

# Компьютерная графика

## Постобработка

Лекция 8

Демяненко Я.М. ЮФУ 2025

# Постобработка (post processing)

**Постобработка (post processing)** — метод обработки в компьютерной графике, который добавляет дополнительные эффекты к буферу изображения перед его отрисовкой на экране с целью улучшения качества изображения.

# Эффекты постобработки в играх

Из эффектов постобработки в играх чаще всего используют:

- Bloom,
- Motion Blur,
- Depth Of Field.

Существует и множество других постэффектов:

- noise,
- flare,
- distortion,
- Sepia
- и др.

# Bloom

Наиболее яркие участки изображения делаются еще более яркими.

Эффект очень яркого света, проявляющийся в виде свечения вокруг ярких поверхностей, после применения bloom фильтра такие поверхности не просто получают дополнительную яркость, свет от них (ореол) частично воздействует и на более темные области, соседствующие с яркими поверхностями в кадре.



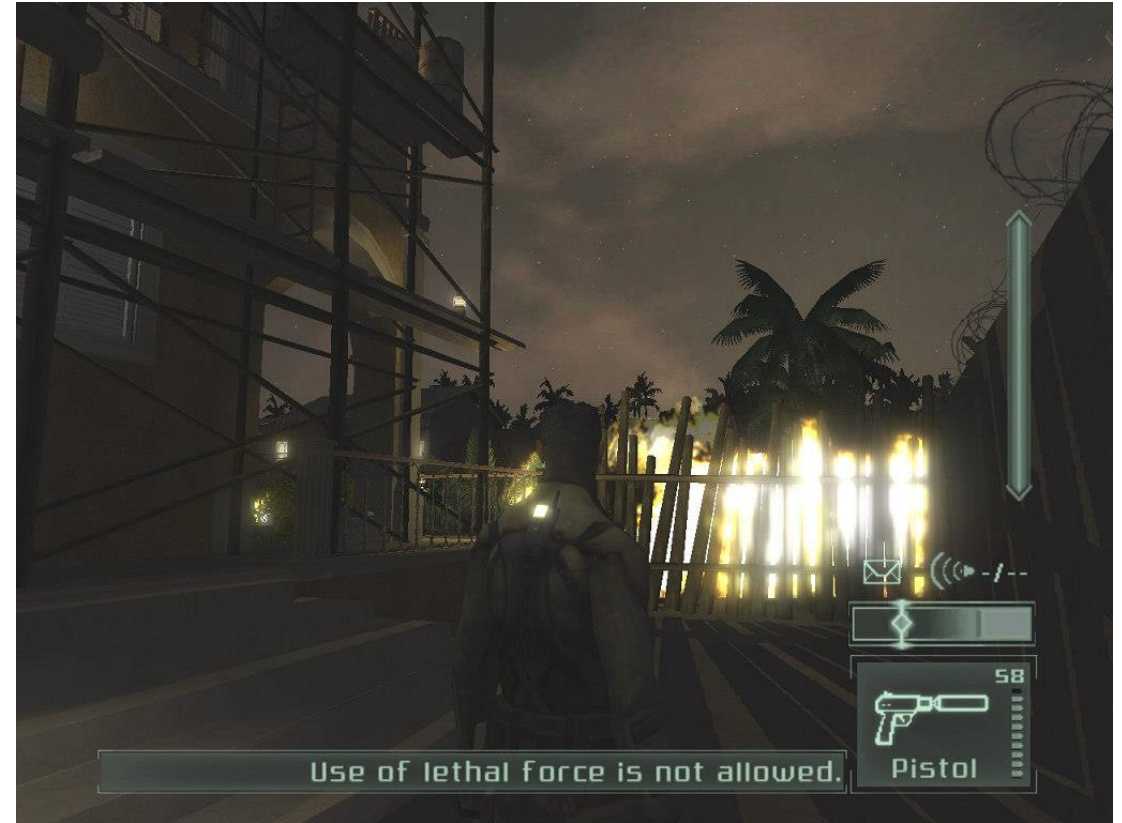
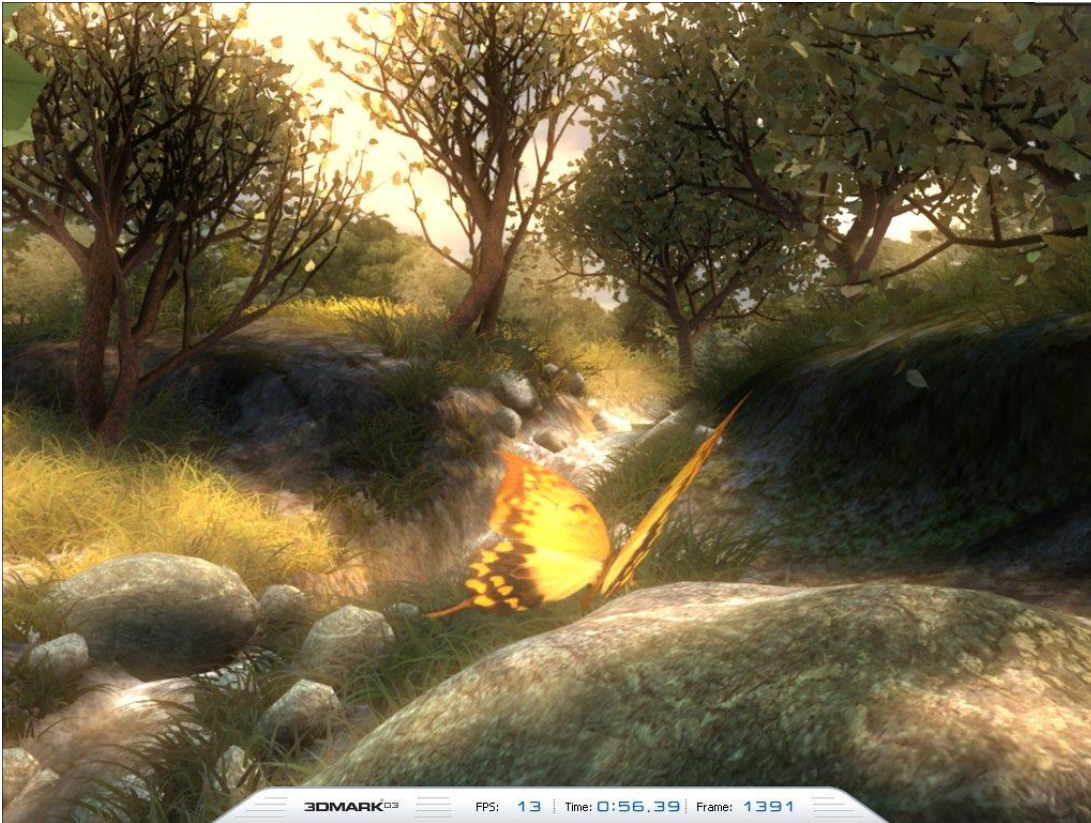
# Одна из реализаций Bloom фильтра

Смешивание смазанного фильтром blur кадра (всего кадра или отдельных ярких его областей, фильтр обычно применяется несколько раз) и исходного кадра.

- Сцена рендерится во фреймбуфер, интенсивность свечения (glow) объектов записывается в альфа-канал буфера.
- Фреймбуфер копируется в специальную текстуру для обработки.
- Разрешение текстуры уменьшается, например, в 4 раза.
- К изображению несколько раз применяются фильтры сглаживания (blur), на основе данных об интенсивности, записанных в альфа-канал.
- Полученное изображение смешивается с оригинальным кадром во фреймбуфере, и результат выводится на экран.



Как и другие виды постобработки, bloom лучше применять при рендеринге в широком динамическом диапазоне (HDR)



Примеры обработки конечного изображения bloom фильтром из 3D приложений реального времени

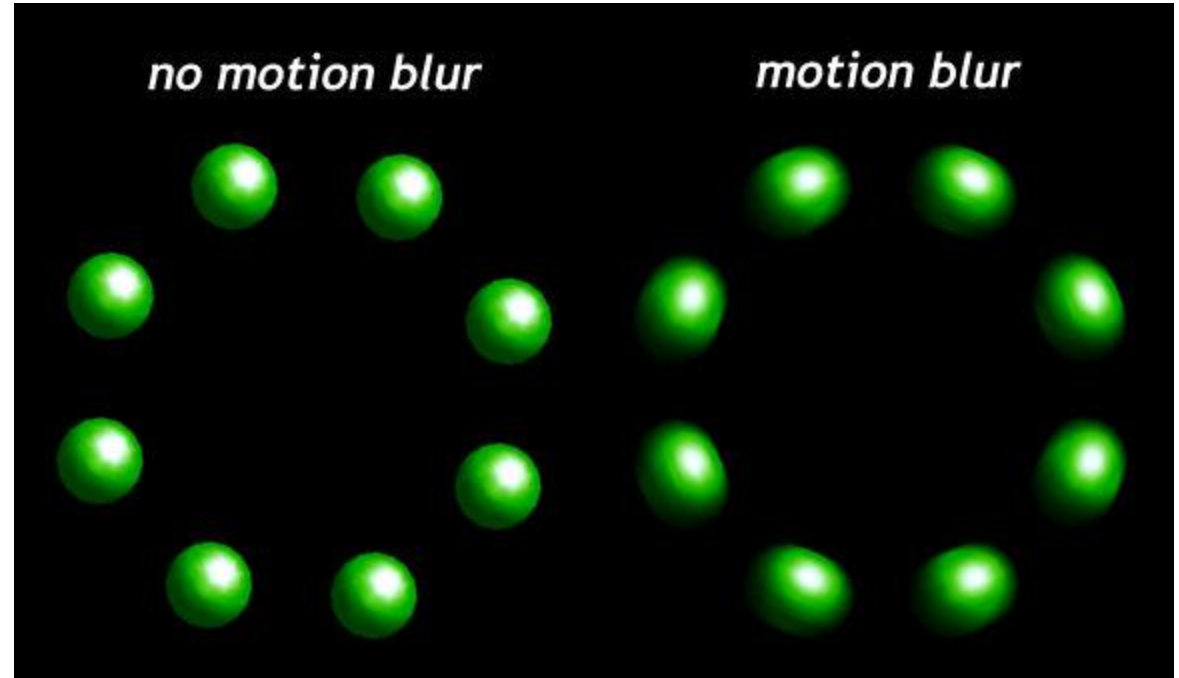
# Motion Blur

Отсутствие смазывания при движении служит и причиной того, почему движение в играх при 25-30 кадрах в секунду кажется дёрганным, хотя кино и видео при этих же параметрах частоты кадров смотрится прекрасно

Для компенсации отсутствия смазывания в движении желательна или высокая частота кадров ( $\geq 60$  кадров в секунду) или использование методов дополнительной обработки изображения (motion blur)

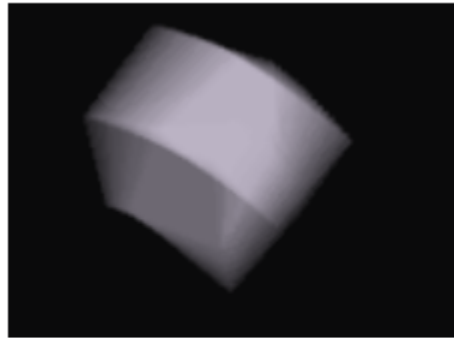
Это применяется и для улучшения плавности анимации и для эффекта фото- и кинореалистичности одновременно

Смазывание в движении (motion blur)



# Самый простой алгоритм motion blur

Самый простой алгоритм motion blur для приложений реального времени заключается в использовании для рендеринга текущего кадра данных из предыдущих кадров анимации



Корректная реализация  
motion blur



Неправильная реализация  
motion blur



# Реализации алгоритма motion blur

Но есть и более эффективные и современные методы motion blur, которые не используют предыдущие кадры, а основываются на векторах движения объектов в кадре, также добавляя лишь еще один шаг постобработки к процессу рендеринга

Это одна и та же картинка?



# Сглаживание переходных процессов во времени

Метод, применяемый для motion blur, называется temporal anti-aliasing, означающий сглаживание переходных процессов во времени

- а. Создаем избыточное количество кадров
- б. Делим последовательную анимацию на группы по 4 кадра
- в. Внутри каждой группы кадров произведём смешивание кадров в один
- г. Составим новую анимацию из этих кадров

# Реализации алгоритма motion blur

Эффект смазывания может быть как полноэкранным (обычно делается постобработкой), так и для отдельных, наиболее быстро движущихся объектов.

# Имитация motion blur

Разлетающиеся частички раскаленного вещества в месте взрыва можно рендерить в виде коротких линий, а не в виде точек

Это создаст впечатление motion blur

То же самое можно отнести и к процессу визуализации водопадов и выстрелов из оружия

# Имитация motion blur

Часто в автосимуляторах можно заметить, что вращение колес автомобиля выглядит очень неестественно.

Колеса автомобиля вращаются с такой бешеной скоростью, что какие-либо детали обода и покрышек просто не должны быть различимы.

Очень удобным решением в этом случае может быть предварительная подготовка нескольких текстур колес, просчитанных с различным количеством переходного сглаживания.

И по мере все более и более быстрого вращения колес использовать все более и более "размытые" текстуры.

Такой же подход возможен и к текстуре дороги.



# Примеры motion blur в играх



# Depth Of Field (DOF)

Depth of field (глубина резкости), это размывание объектов в зависимости от их положения относительно фокуса камеры.

В компьютерной графике каждый объект отрендеренного изображения идеально четкий, так как линзы и оптика не имитируется при расчетах.

Поэтому, для достижения фото- и кинореалистичности приходится применять специальные алгоритмы.

Эти техники симулируют эффект разного фокуса для объектов, находящихся на разном расстоянии.



# Одна из реализаций Depth Of Field

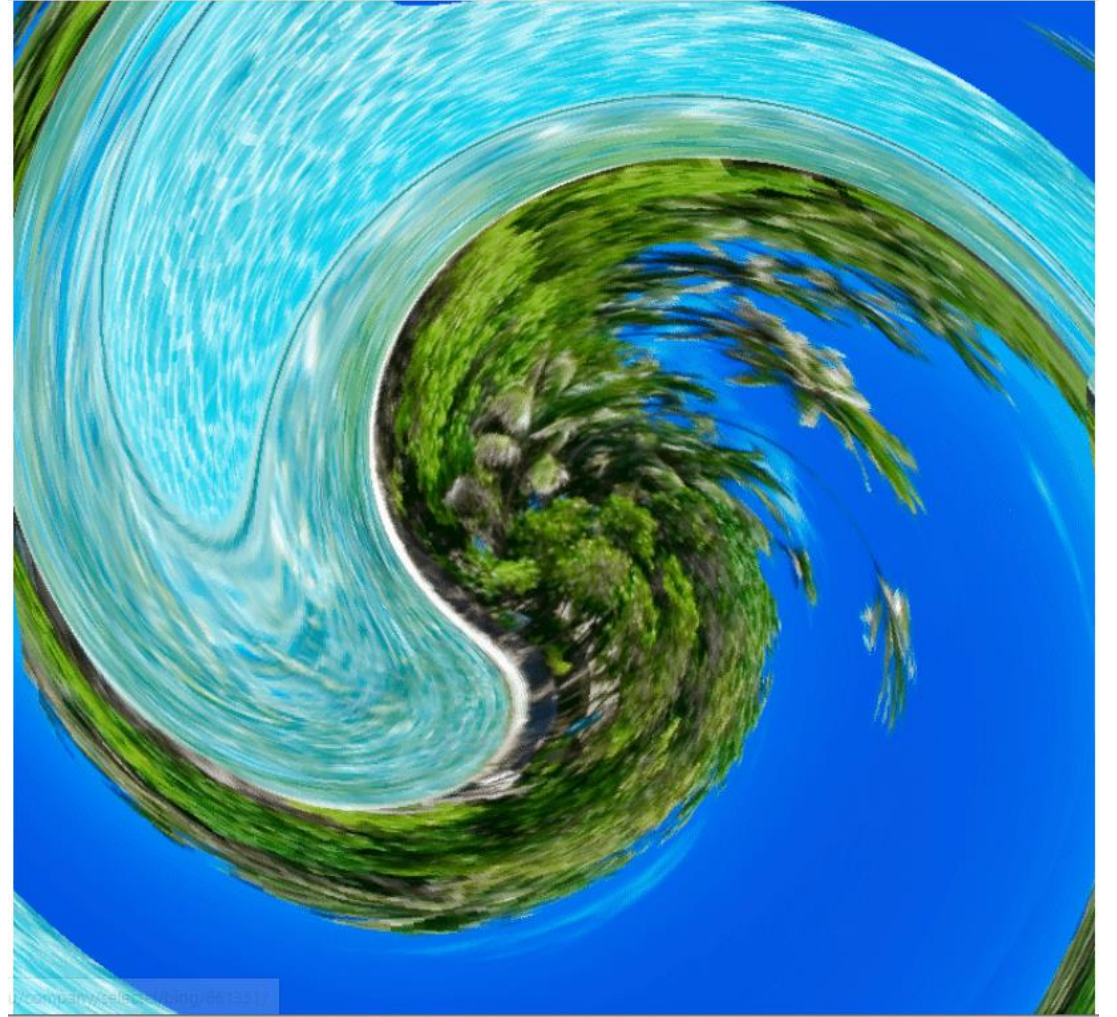
Одним из распространенных методов при рендеринге в реальном времени является смешивание оригинального кадра и его смазанной версии (несколько проходов blur фильтра) на основе данных о глубине для пикселей изображения.



# Примеры depth of field в реальном времени



Какие идеи?



# Fog (Туман)

Окружающий туман — эффект, который используется во всех играх, поскольку позволяет придать объектам или сцене реалистичность, а также почувствовать глубину сцены.

Кроме того, он позволяет скрыть ненужную геометрию.



# Интенсивность света в сцене с туманом

В сцене с туманом, интенсивность света, которая достигает камеры пользователя, зависит от трех компонент:

- Поглощение — величина, которая регулирует перекрытие световой интенсивности частиц вдоль трассируемого луча.
- Рассеивание — это выбивание лучей из луча, который идет к глазу.
- Рассеивание внутрь — это попадание (из-за рассеивания) других лучей в луч, который идет к глазу.

Эти три компоненты обобщают и характеризуют одним коэффициентом  $g$ , т.е. плотностью

# Эффект тумана зависит от трех основных компонент

- Цвет тумана ( $C$ )
- Плотность ( $g$ )
- Расстояние ( $d$ )

Поскольку объекты находящиеся вдали должны больше поддаваться эффекту тумана, ввели коэффициент  $d$ , т.е. расстояние.

Окончательный цвет вершины получают интерполированием цвета тумана и текущего цвета

# Встроенная реализация GL или через шейдеры

```
glEnable( GL_FOG );  
glFogi( GL_FOG_COLOR, color);  
Draw();  
glDisable( GL_FOG );
```

# Линейный туман

Самый простой из всех видов

Основное уравнение:

$$f = (Z_{\text{fogend}} - d) / (Z_{\text{fogend}} - Z_{\text{fogstart}})$$

В таком случае частицы тумана распределяются равномерно от точки  $Z_{\text{fogstart}}$  и до точки  $Z_{\text{fogend}}$ .

$d$  — представляет глубину, или расстояние от камеры до точки тумана

$f$  — влияние тумана на цвет вершины, т.е. при 0 — цвет вершины будет цветом тумана (случай когда геометрия находится вдали от зрителя), а при 1 — наоборот соответственно.

# Экспоненциальный (квадратичный) туман

Плотность тумана более быстро и плавно уменьшается, чем в случае с линейным туманом, поэтому эффект приближен к реальному эффекту

Основное уравнение для экспоненциального тумана:

$$f = \exp(-d * g)$$

для экспоненциального квадратичного тумана:

$$f = \exp(-(d * g) * (d * g))$$

# Техника depth fade

В число туманных эффектов входят разные явления, например, пыль и дым.

Обычно такие эффекты представляют собой полупрозрачную плоскость с текстурой и частицами.

Но у такого подхода есть одна проблема — в месте пересечения этой плоскости с геометрией на сцене возникает явная граница, из-за чего эффект выглядит устаревшим и фальшивым.

Тем не менее это самый простой способ его реализации.





# Как устроено depth fade?

Сначала идёт визуализация обычной геометрии — туда не входят прозрачные и полупрозрачные объекты.

Это необходимо для заполнения буфера глубины, чтобы знать, как далеко от камеры находится каждый пиксель.

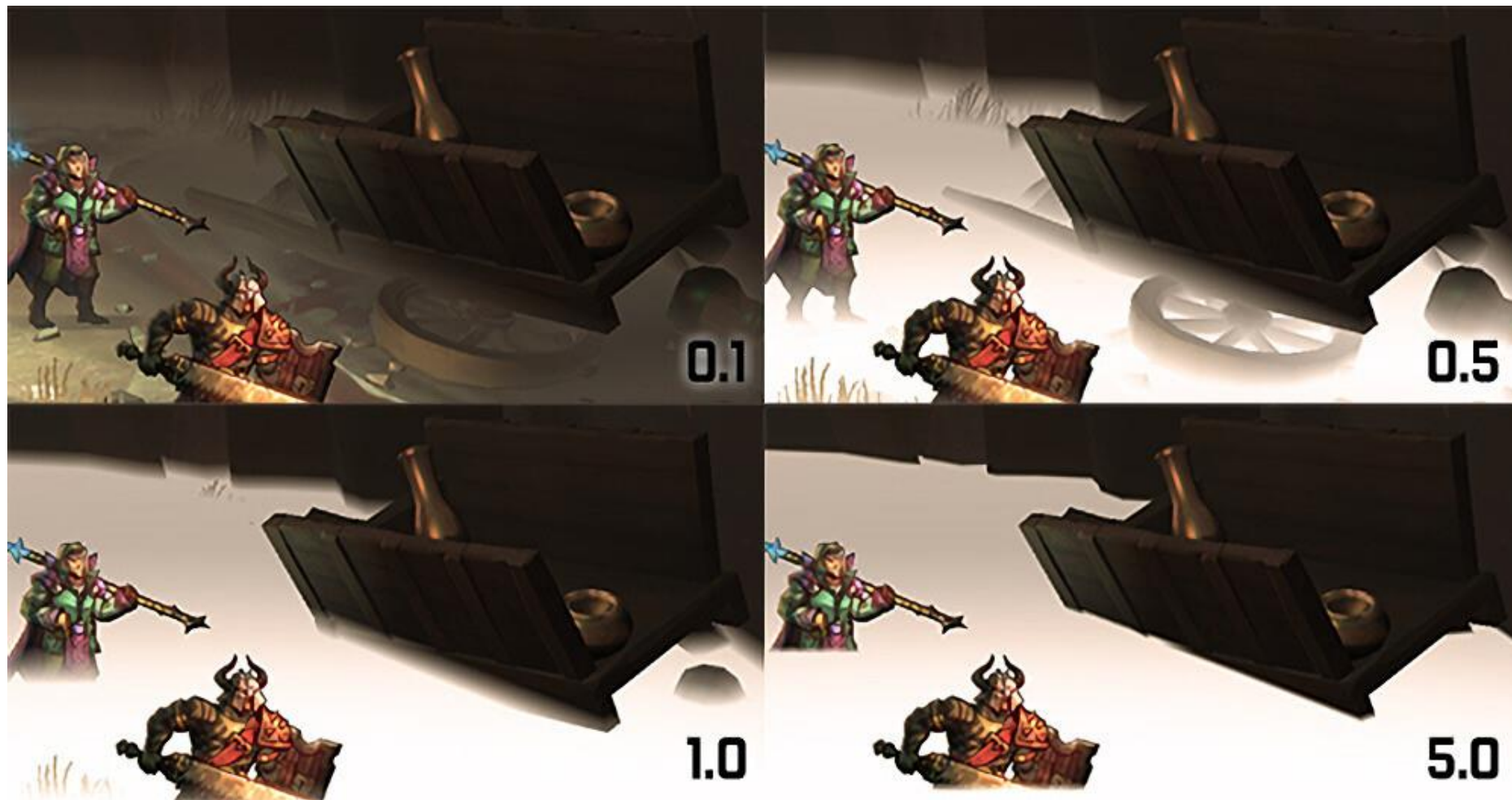
Затем при рендеринге объектов с depth fade фрагментный шейдер оценивает расстояние в буфере глубины и сравнивает его со своими показателями.

Это помогает определить, насколько близко пиксели находятся к области пересечения объекта и плоскости тумана: чем ближе, тем выше прозрачность.

Усиление прозрачности происходит до того момента, пока плоскость тумана не станет полностью невидимой в точке пересечения.

Если изменить расстояние действия эффекта, то можно легко менять плотность тумана.

При высокой плотности тумана эффект сглаживания границ пропадает. Это можно увидеть на нижнем правом примере



# Объёмный туман в Red Dead Redemption 2



# Объёмные лучи

Программная техника в 3D графике, предназначенная для создания эффекта трёхмерных лучей света, проходящих через окружение трёхмерной сцены.

В реальном мире эффект трёхмерных лучей представлен так называемым оптическим эффектом Тиндаля, когда лучи света проходят через облако газа, дыма, пыли, аэрозоля или пара в затемнённом пространстве.

При прохождении света через такую среду лучи света «подсвечивают» частички пыли или жидкости, которые распылены в воздухе, в результате чего можно увидеть эффект «объёмных лучей».

Ещё одним проявлением подобного оптического эффекта являются сумеречные и противосумеречные лучи.

Фотография: Пример сумеречных лучей, которые возникают во время заката





Фотография: Пример эффекта Тиндаля, когда солнечные лучи проходят через туман



Компьютерная графика: лесная местность в анимационном фильме «Big Buck Bunny», отчётливо видны объёмные лучи.





# Эффект прохождения лучей через водную аэрозоль (дождь) Реализация шейдерами



Автор: Акопджанова Николь



Компьютерная графика: объёмные лучи в компьютерной игре «Crysis», которая использует игровой движок «CryEngine 2».



# Принцип работы

Объемное освещение требует наличия двух компонентов: **теневой карты** (англ. shadow map) освещаемого пространства и **буфера глубины**.

Для каждого семпла проводится определение: освещается ли он светом из нужного источника света или нет; для этого используются данные из теневой карты. В итоге только освещенные семплы влияют на окончательный цвет пикселя.

Для функционирования в режиме реального времени вышеописанная методика требует оптимизации.

Одним из возможных оптимизационных подходов является рендеринг освещаемого объёма в намного меньшем разрешении чем то, которое используется в изначальном графическом контексте.

Вследствие такого подхода создаются нежелательные эффекты алиасинга, которые могут быть устранены посредством применения фильтра размытия.

# Анизотропная фильтрация

Анизотропная  
фильтрация  
заменила  
устаревшие  
технологии —  
Bilinear или  
Trilinear Filtering

Пример  
включённой и  
выключенной  
анизотропной  
фильтрации



# Какое выбрать сглаживание?



Без  
сглаживания



FXAA



TAA



SSAA x16



# FXAA (Fast-Approximate Anti-Aliasing)/ MLAA (Morphological Anti-Aliasing)

Варианты сглаживания при помощи постобработки от NVIDIA и AMD соответственно.

Видеокарта рендерит изображение без изменений, но перед отображением на экране к готовому кадру применяется фильтр, который размывает края объектов.

Этот процесс практически не сказывается на fps, но и результат получается достаточно сомнительный

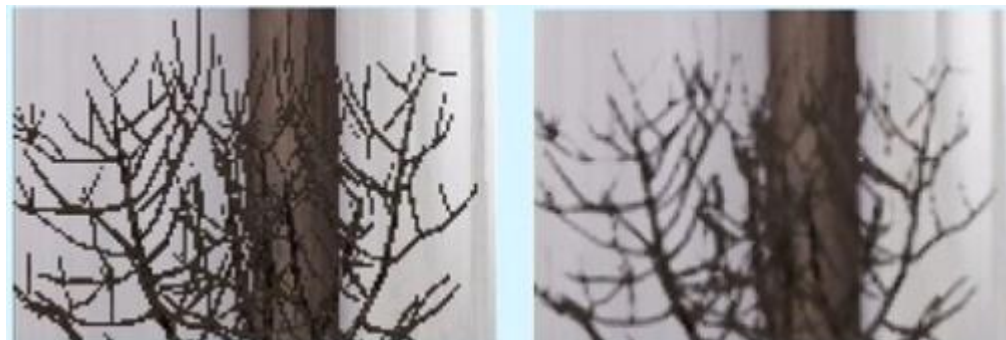


# ТАА (Temporal Anti Aliasing)

Метод постобработки при котором система анализирует один или несколько предыдущих кадров.

Этот метод лучше всего борется с мерцанием по краям движущихся объектов, но приводит к «замыливанию» и артефактам на границах.

Немного требовательнее, чем FXAA. Некоторые игры предлагают дополнительный фильтр, который добавляет картинке резкости после применения ТАА.



# SSAA (Super Sampling Anti Aliasing)

Один из первых способов сглаживания, но до сих пор в числе наиболее эффективных.

Технология заставляет видеокарту рендерить изображение в более высоком разрешении, а потом сжимает до разрешения экрана.

Основной недостаток — затраты ресурсов получаются огромные, и эффективным SSAA можно считать только на очень мощных ПК.



# MSAA (Multi Sampling Anti Aliasing)

С каждого пикселя собирается несколько проб цвета (до 8х) в разных местах.

Если одна или больше проб оказывается другого цвета, система выбирает «промежуточный» оттенок.

Анализ и сглаживание применяется точно только к границам объектов, поэтому влияние на производительность ниже, чем у SSAA.



## ТХАА (ТAA + MSAA)

Технология от NVIDIA, которая комбинирует суперсемплинг по двум или четырём точкам с временным сглаживанием.

Размытие на границах получается минимальным, изображение описывают как «мягкое» и «кинематографичное».

Требовательность ниже, чем у MSAA и SSAA, но выше, чем у других описанных методов.

# SMAA (Sub-Pixel Morphological Anti-Aliasing)

Метод сглаживания, который применяет анализ границ объектов (как MSAA), но при постобработке (как FXAA).

В результате — немного требовательнее, чем FXAA, но границы получаются менее размытыми.

# NVIDIA DLSS 2.0 (Deep Learning Super Sampling)

Единственный способ сглаживания, который улучшает, а не ухудшает производительность.

Видеокарта рендерит изображение в более низком разрешении, чем требуется, а потом увеличивает и сглаживает картинку при помощи аналога TAA и нейросети, которая работает на отдельных тензорных ядрах.

Поддерживается только видеокартами GeForce RTX 2060 и выше.

# Разные типы сглаживания на примере Total War: Warhammer 2



И ещё



**4K**  
**(без сглаживания)**



**MSAA 8x**



**4K**  
**DLSS 2.0 (из 1080p)**

## Что лучше?

Наиболее предпочтительным способом сглаживания без сомнения можно назвать DLSS 2.0, но его поддерживает не так много видеокарт и в небольшом количестве игр.

В остальных случаях стоит рассматривать MSAA, TXAA и SMAA