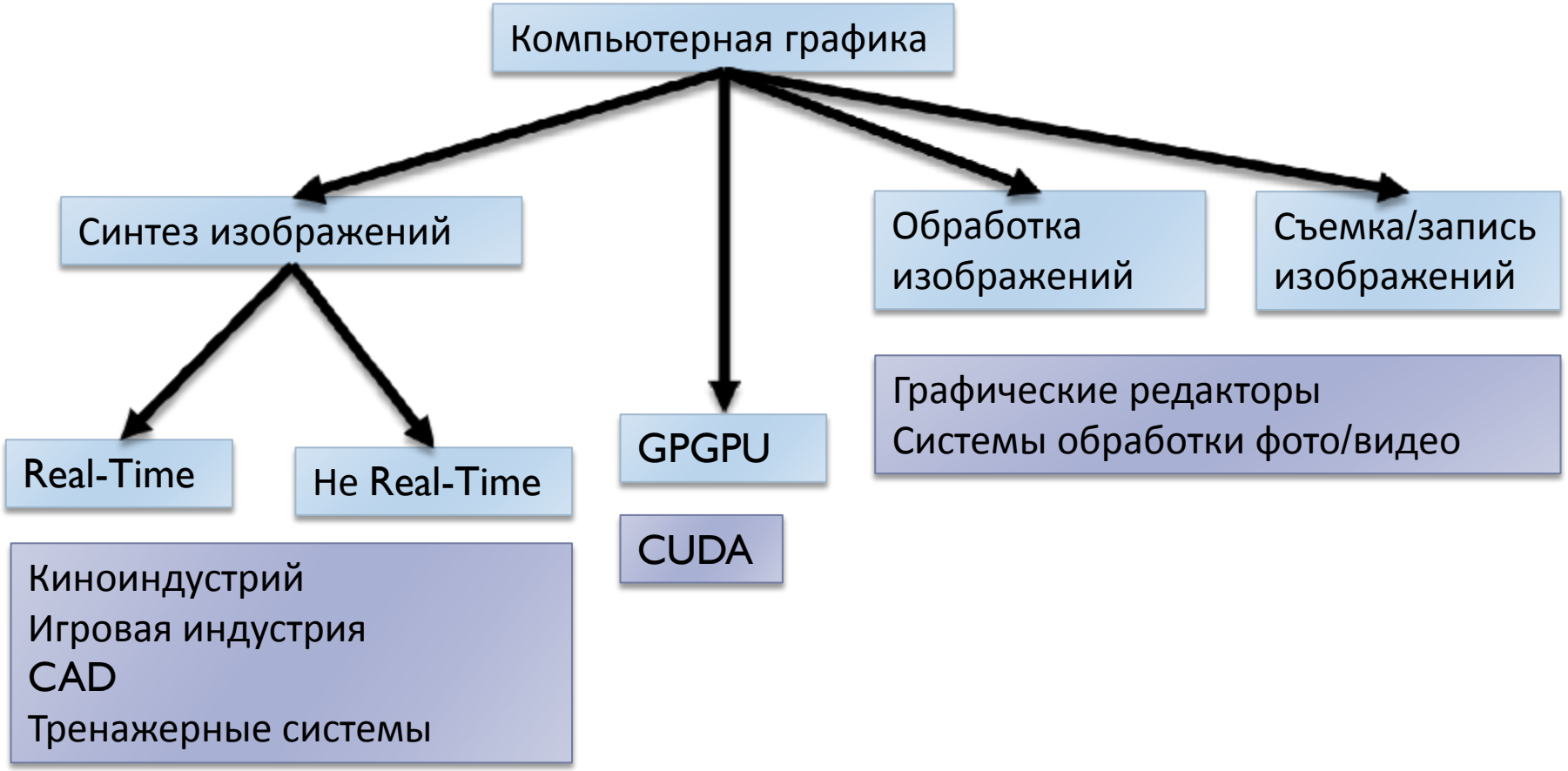
Компьютерная графика и визуализация в реальном времени

Лекция 0
Введение, виды синтеза изображений

Алексей Романов

Орг. вопросы

- Aleksei.A.Romanov@gmail.com – список группы с версиями OpenGL
  - Win - GPU caps viewer
  - *nix - glxinfo | grep -i opengl
- [cg_course_au_2017_autumn] – префикс темы письма
- Зачет
- Отчетность – сдача задач (4-6 штук)

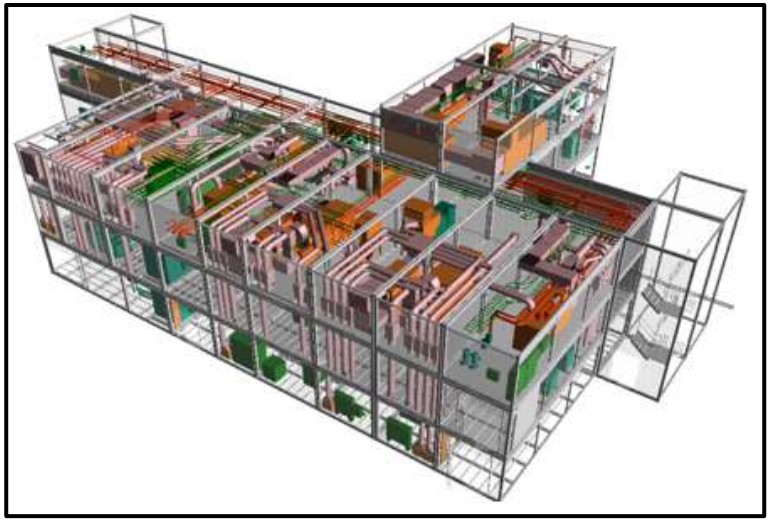

Компьютерная графика
Синтез изображений
Обработка изображений
Съемка/запись изображений
Real-Time
Не Real-Time
GPGPU
Графические редакторы
Системы обработки фото/видео
Киноиндустрий
Игровая индустрия
CAD
Тренажерные системы
CUDA

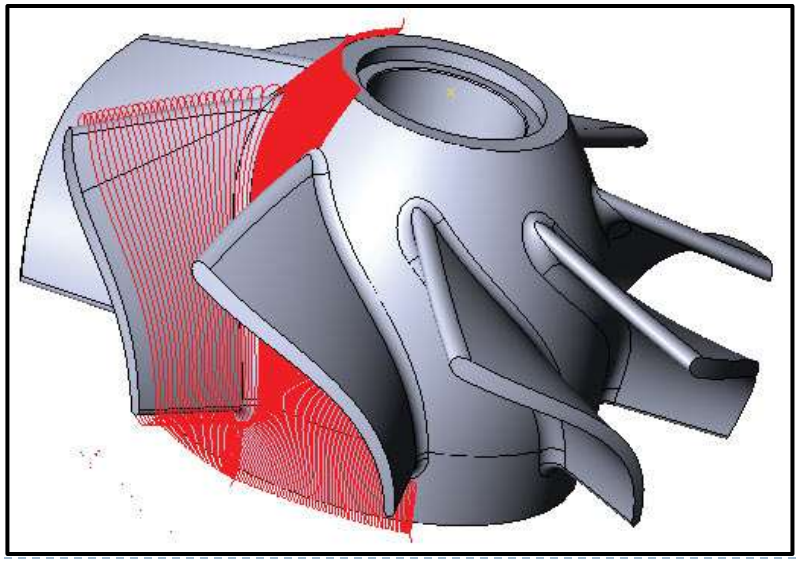
Киноиндустрия

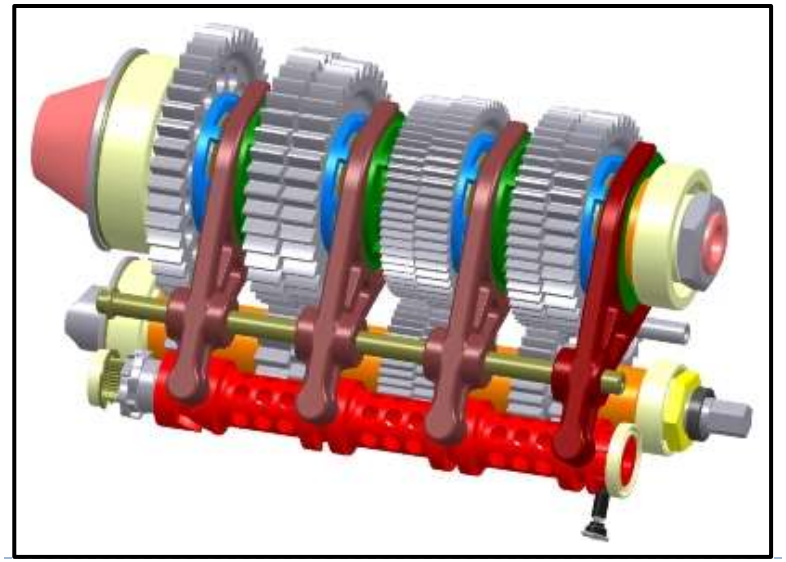
- Анимационные фильмы
- Спец-эффекты

Computer Aided Design (CAD)

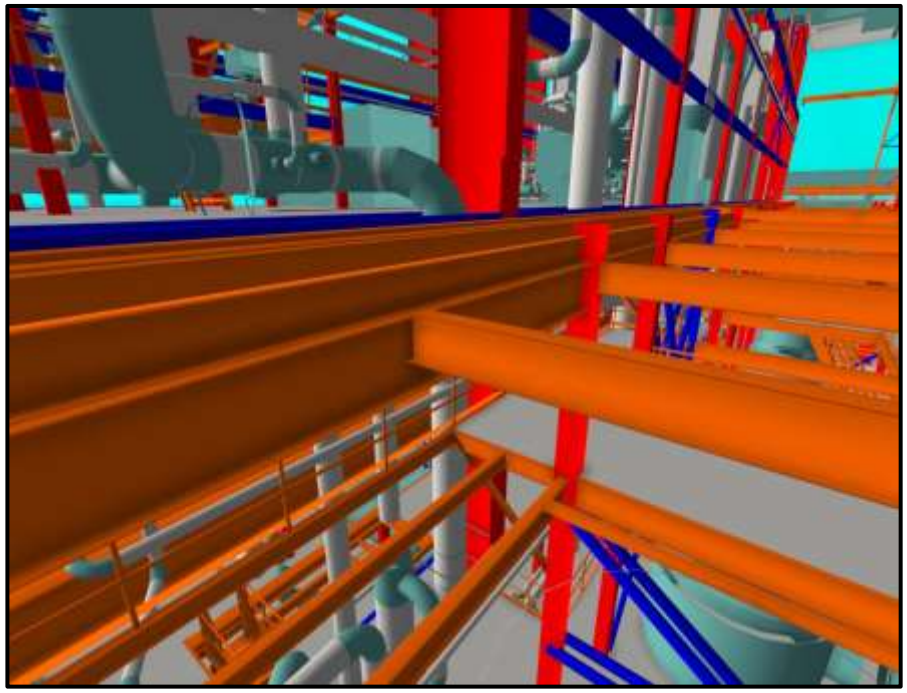
- Архитектура
- Машиностроение
- Сложность:
  - Огромное количество геометрии
  - Прозрачность большинства объектов

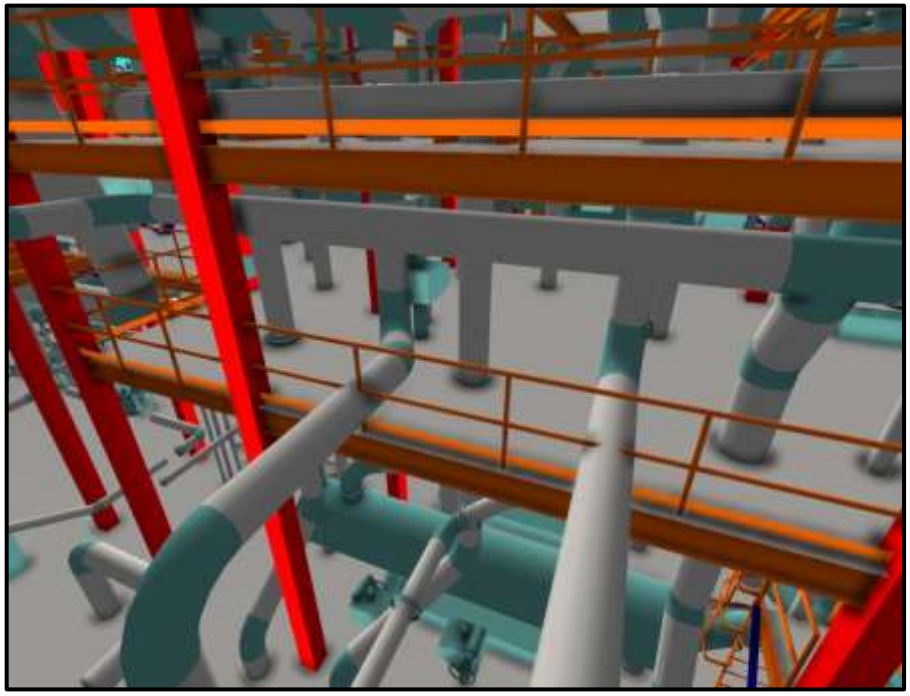






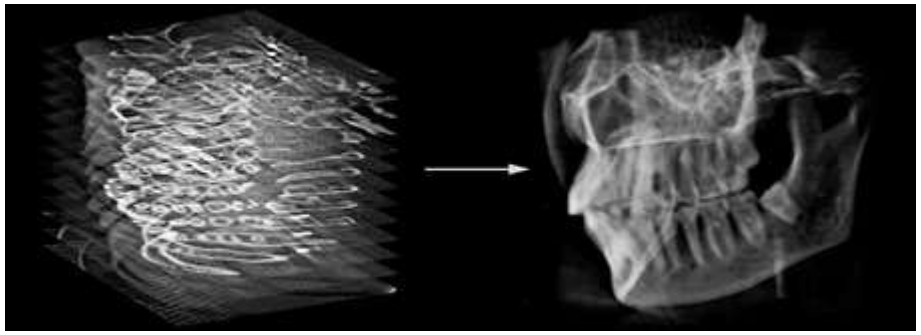
CAD





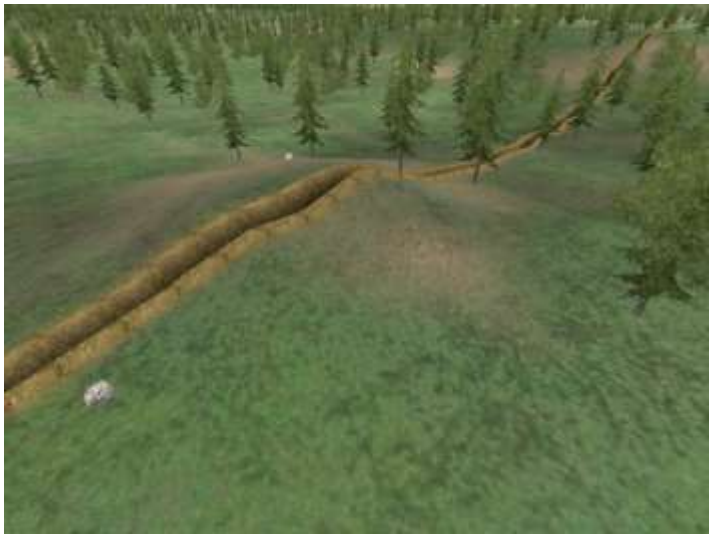
Медицинская визуализация

- Визуализация снимков КТ, МРТ
  - Большой объем данных
  - Неравноплотность прозрачности



Тренажерные системы

- Визуализация ландшафта
  - Большой размер поверхности ландшафта (до 100x100 км^2)
  - Произвольная степень детализации
  - Плавное изменение детализации
- Редактирование ландшафта
  - Изменение геометрии ландшафта в реальном времени
  - Редактирование контуров поверхностных материалов в реальном времени



Компьютерные игры

- Многомиллиардная индустрия
- Оказывает значительное влияние на развитие компьютерной графики
  - Разработка новых алгоритмов
  - Совершенствование аппаратуры (рост по «закону» Мура)

Обработка изображение

- Фильтрация изображений
- Задачи компьютерного зрения
  - Распознавание шаблонов
  - Отслеживание элементов изображения

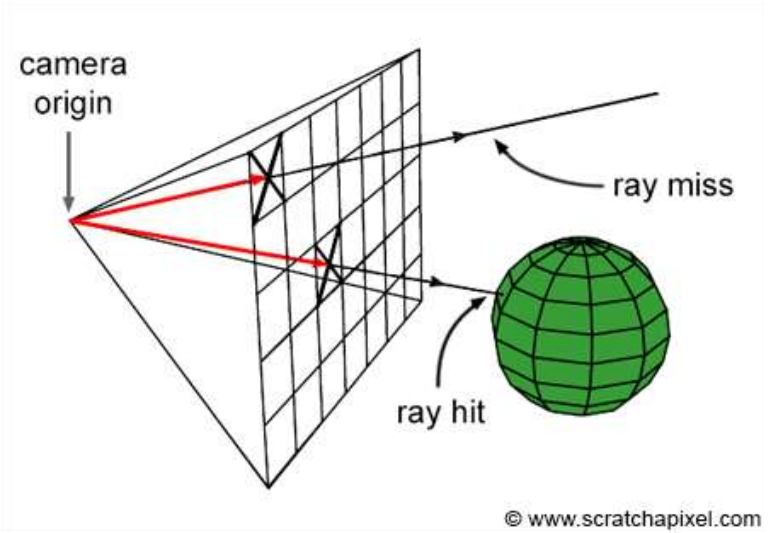
GPGPU (General Purpose GPU)

- Адаптация неграфических задач под реализацию на графическом процессоре
  - Классические алгоритмы: сортировки, n-статистика
  - FFT
  - Математические операции
- Неграфический API для параллельных вычислений на видеокарте
  - AMD FireStream
  - DirectCompute
  - CUDA
  - OpenCL
  - OpenGL (compute shaders)

Рост производительности в ряде случае до 100-300 раз

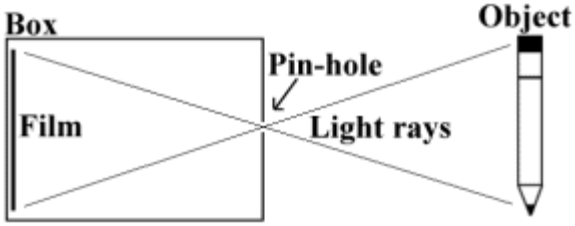
Синтез изображений

- Сцена
- Камера
- Алгоритм синтеза
  - Определение пересечения луча с объектом сцены
- Трассировка лучей
- Ray-Marching
- Rendering

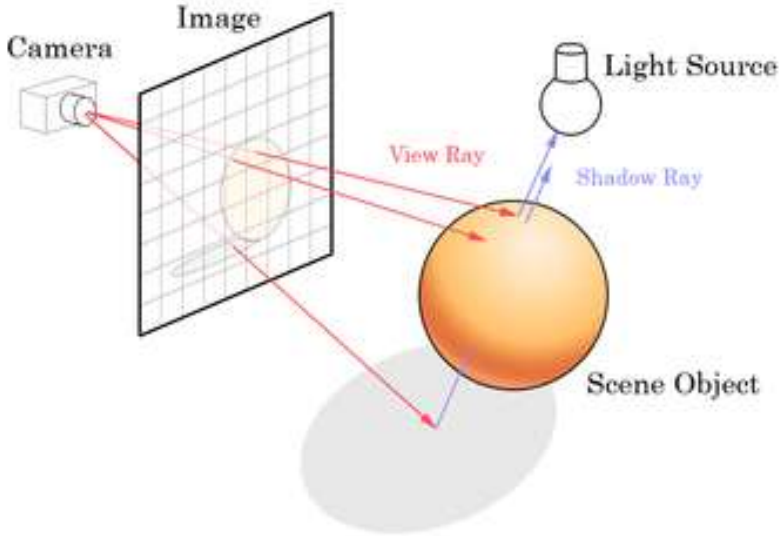


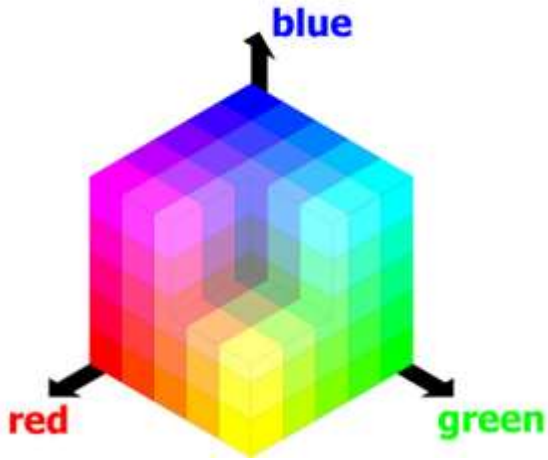
Модель камеры

- *pinhole camera (camera obscura)*



- Используемая камера

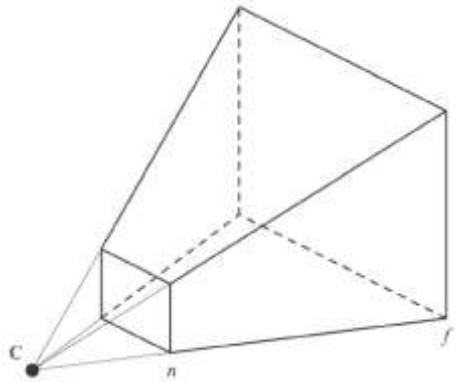




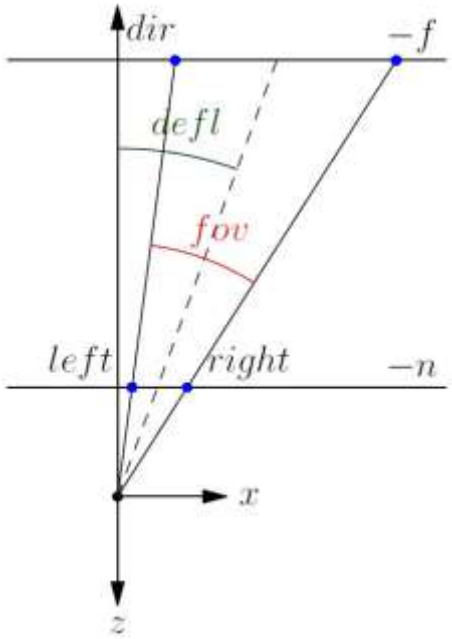
В RGB представимы не все цвета, воспринимаемые человеком

Пирамида видимости

Задается в систем координат камеры



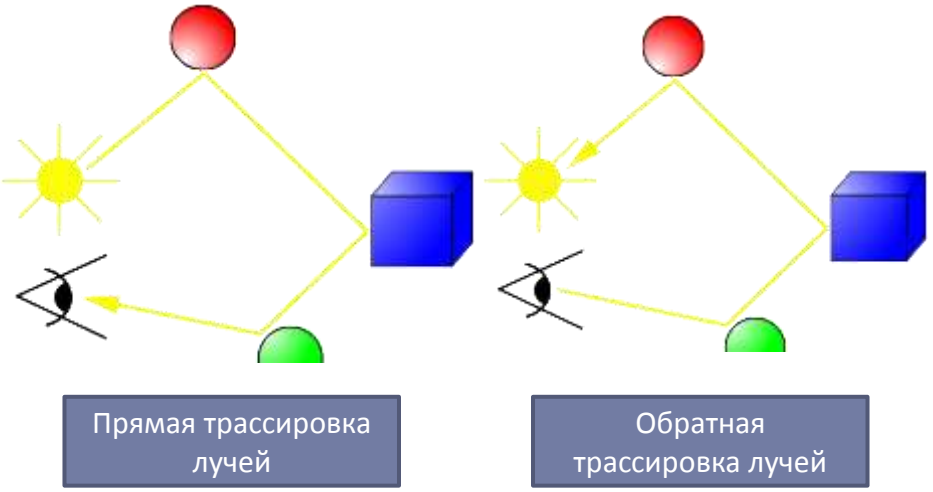
n (near)	Расстояние до ближней плоскости
f (far)	Расстояние до дальней плоскости
fov (field of view)	Угол обзора
defl (deflection)	Угол отклонения



Трассировка лучей

Изображение – результат аккумуляции на сенсоре лучей, выпущенных источниками света

- Прямая
  - Трассировка лучей из источников света
  - Лишь небольшая часть попадет в камеру
  - Еще меньшая даст хоть какой-либо значимый вклад в результирующий цвет
- Обратная
  - Трассировка из камеры
  - Будет учтен луч с наибольшим вкладом в результирующий цвет

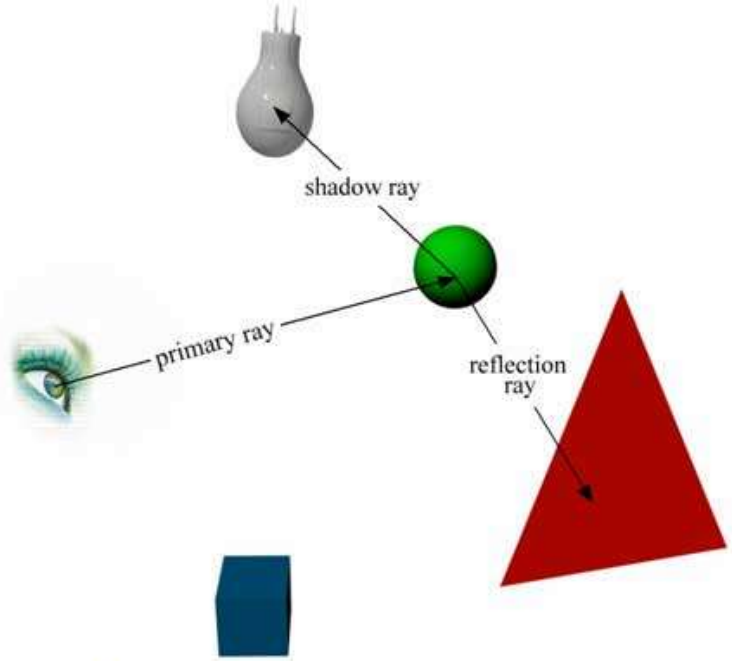


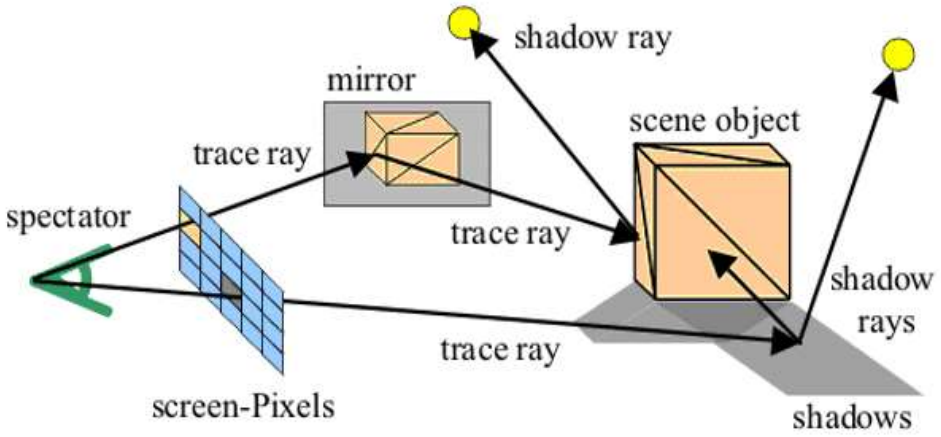
Сцена

- Объекты, задаваемые аналитически
  - Плоскость
  - Сфера
  - Полигон
  - Тор
  - Эллипсоид
  - …
- *CSG (constructive solid geometry)*

Классификация трассируемых лучей

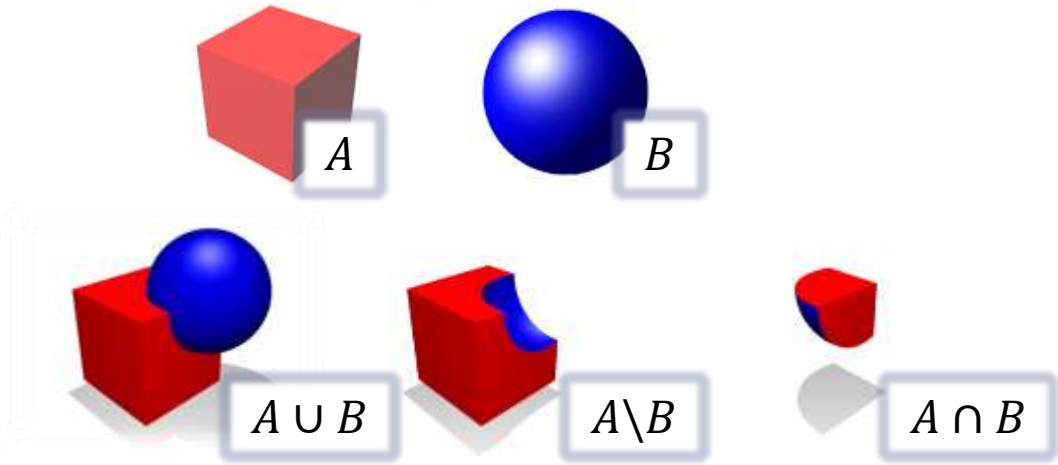
- *Pixel/Primary ray* – луч, попадающий на плоскость экрана
- *Illumination/Shadow ray* – луч, передающий освещение от источника света/объекта другому объекту
- *Trace ray – reflection & refraction*

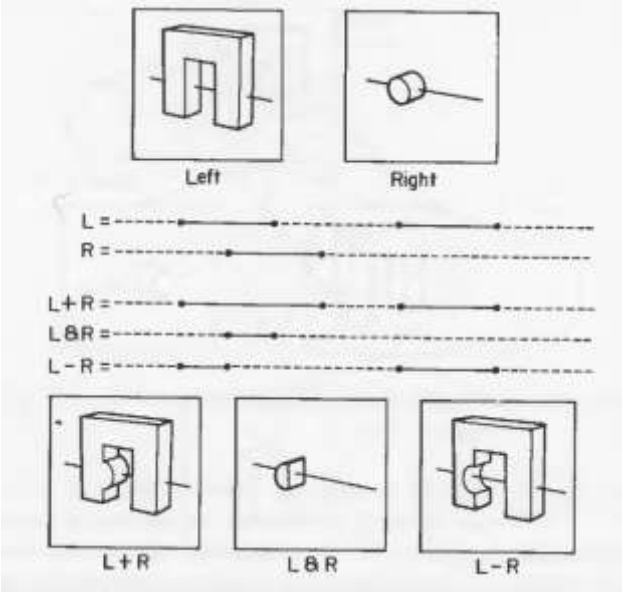




CSG (Constructive Solid Geometry)

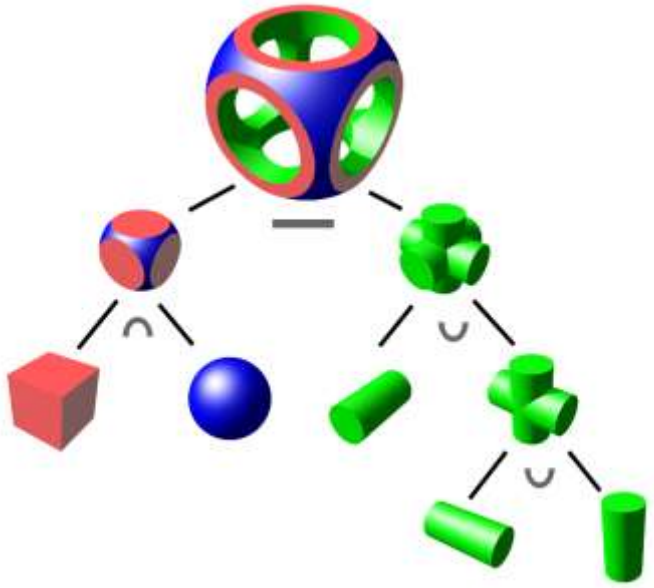
- Комбинация базовых выпуклых объектов, используя операции над множествами
  - Объединение
  - Разность
  - Пересечение
- Диаграмма Рота (*Roth diagram*)





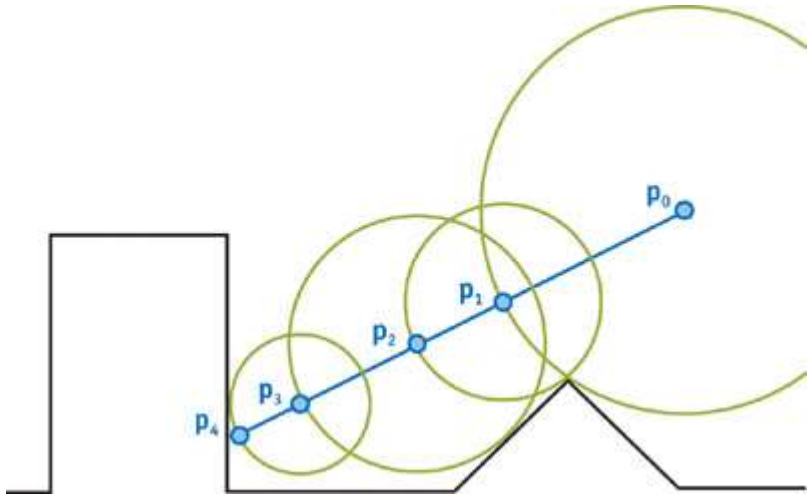
Дерево *CSG*

- Обход от узлов до корня
  - Дочерний узел выдает родительскому список сегментов
  - Родительский узел применяет соответствующую операцию, используя диаграммы Рота
- Могут содержать информацию по охватывающим объемам для ускорение визуализации
- Дополнительные оптимизации (*i.e.* $A = \emptyset \Rightarrow A \backslash \mathrm{B} = \emptyset$), позволяют уменьшить количество расчетов пересечений



Ray Marching

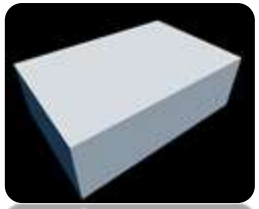
- Для сцены определена функция расстояния до ближайшей точки



Расстояние до объектов сцены

```
float sdSphere( vec3 p, float s )
{
  return length(p)-s;
}
```

```
float udBox( vec3 p, vec3 b )
{
  return length(max(abs(p)-b,0.0));
}
```



CSG via RayMarching

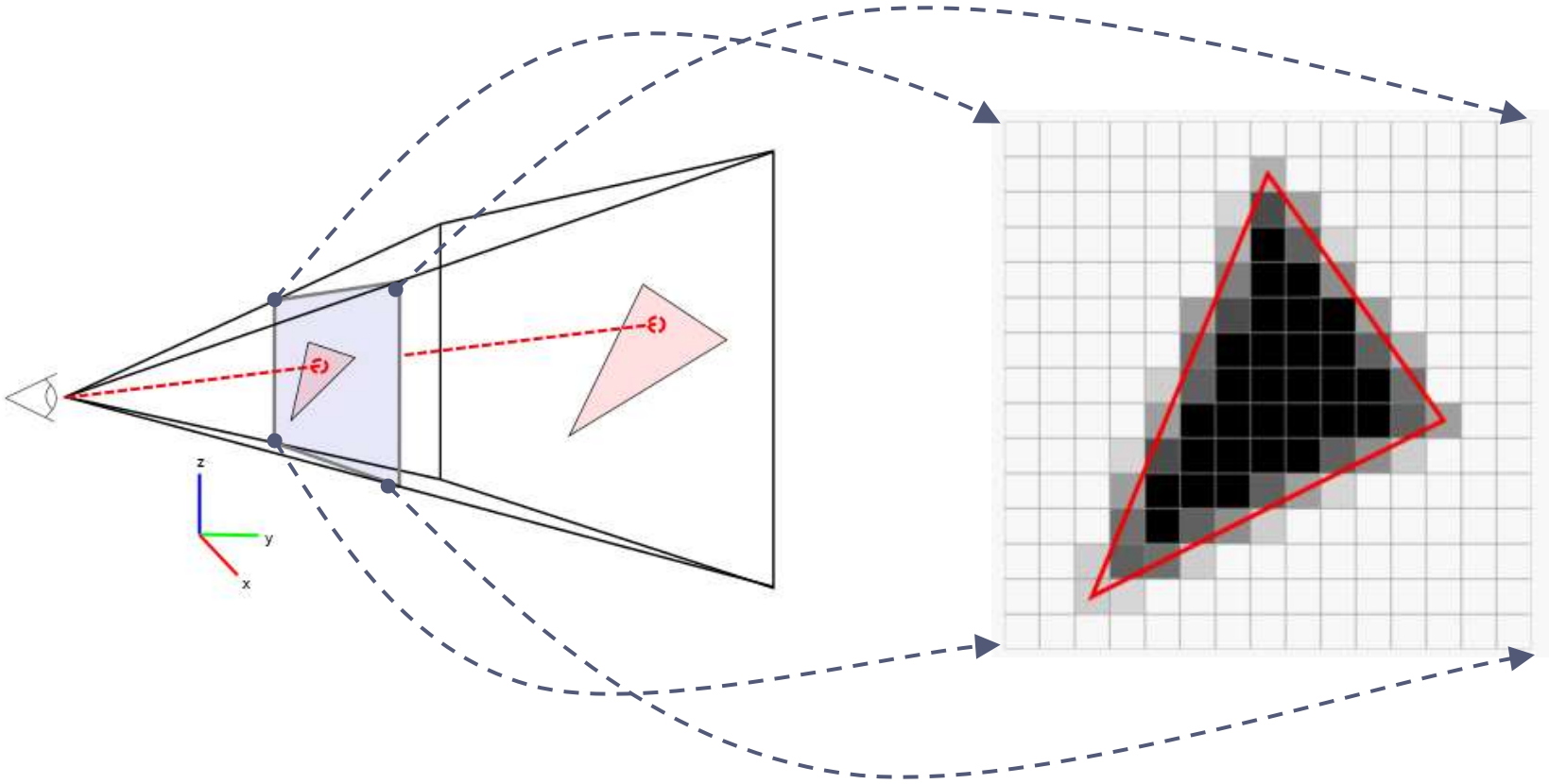
```
float intersectSDF(float distA, float distB) {
    return max(distA, distB);
}

float unionSDF(float distA, float distB) {
    return min(distA, distB);
}

float differenceSDF(float distA, float distB) {
    return max(distA, -distB);
}
```

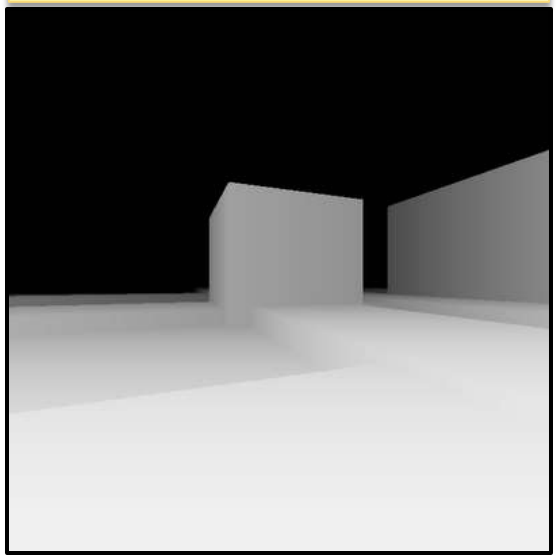
Rendering

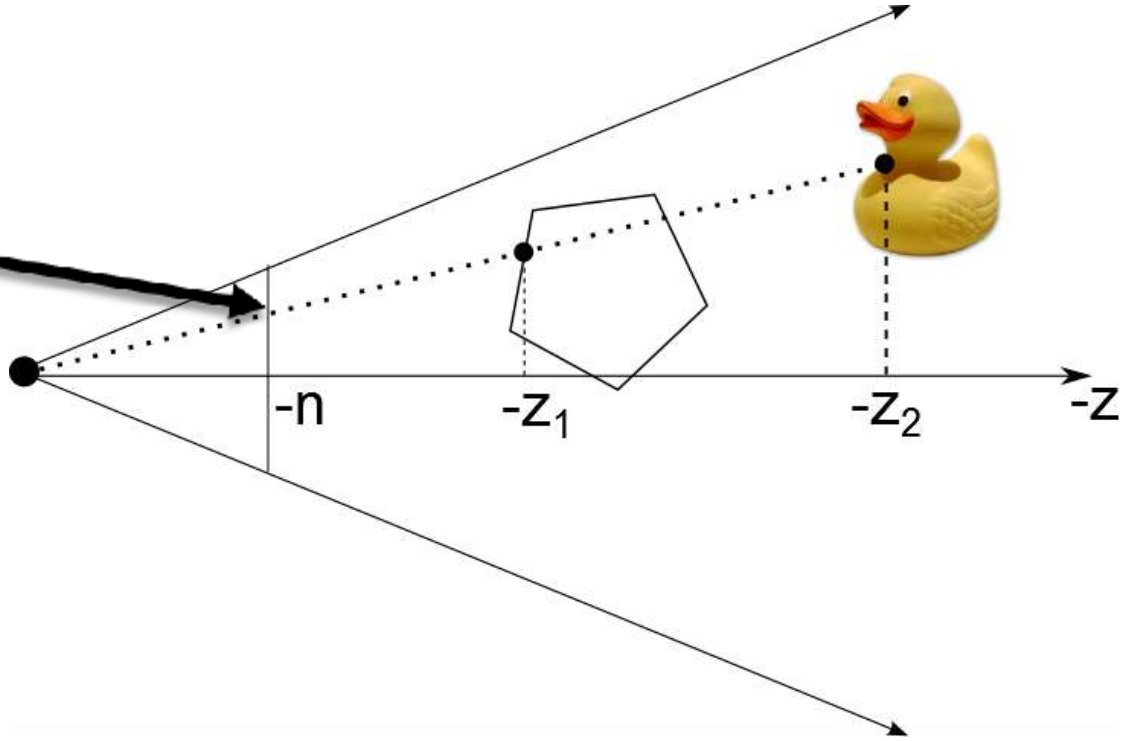
Проекция сцены на плоскость экрана с дальнейшей растеризацией



Удаление невидимых поверхностей, z тест (тест глубины)

Буфер глубины





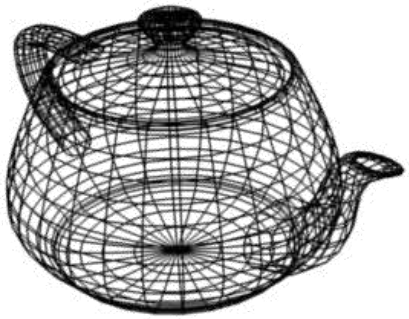
Rendering

Проекция сцены на плоскость экрана с дальнейшей растеризацией

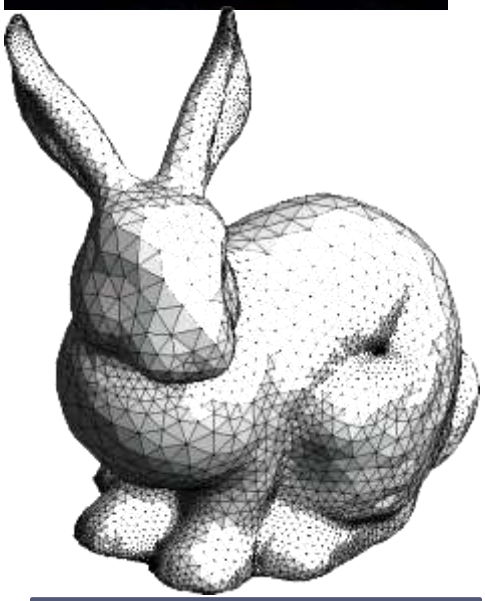
Сцена

Геометрия задается композицией графических примитивов

- Точки
- Линии
- Треугольники
- Четырехугольники



Utah teapot



Stanford bunny
1994

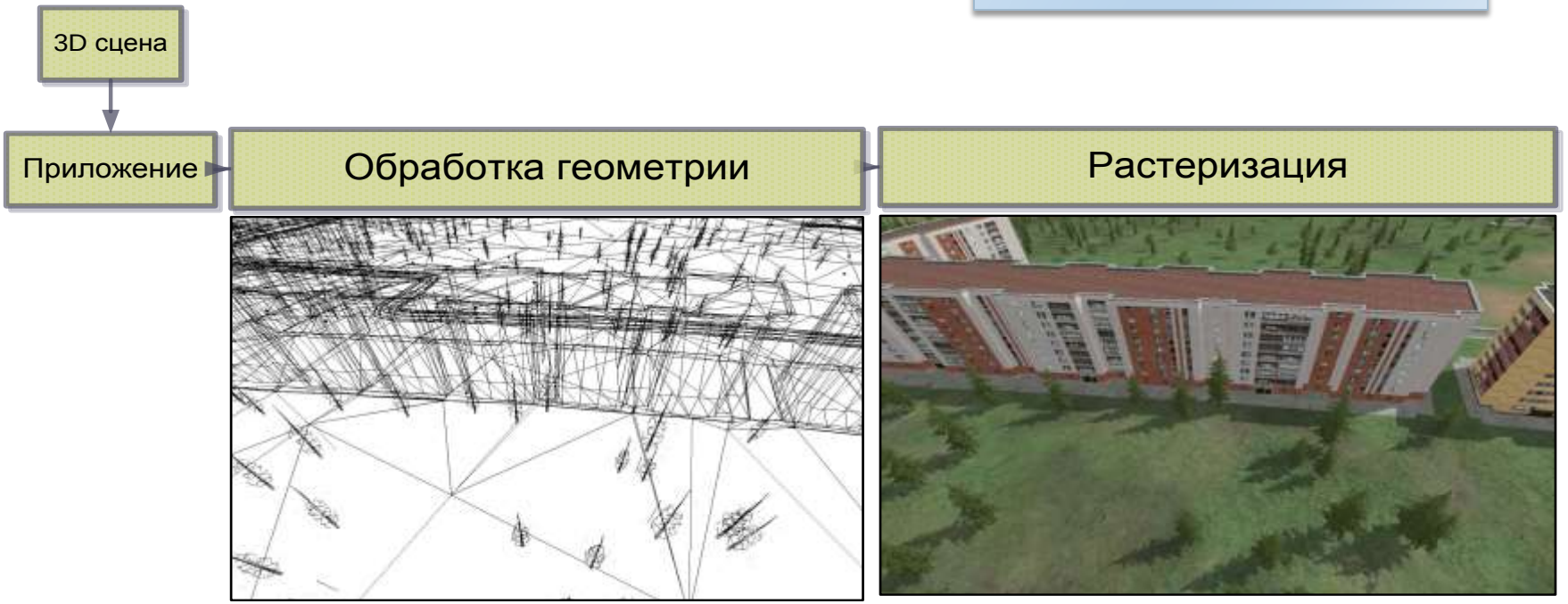
Графический конвейер

Фиксированная последовательность этапов, выполняемых на CPU/GPU для отображения сцены. Реализуется драйвером видеокарты.

Конвейер состоит из 3 частей:

- Вызовы команд визуализации
- Обработка геометрии
- Растеризация

Существуют 2 распространенных API, предоставляющих возможности графического конвейера

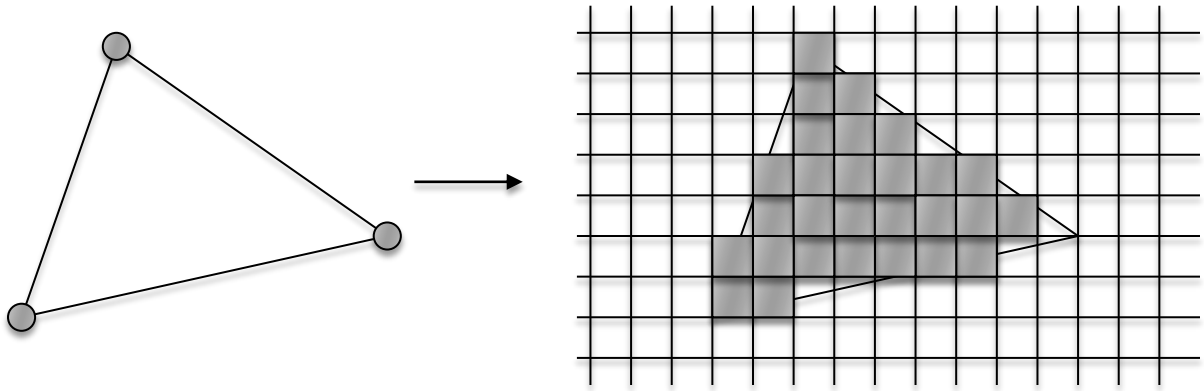


Графический конвейер

- Вызовы команд визуализации
  - Отводятся на самостоятельное изучение
  - На сайте курса будут выложены примеры по работе с OpenGL
- Обработка геометрии

  Процесс обработки вершинных атрибутов, проекции вершин на экран и генерации примитивов для растеризации
- Растеризация

  Процесс заливки пикселей, занимаемых проекцией геометрии с интерполяцией и применением вершинных атрибутов



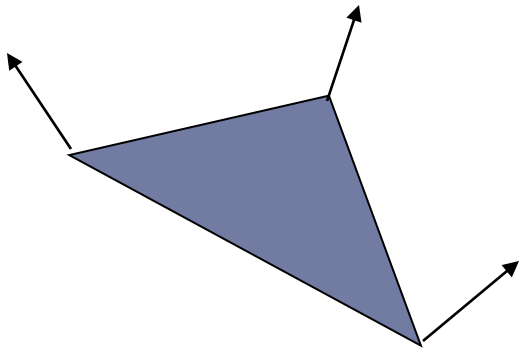
Графические API

- Стандарт OpenGL
  - Кроссплатформенность
  - Позволяет использовать новые возможности видеокарты сразу после выпуска драйверов
  - Тесно связан с OpenGLES и WebGL
  - C-подобный API
  - Минималистичный SDK
- Стандарт Vulkan
  - OpenGL on steroids
  - Non-friendly для начинающих
- Библиотека DirectX
  - Удобные интерфейсы
  - Стандартные вспомогательные библиотеки
  - Обширный SDK
  - Большое количество средств отладки
  - Только для Windows и Xbox

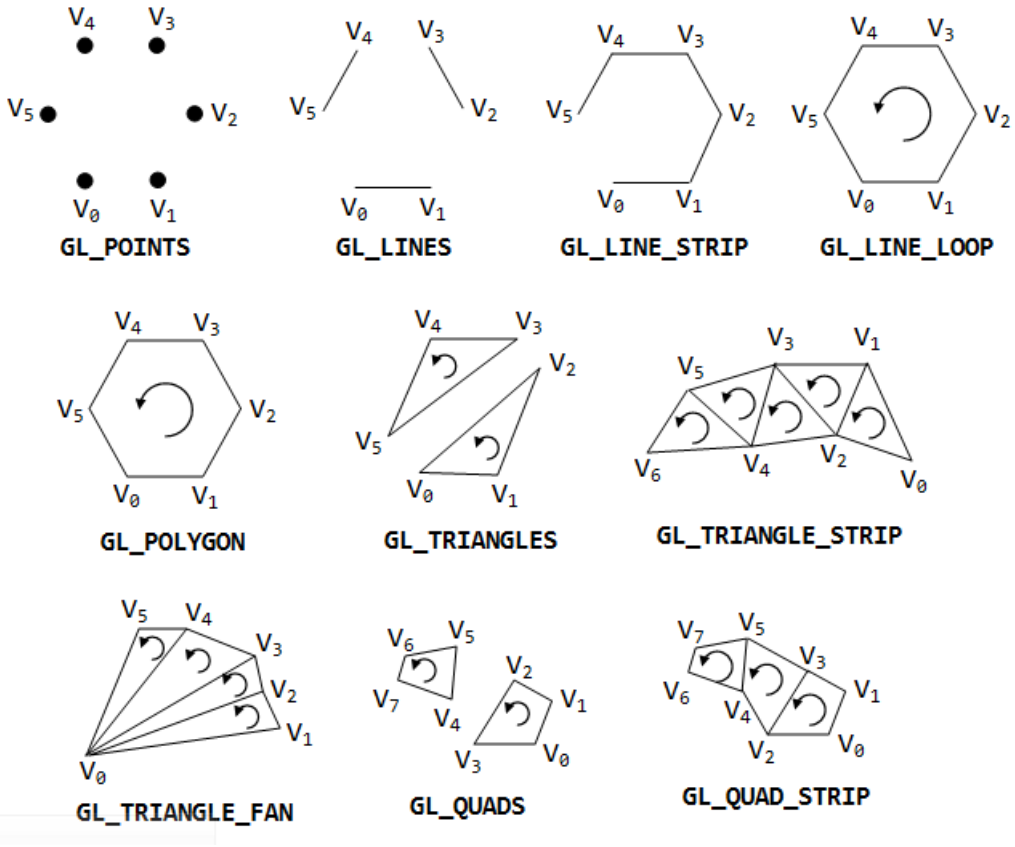
Мы будем изучать OpenGL

Геометрия в OpenGL

- Геометрия – набор вершин и индексов
- Вершина – набор атрибутов
- Типичные атрибуты
  - Позиция
  - Цвет
  - Нормаль
  - Текстурные координаты



Графические примитивы OpenGL



Гамма-коррекция

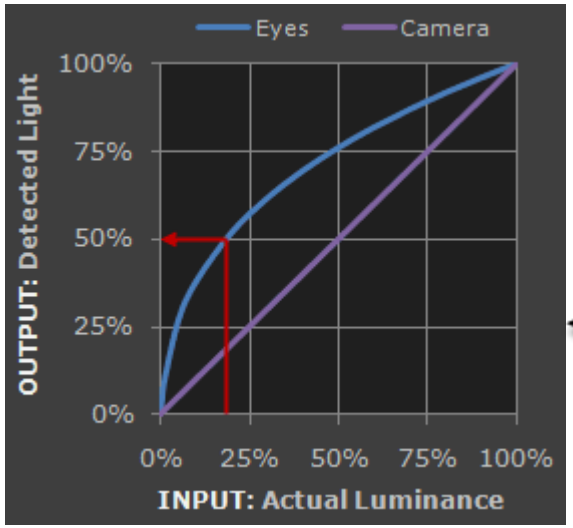
- $f(x) \sim x^{\gamma}$
- Как физическое явления
- Как перцептивное явление

Гамма-коррекция, физическое явление

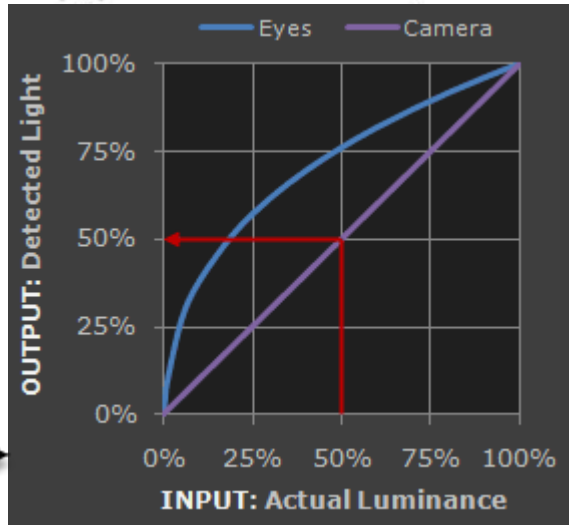
- Дисплей CRT нелинейно воспроизводит яркость от напряжения – нелинейность связана с оссобенностями электронной пушки, а не со свойствами люминофора
- Напряжение линейно зависит от величин, записиваемых в буфер кадра

Гамма-коррекция, перцептивное явление

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



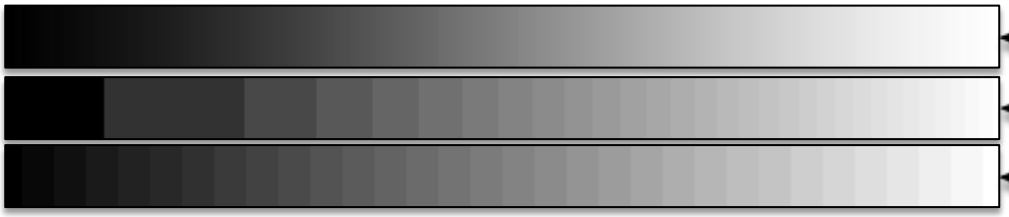
Исходное изображение



Восприимчивость глаза

Восприимчивость камеры

- Улучшает восприятие света низкой яркости
- Кодирование изображений



Исходный градиент

Линейное кодирование 8-bit

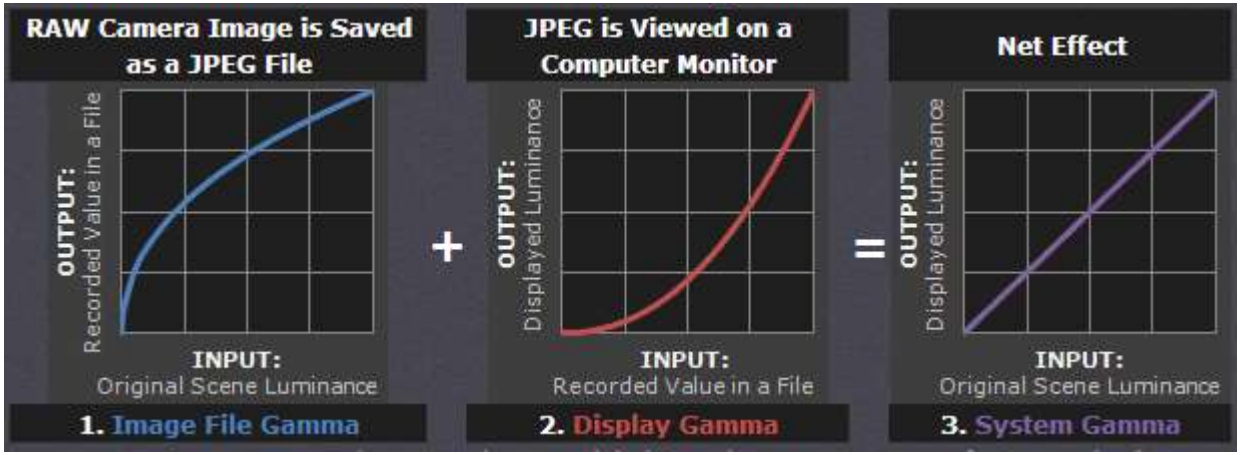
Гамма кодирование 8-bit

Другие нелинейные перцептивные характеристики

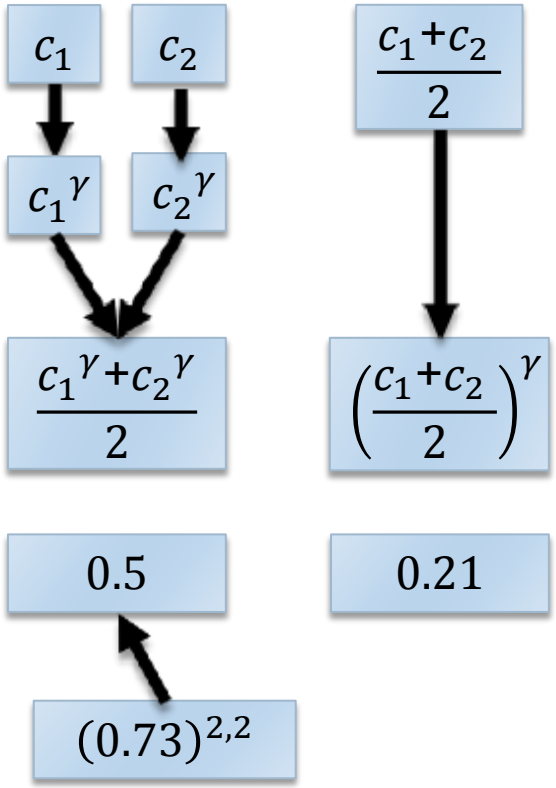
Перцептивная характеристика	Физическая величина	Степень
Громкость	Уровень звукового давления	0.67
Соленость	Концентрация хлорида натрия	1.4
Запах	Концентрация ароматических молекул	0.6
Тяжесть	Масса	1.45

Гамма-коррекция

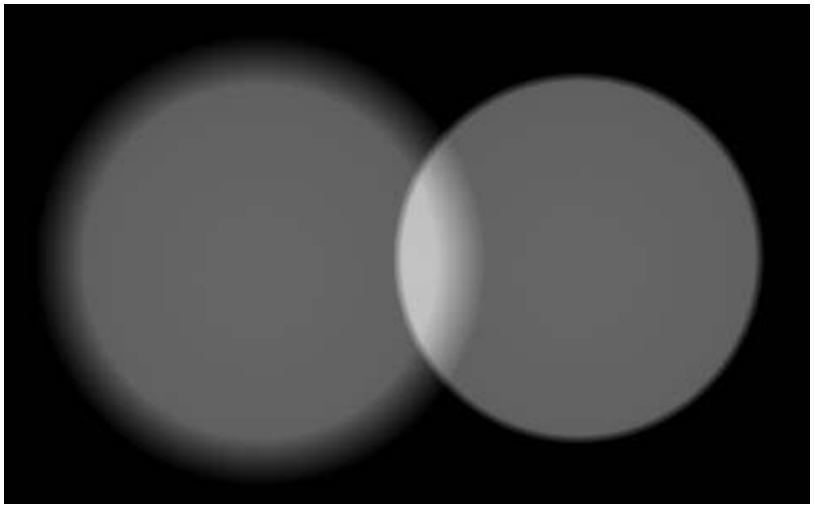
По счастливой случайности $\gamma_{display} \cong \frac{1}{\gamma_{eye}}$



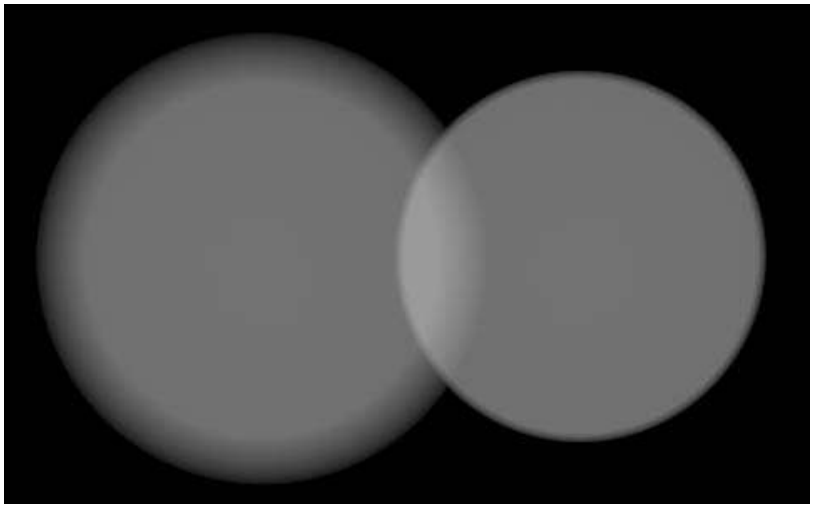
- $\gamma_{display} = \frac{1}{\gamma_{eye}} = \gamma = 2.2$



Гамма-корректное аддитивное смешивание цветов

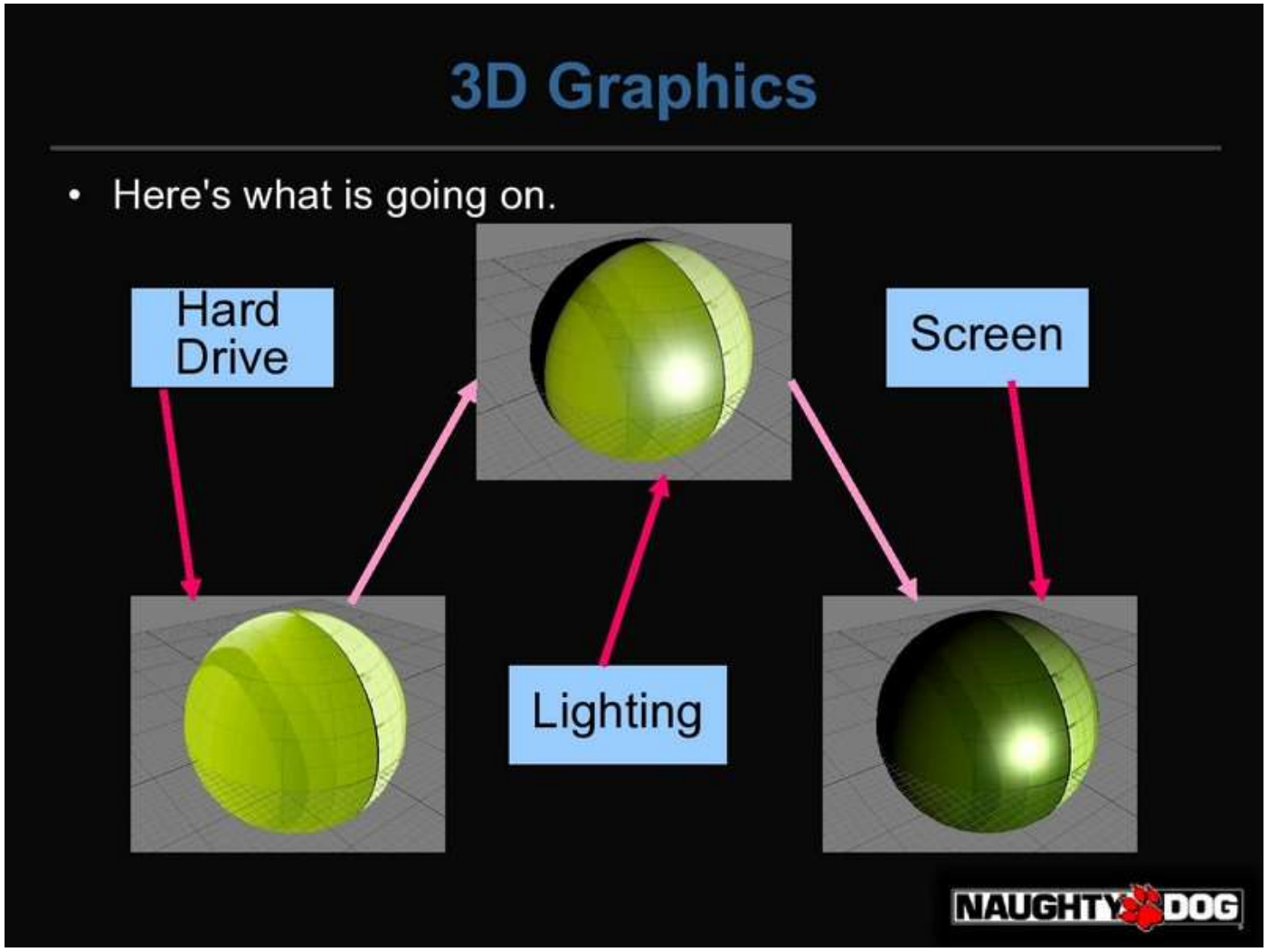


$c_1 + c_2$

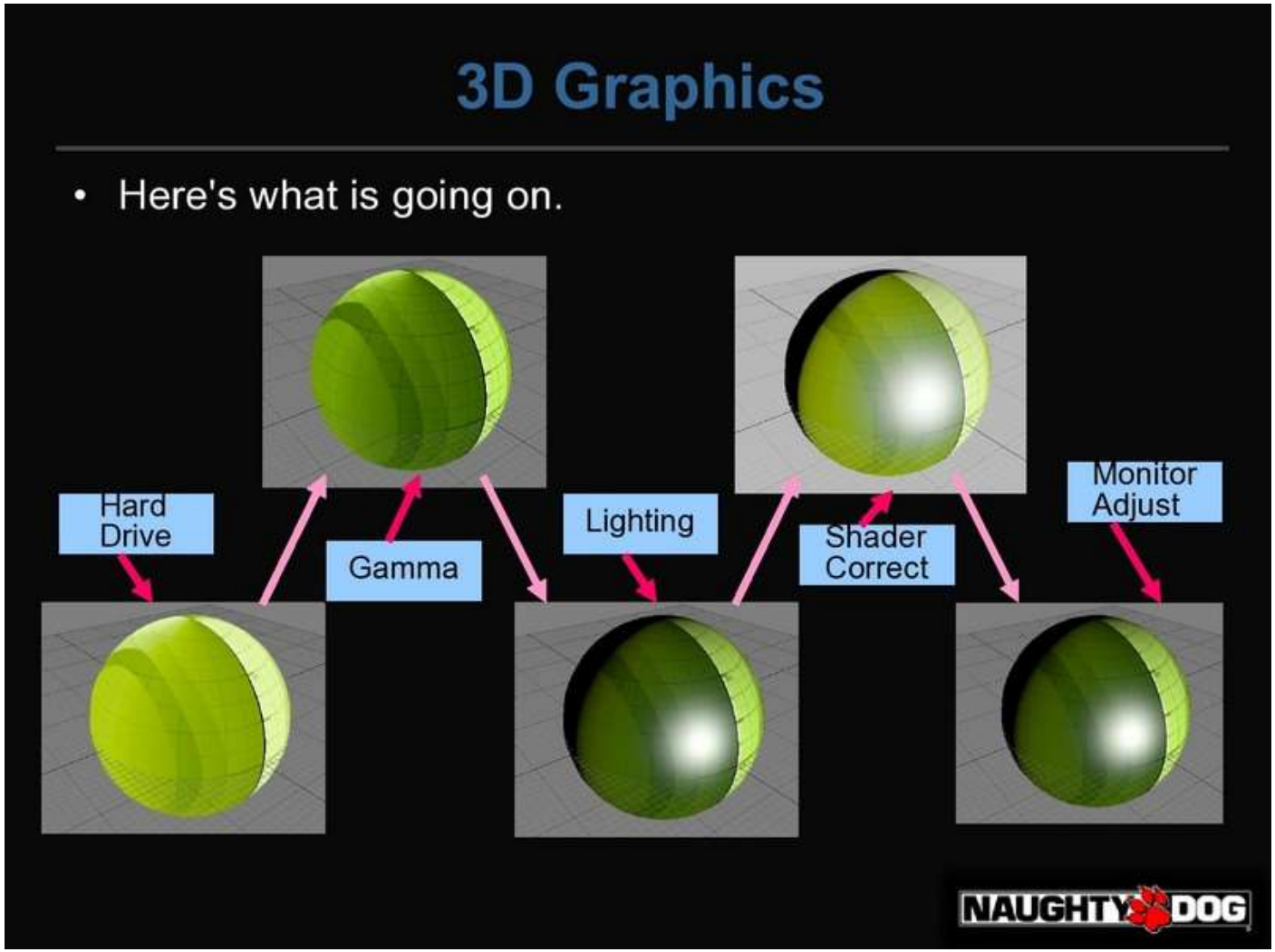


$\left(c_1^{\frac{1}{\gamma}} + c_2^{\frac{1}{\gamma}}\right)^{\gamma}$

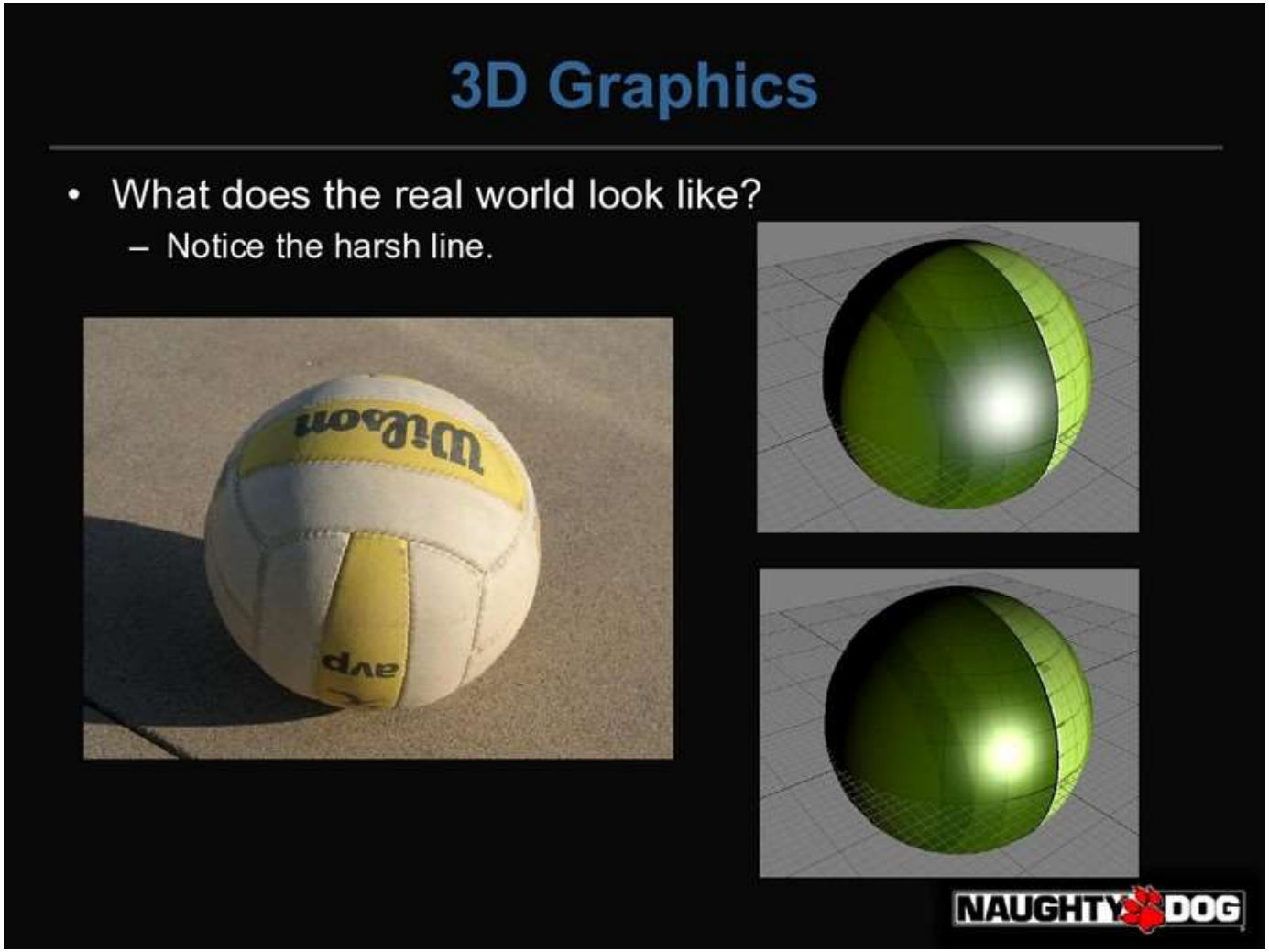
Не забываем про гамму коррекцию



Еще раз о гамма коррекции



Еще раз о гамма коррекции



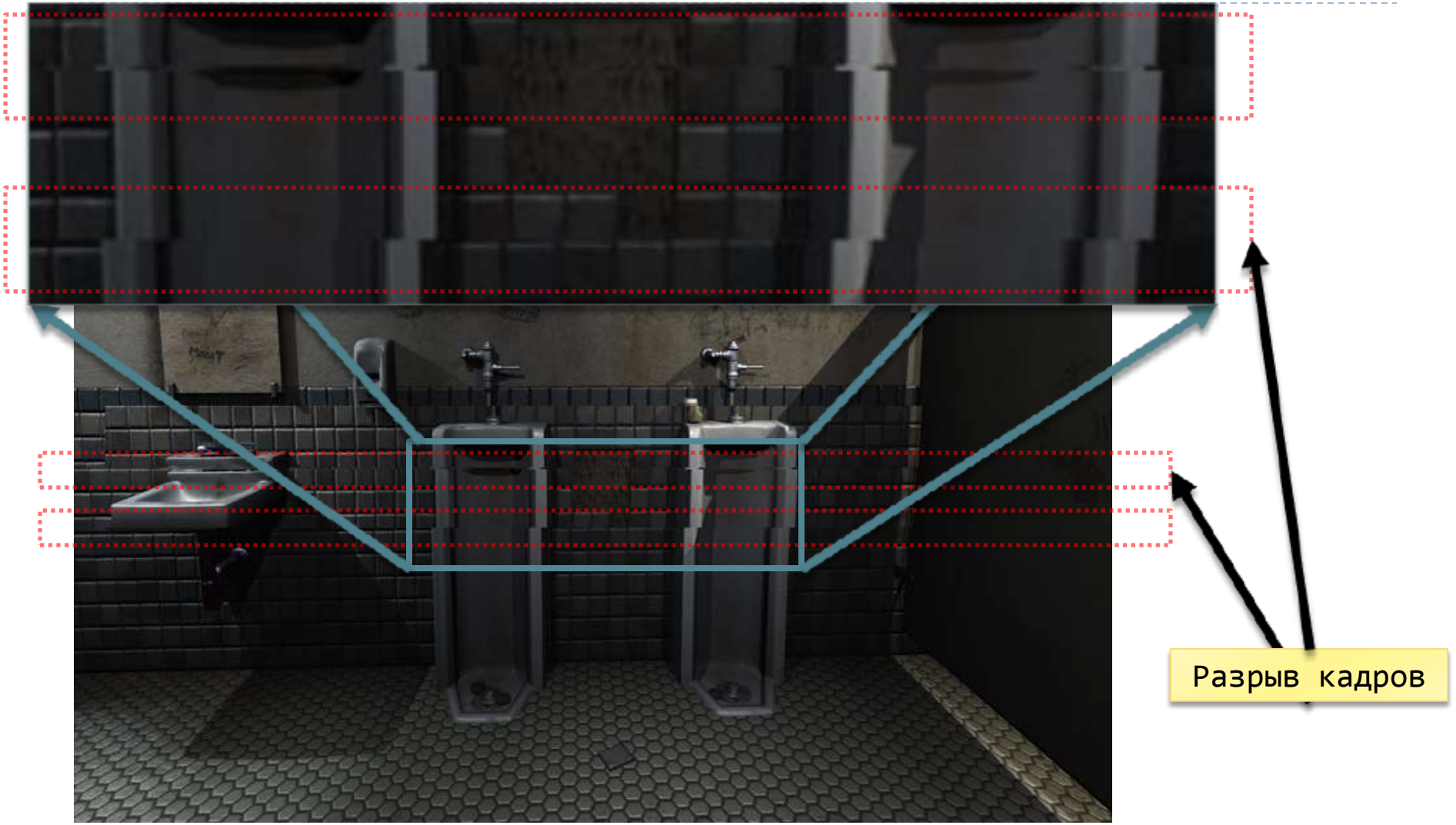
Двойная буферизация

- Отображение буфера кадра сразу на экран

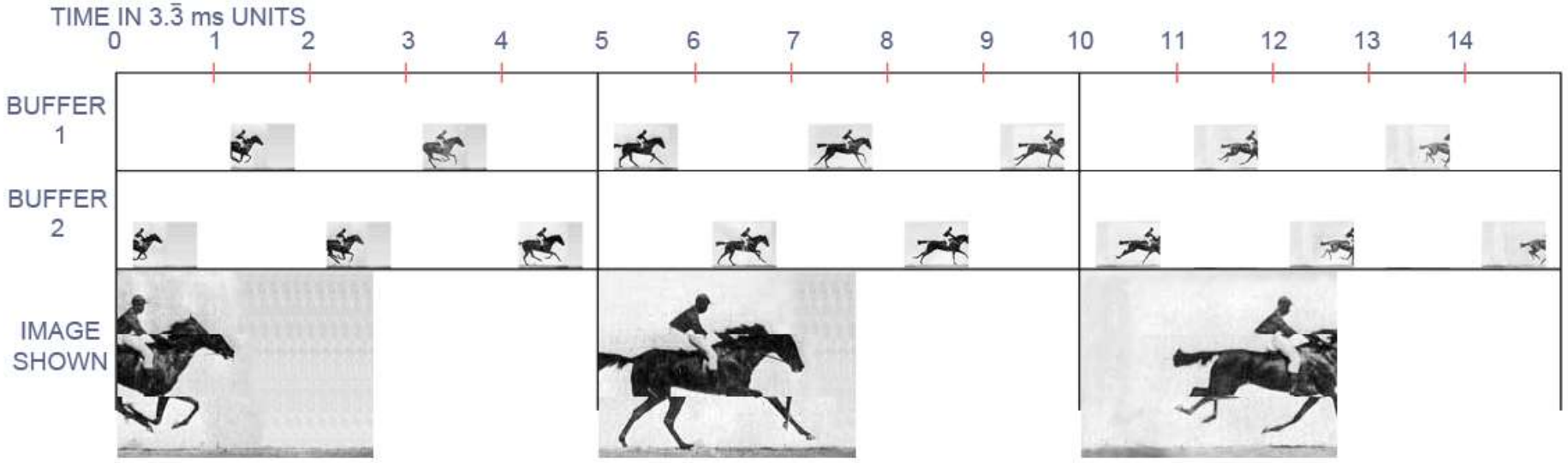
  Изображение может состоять из неполного кадра

- Двойная буферизация

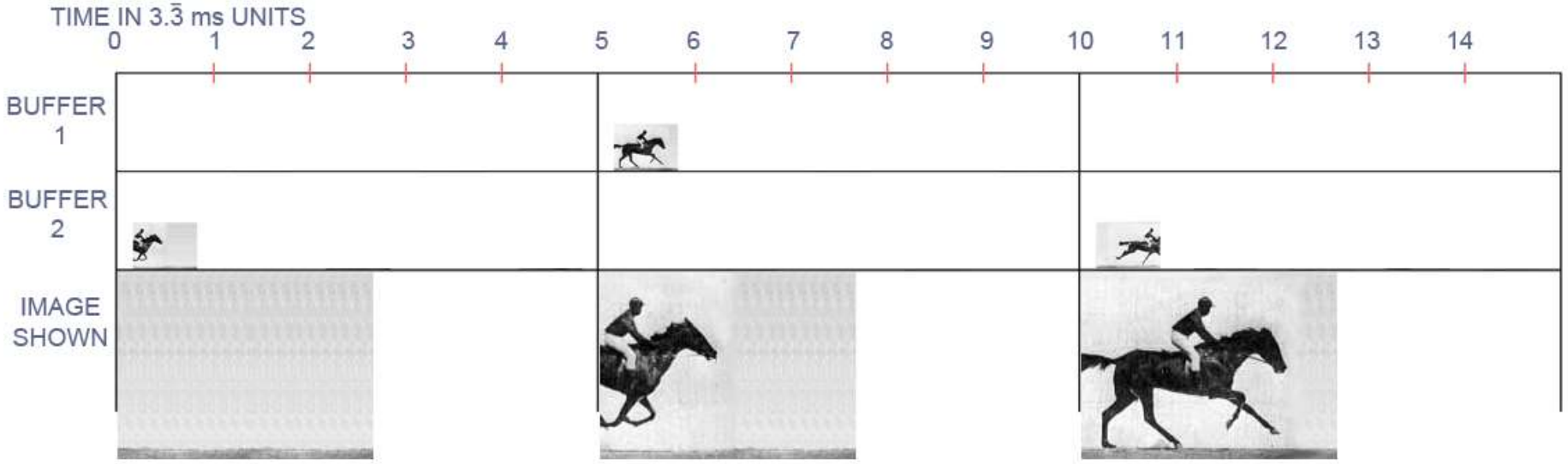
Screen tearing



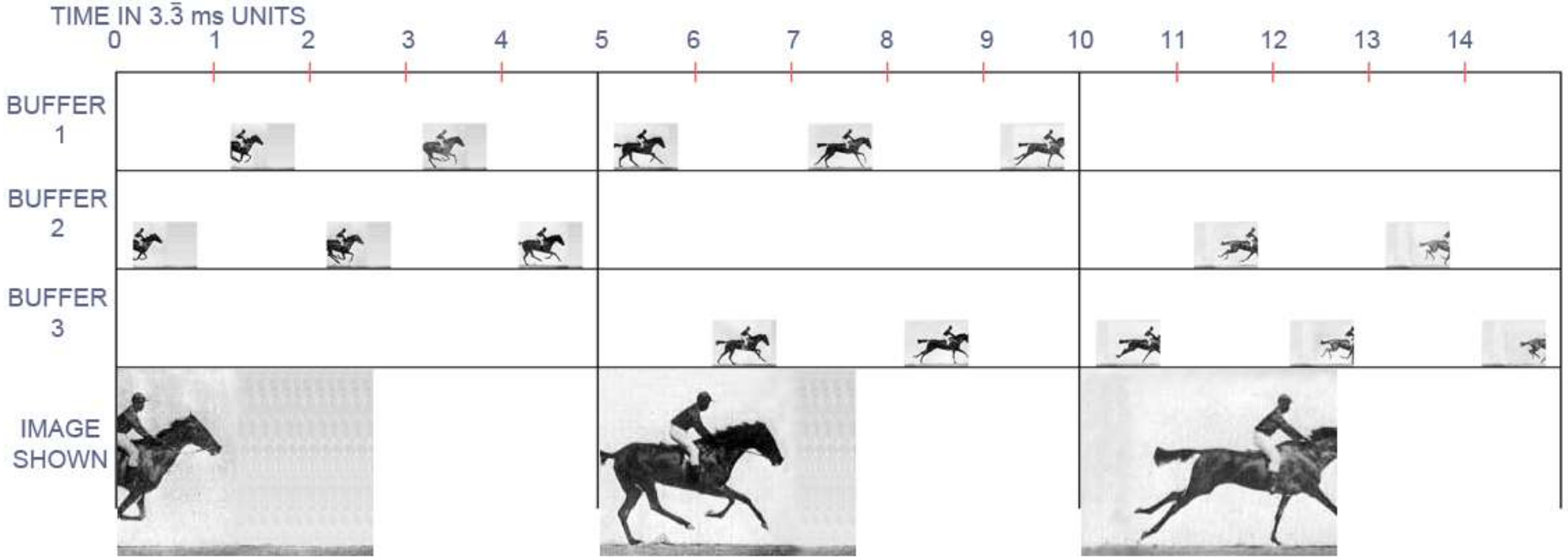
Двойная буферизация



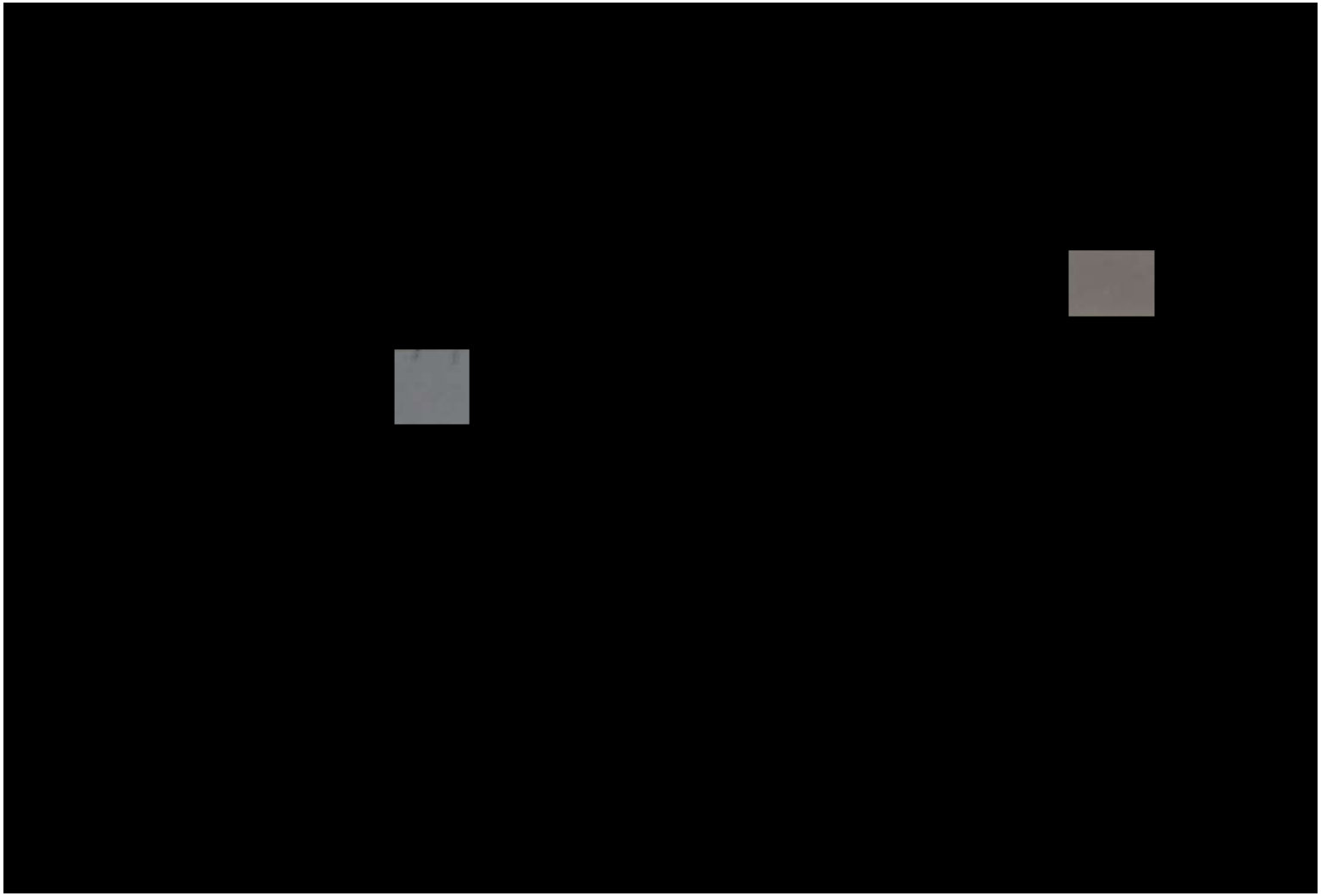
Двойная буферизация с вертикальной синхронизацией



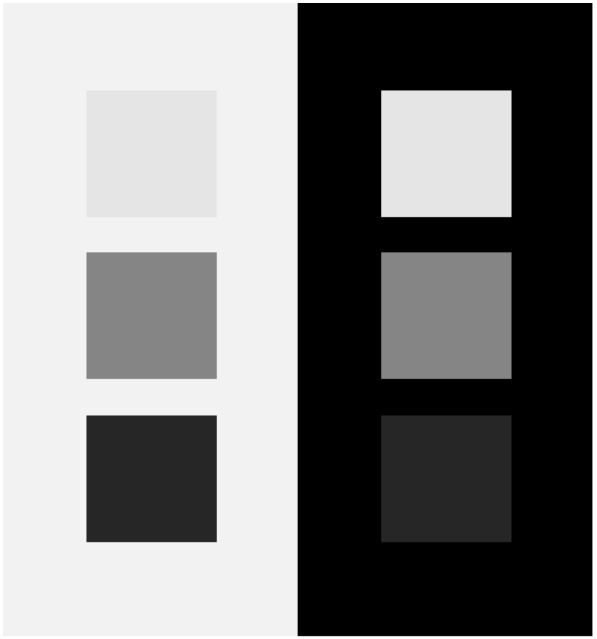
Тройная буферизация с вертикальной синхронизацией

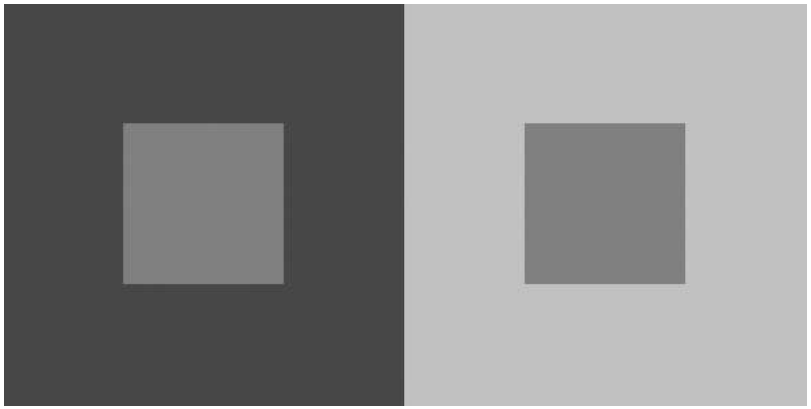


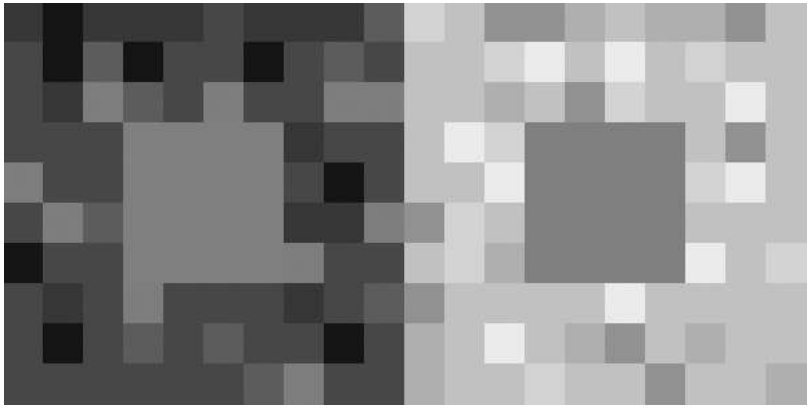
Восприятие света

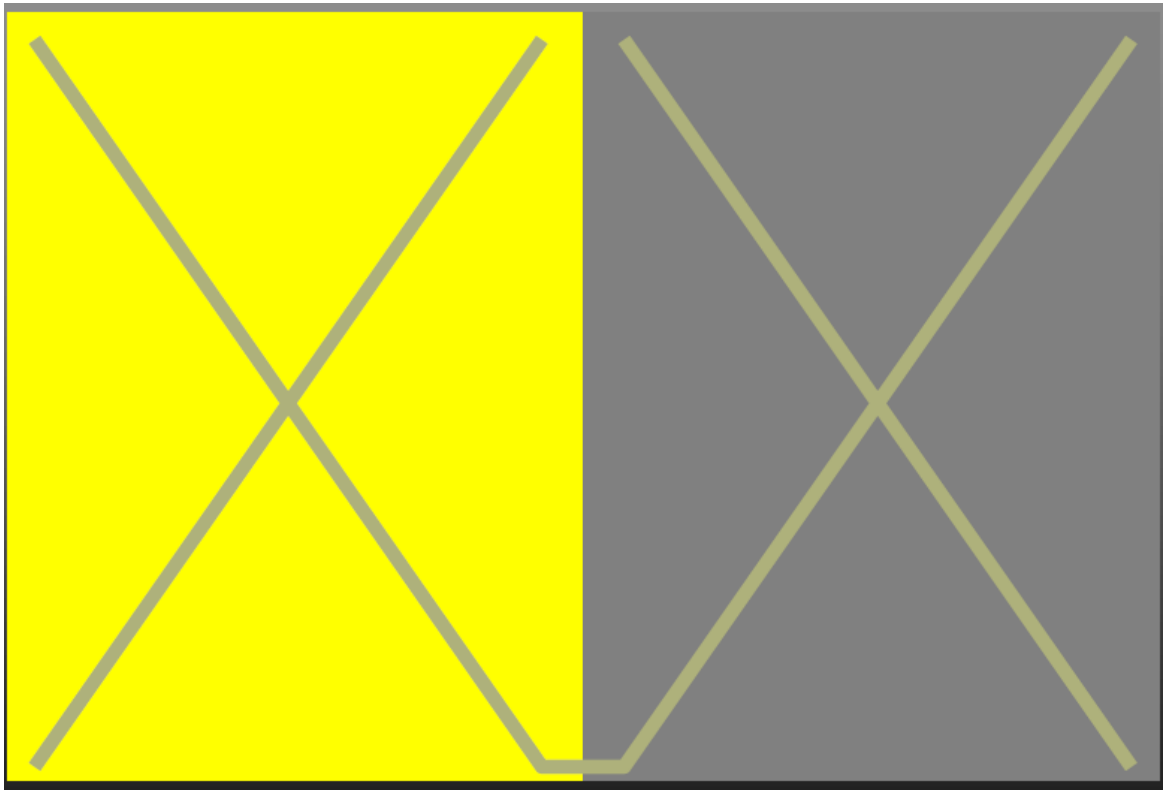


Surround effect



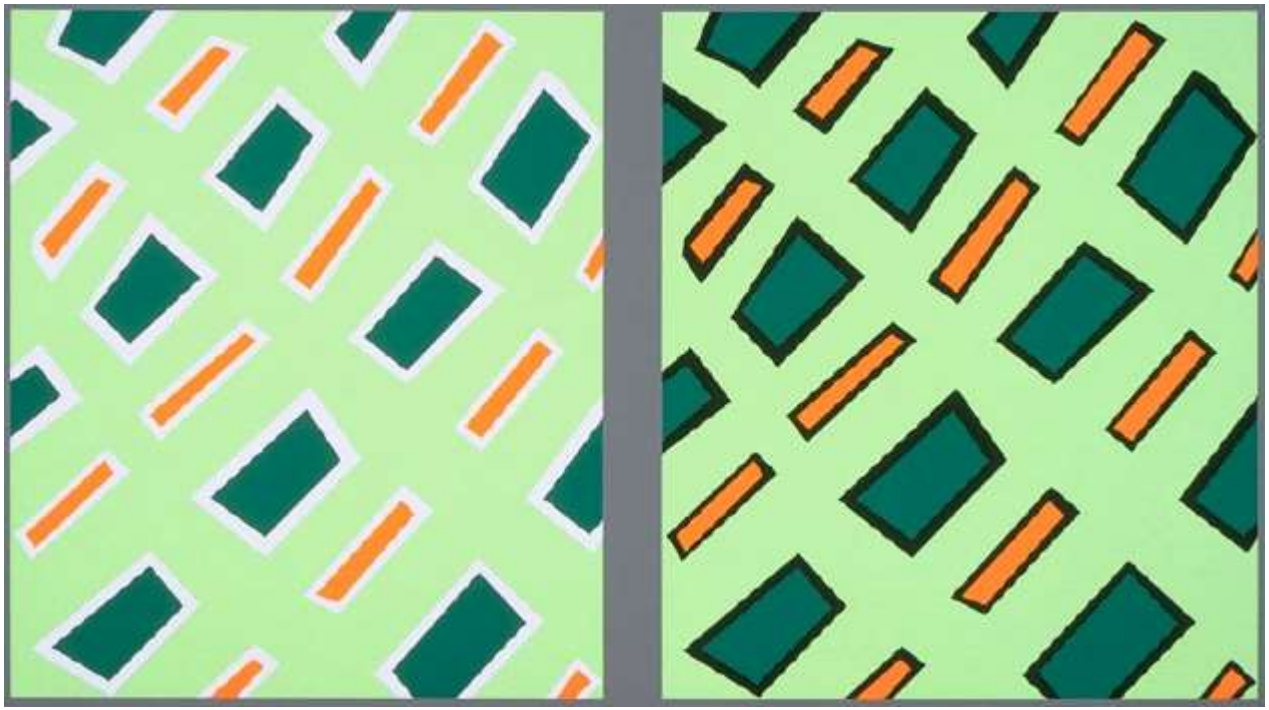






Эффект Безольда

Влияние окантовки



Вопросы