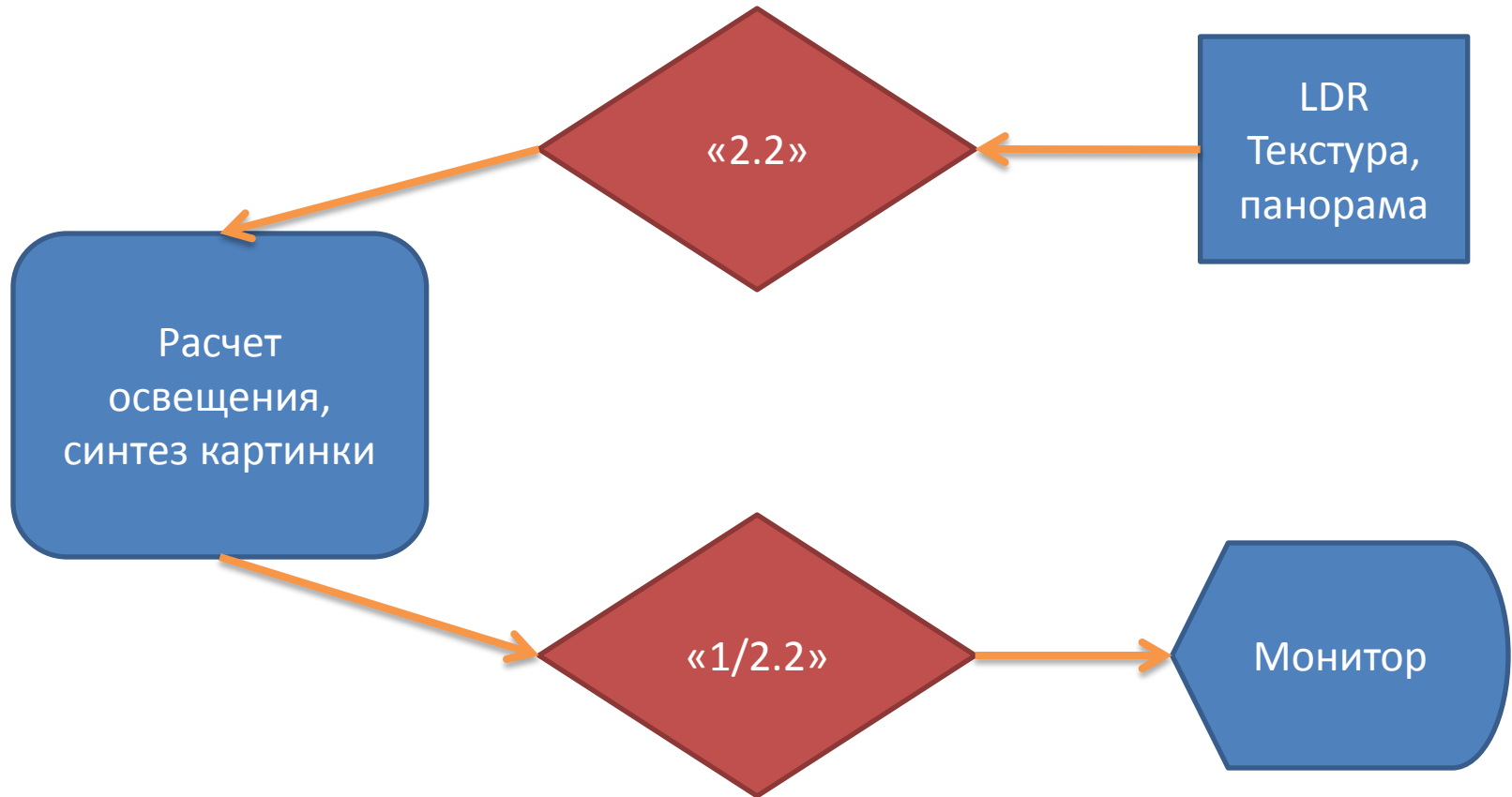
Как работать с гаммой

- Варианты использования изображений для синтеза фотореалистичных изображений:
 - Текстуры
 - Источники света (панорамы и т.п.)
- Алгоритмы, работающие с изображениями, должны использовать линейное представление излучения

Как понять, какая гамма в изображении?

- Если есть профиль, часто гамма указывается с профилем
- 99% процентов обычных изображений – JPEG, BMP и т.п. с гамма-коррекцией
- HDR-изображения без гамма-коррекции
- На платформе Windows это гамма 2.2. На других платформах может отличаться.
- Если в изображении нет профиля или просмотрщик не поддерживает профили, изображение может выглядеть неправильно!

Процесс использования изображений в процессе синтеза



Пример: работа с текстурами без и с гамма-коррекцией!

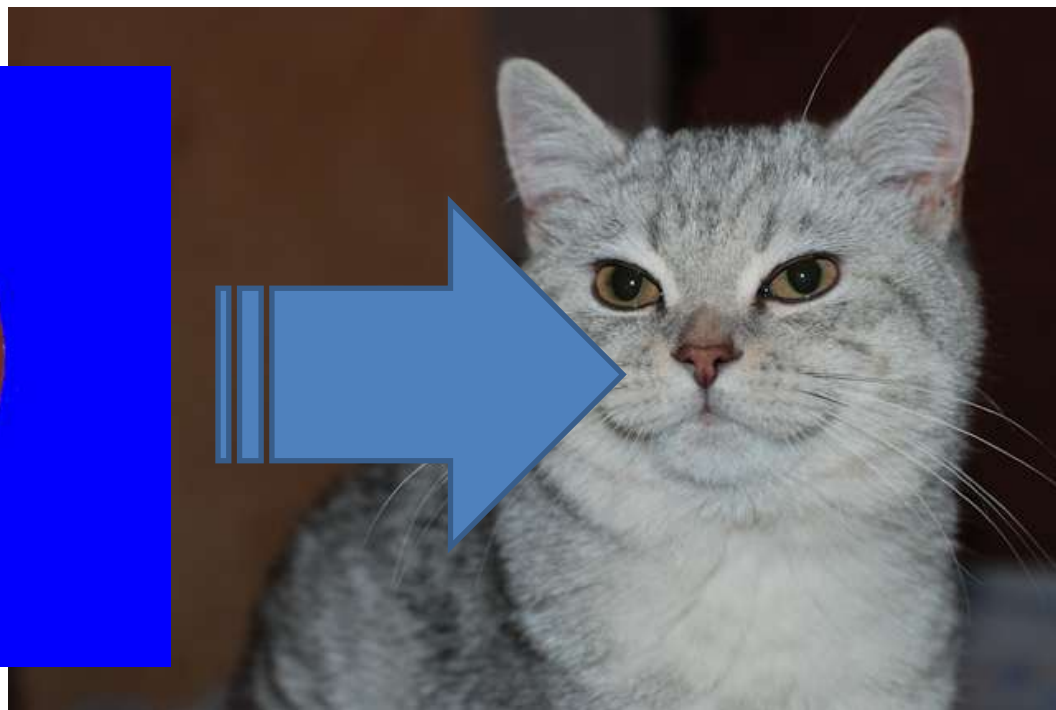
- Изображение 1 (JPEG): 200
- Изображение 2 (JPEG) : 50.
- Нужно наложить поверх с прозрачностью 0.4.
- Цвет = $200 * 0.4 + 50 (1 - 0.4) = 80 + 30 = 110$

Пример: работа с текстурами без и с гамма-коррекцией!

- Изображение 1 (JPEG, sRGB): 200
- Изображение 2 (JPEG, sRGB) : 50.
- Делаем гамма-преобразование, чтобы получить корректные значения излучения для данного источника.
- $(200 / 255) ^{2.2} = 149,4$ (округлено)
- $(50 / 255) ^{2.2} = 7,0$ (округлено)
- Цвет = $149,4 * 0.4 + 7,0 (1 - 0.4) = 92,5$
- Перед выводом делаем опять гамма-коррекцию:
- $(92,5 / 255)^{(1/2.2)} = 160,8151$!!!
- VS 110 !

Пример: смешивание изображений

50%





50% без гамма-преобразования



50%, с гамма-преобразованием

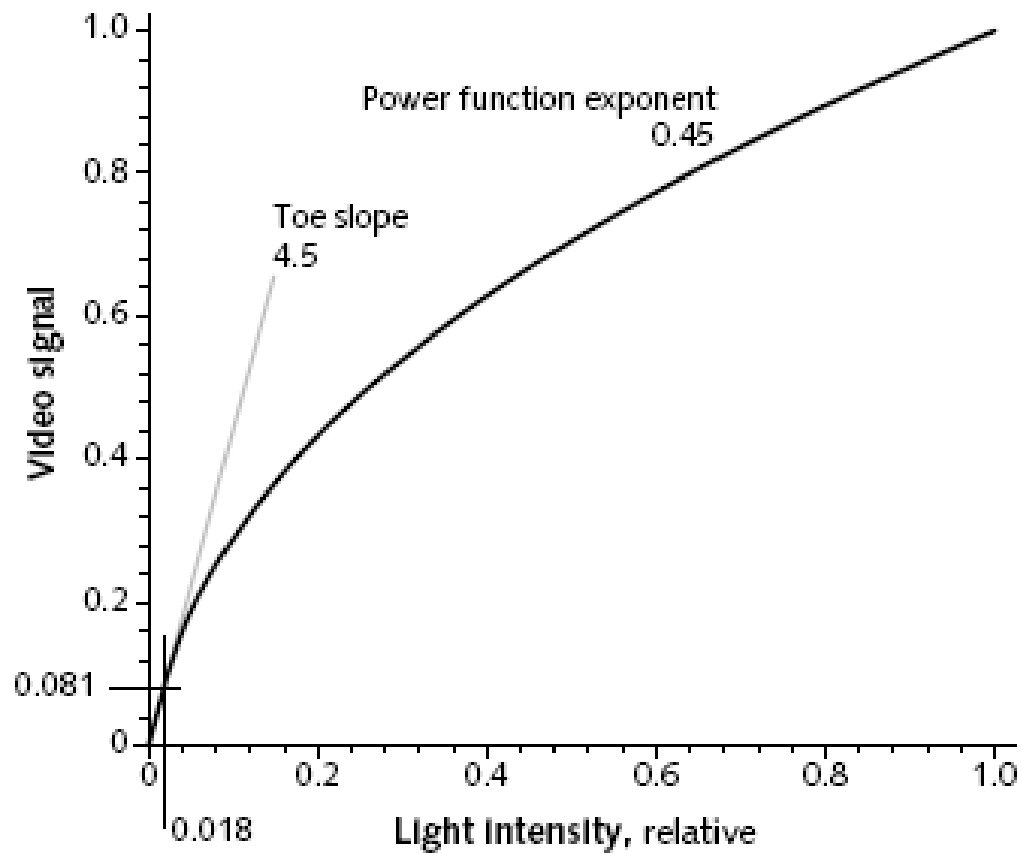
А почему там шум какой-то вылез?

- Потому что произошла потеря точности из-за преобразования
- $C = (\text{БУТЕ})C^{(2.2)}$
- ...
- $C = (\text{БУТЕ})C^{(1/2.2)}$

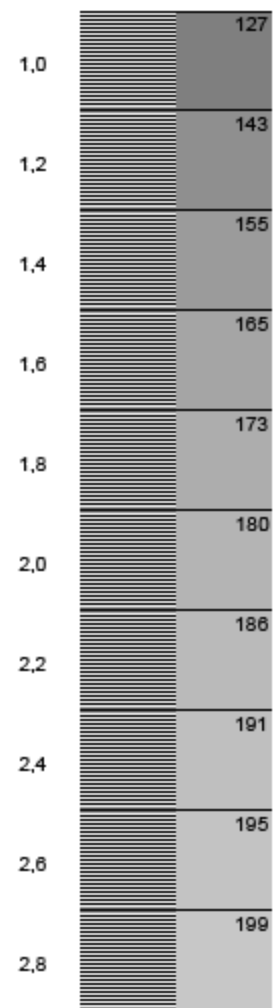
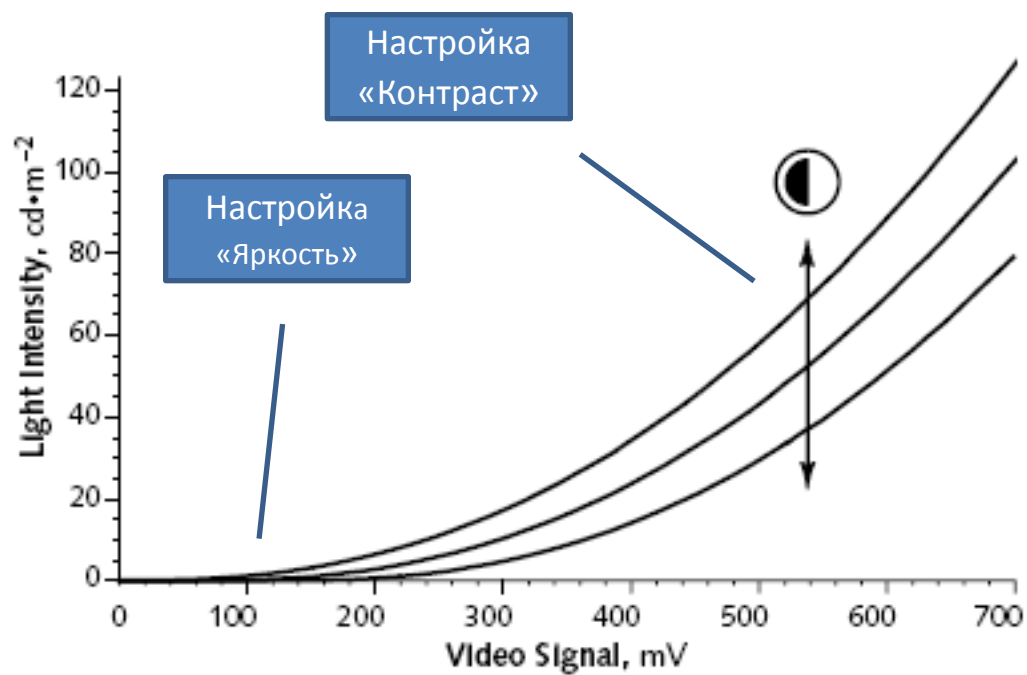


Вывод: если учитывать гамму, надо работать в floating-point или сразу готовить текстуры / освещение в линейном диапазоне

Прямая около нуля



Определение гаммы



Итоги

- Три причины для гамма-коррекции:
 - Компенсация нелинейности монитора
 - Однородное для восприятия кодирование
 - Учет окружения
- Гамма-коррекция в разных операционных системах
 - 2.2 – Windows
 - 1.8 - Mac до 2009, 2.2 - после
 - Профили!
- Применении гамма-коррекции в синтезе изображений
 - Во время синтеза работаем в линейном пространстве
 - Текстуры с коррекцией- это надо учитывать
 - После синтеза создаем floating-point результат
 - Применяем коррекцию

В следующий раз - HDR

- Получение HDR-изображений
- Кодирование HDR-изображений
- Преобразование HDR -> LDR (tone-mapping)