Реализация HDR на картах серии Radeon X1000

Обзор материала

- Что такое «HDR»?
- Факты о «HDR»
- Возможности Radeon X1x00
- Стратегии реализации «HDR»
- Использование целочисленных форматов
- Обсужение методов фильтрации
- Обзор методов

Что такое «HDR»?

- Большой динамический диапазон
- Динамический диапазон – отношение самого яркого и самого темного значения
  - Под «HDR» подразумевается динамический диапазон больше чем 255:1 использующийся при «нормальной» отрисовке
- Обычно для «HDR» нужен диапазон больше чем [0..1]
- Позволяет видеть детали как в тени, так в сильноосвещенных местах

Факты о «HDR»

- Динамический диапазон в реальном мире может быть огромен (например 1,000,000,000:1) и глаза могут адаптироваться к нему
- Динамический диапазон человеческого глаза около 30,000:1 после адаптации
- Динамический диапазон цифровой камеры около ~500:1

Факты о «HDR» в 3D графике

- Использование FP16 не обязательно, но это один из очень хороших вариантов
- Очень большой диапазон не обязателен чтобы выглядело хорошо
  - Все дело в хорошем художественном оформлении и изящной имплементации
- Существуют достаточно недорогие методы
  - Позволяют использовать «HDR» даже на «low-end»
- 3D графика – хорошо выглядящий «обман», и HDR не исключение ☺

Возможности Radeon X1x00

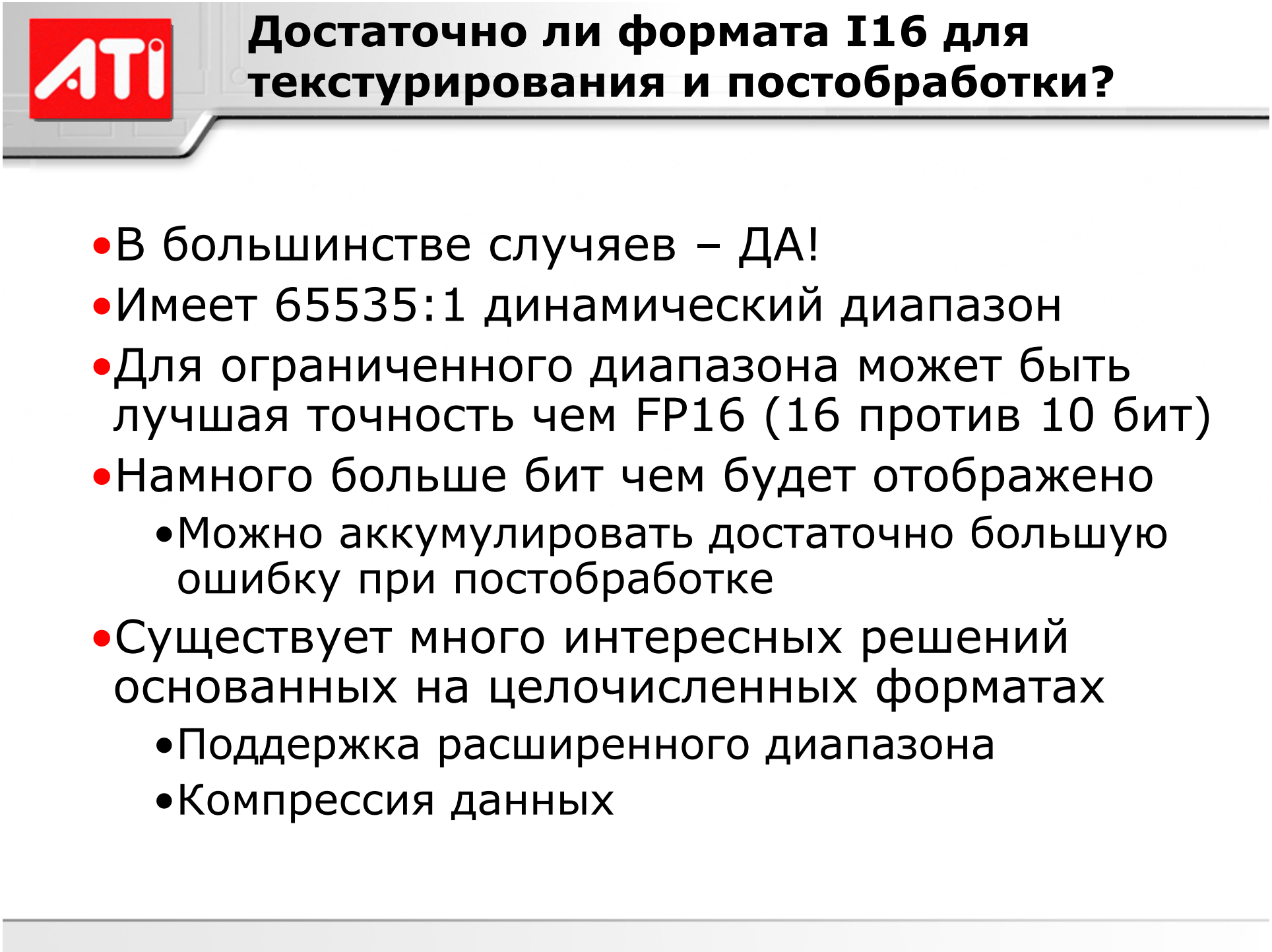
- I10 форматы
  - Фильтрация
  - Блендинг
  - MSAA
- I16 форматы
  - Фильтрация
- FP16 (4 канала)
  - Блендинг
  - MSAA
- I16 дополняет FP16 по функциональности

Реализация для «Low-End» (Radeon X1300- X1600)

- Наибольше подходит для «MDR»
- Используйте I10 для отрисовки (с MSAA)
  - Например в 2.8 формате с фикс. точкой
  - Гамма коррекция увеличивает диапазон
- При MSAA копируется в I10 буфер для постобработки
- Постобработка используя I10 или I16 форматы (оба фильтруемые)
- Используйте I10 или I8 для вывода изображения

Реализация для «High-End» (Radeon X1600- X1900)

- Используйте FP16 для отрисовки (с MSAA)
- Используйте альтернативные «HDR» форматы для хранения текстур
- Копируйте в буфер формата I16 для постобработки
- Постобработка используя I10 или I16 форматы (оба фильтруемые)
- Используйте I10 для вывода изображения



Достаточно ли формата I16 для текстурирования и постобработки?

- В большинстве случяев – ДА!
- Имеет 65535:1 динамический диапазон
- Для ограниченного диапазона может быть лучшая точность чем FP16 (16 против 10 бит)
- Намного больше бит чем будет отображено
  - Можно аккумулировать достаточно большую ошибку при постобработке
- Существует много интересных решений основанных на целочисленных форматах
  - Поддержка расширенного диапазона
  - Компрессия данных

Использование целочисленных форматов

- Много различных способов
  - Fixed scaling
  - RGBS, Compressed RGBE
  - RGBE, EEE
  - PPP
  - RGBS+PPP
- В примерах используется огромный диапазон [0.004, 76800.0]
  - Это 19,200,000:1 динамический диапазон!
  - Имитация очень большой выдержки (ночное небо выглядит как днем)

Формат «Fixed Scaling»

- Используется I16 с масштабированием
  - Пример: формат 8.8 с фикс. точкой
- Положительные черты
  - Диапазон 0..255 с хорошей точностью (8.8)
  - Пригодно для просто повышенной яркости
  - На практике можно ограничить диапазон
  - Соответствует билинейной фильтрации FP16
  - Метод прост и дешев в кодировании / декодировании
- Недостатки
  - Не работает для очень больших диапазонов

Реализация «Fixed Scaling»

- Кодирование

```
// Convert to [0..1] range
color.rgb *= invMaxValue;
```

- Декодирование

```
// Decode to full range
float3 tex = tex2D(Texture, texCoord).rgb;
tex.rgb *= maxValue;
```

Пример «Fixed Scaling»

Формат «RGBS»

- Переменный фактор масштабирования хранится в альфа канале
  - Плавающая точка с линейным распределением по диапазону
- Положительные черты
  - Дешевое декодирование (2 инстр.)
  - Больший диапазон чем у «fixed scale»
  - В целом не плохое качество для достаточно больших диапазонов
- Недостатки
  - Не работает с огромными диапазонами
  - Не дешевое кодирование (до 9 инстр.)

Реализация «RGBS»

- Кодирование

```
// Might need to clamp
color.rgb = min(color.rgb, maxValue);
// Find max value
float maxChannel = max(max(color.r, color.g), color.b);
// Move scale to alpha
color.rgb /= maxChannel;
color.a = maxChannel * invMaxValue;
```

- Декодирование

```
// Decode to full range
float4 tex = tex2D(Texture, texCoord);
tex.rgb *= tex.a * maxValue;
```

Улучшенный «RGBS»

- В темных частях изображения могут быть проблемы с точностью
  - Недостаточно бит значения масштаба
  - Это потому что кодировка цвета использует максимальное кол-во бит
- Решение: перераспределить используемые биты между цветом и значением масштаба
  - Используйте одинаковое кол-во бит
  - Подгоняется специальным множителем
    - Цвет делится на «adjustment value»
    - Масштаб умножается на «adjustment value»

$$\frac{\max(R,G,B)}{f} = f \cdot A \qquad \text{Adjustment: } f = \sqrt{\frac{1}{A}}$$

Реализация улучшенного «RGBS»

- Кодирование

```
// Might need to clamp
color.rgb = min(color.rgb, maxValue);
// Find max value
float maxChannel = max(max(color.r, color.g), color.b);
// Move scale to alpha
color.rgb /= maxChannel;
color.a = maxChannel * invMaxValue;
// Redistribute bits
color.a *= rsqrt(color.a);
color.rgb *= color.a;
```

- Декодирование

```
// Decode to full range
float4 tex = tex2D(Texture, texCoord);
tex.rgb *= tex.a * maxValue;
```

Пример формата «RGBS»

Формат «RGBE»

- Похож на RGBS, но в альфе хранится экспонента (имплементация FP)
  - HW фильтрует до конвертирования
- Положительные черты
  - RGBE8 хорош для хранения HDR текстур
  - RGBE16 может кодировать огромные диапазоны с прекрасной точностью
  - Дешевое декодирование (3 инстр.)
- Недостатки
  - Фильтрация несколько хуже чем у «RGBS»

Обработка экспоненты

- У RGBE8 и RGBE16 очень много бит для экспоненты
  - FP16 – только 5 бит
  - FP32 –8 бит
  - RGBE8 – 8 бит
  - RGBE16 – 16 бит!
- Не используйте экспоненту напрямую – слишком расточительно
  - «Bias» и «scale» для подгонки диапазона
  - Для текстур анализ может быть «off-line»

Реализация «RGBE»

- Кодирование

```
// scale = maxExp - minExp;
// bias  = minExp;
// invScaleBias = float2(1.0 / scale, -bias / scale)
float maxChannel = max(max(color.r, color.g), color.b);
color.rgb /= maxChannel;
// Encode exponent
color.a = log2(maxChannel) * invScaleBias.x + invScaleBias.y;
```

- Декодирование

```
// Decode to full range
float4 tex = tex2D(Texture, texCoord);
tex.rgb *= exp2(tex.a * scaleBias.x + scaleBias.y)
```

Пример «RGBE»

Компрессия форматов «RGBE»/«RGBS»

- Размер текселей RGBE16 слишком большой
  - Не приемлем для большинства игр
- Так много бит не нужно для кодирования цвета
- Можно использовать различную точность для разных кодируемых компонент
  - 16 бит текстура для экспоненты или масштаба
  - RGBA8+L16 – 48 бит/тексел
  - RGBA8+L8 – 40 бит/тексел
  - DXT1+L16 – 20 бит/тексел

Пример «RGBE» с компрессией

Формат «PPP»

- Значения возведенные в степень [0..1]
  - Лучше всего подходит значение степени 0.5
- Положительные черты
  - Большой диапазон и неплохое качество фильтрации, вцелом хорошее решение
  - Не использует альфа канал
  - Работает даже с I10 в случае с «MDR»
  - Достаточно дешевое решение для степени 0.5
- Недостатки
  - Дорогое декодирование для степени ≠ 0.5
  - Возможно придется ограничить диапазон для 0.5 значения степени

Реализация «PPP»

- Кодирование (степень 0.5)

```
// Use square root for power 0.5
color.rgb *= invMaxValue;
color.rgb *= rsqrt(color.rgb);
```

- Декодирование (степени 0.5)

```
// Decode to full range
float3 tex = tex2D(Texture, texCoord).rgb;
tex.rgb *= tex.rgb * maxValue;
```

Пример «PPP»

Формат «RGBS+PPP»

- Возможно использование «PPP» кодирования для улучшения качества и диапазона «RGBS» метода
- Положительные черты
  - Большой диапазон
  - В целом отличное качество
  - Подходит даже для формата I10
- Недостатки
  - Дорогая кодировка (до 10-14 инстр.)

Пример «RGBS+PPP»

Формат «EEE»

- Хранит поканальную экспоненту
  - Использование «scale» и «bias» позволяет использовать весь доступный диапазон
- Положительные черты
  - Лучше чем «RGBE» в худших случаях
  - Подходит даже для I10 в «MDR» случаях
  - Дешевое кодирование (4 инстр.)
  - Альфа не используется
- Недостатки
  - Более дорогое декодирование (4 инстр.)
  - Нет возможности представить нуль

Реализация «EEE»

- Encoding

```
// Use exponent scale and bias
color.rgb = log2(color.rgb) * invScaleBias.x + invScaleBias.y;
```

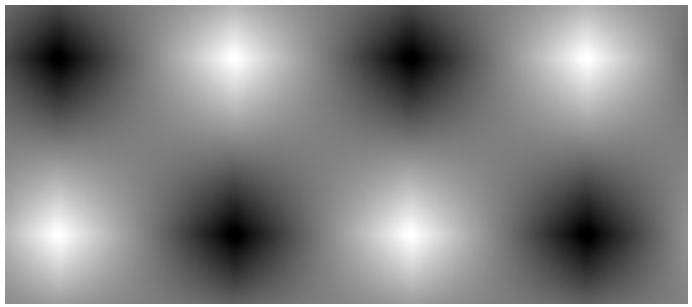
- Decoding

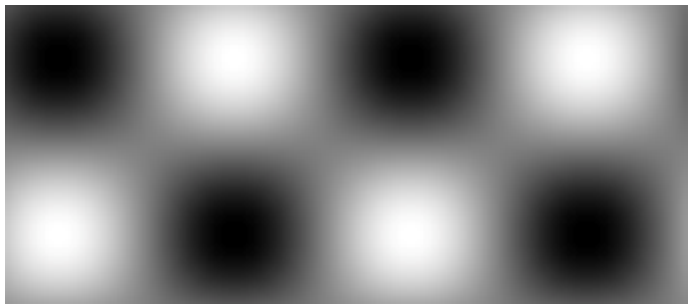
```
// Decode to full range
float3 tex = tex2D(Texture, texCoord).rgb;
// Deal with 0 values
tex = (tex > float3(0,0,0))?
        exp2(tex * scaleBias.x + biasBias.y) :
        float3(0,0,0);
```

Пример «EEE»

Как насчет фильтрации?

- Да, не работает с билинейной фильтрацией, но это не проблема
- Билинейная фильтрация далеко не самая лучшая для востановления сигнала
  - Используется потому что недорога и выглядит «достаточно хорошо»
- Существуют фильтры и получше...
  - Сравните билинейную и бикубическую фильтрацию



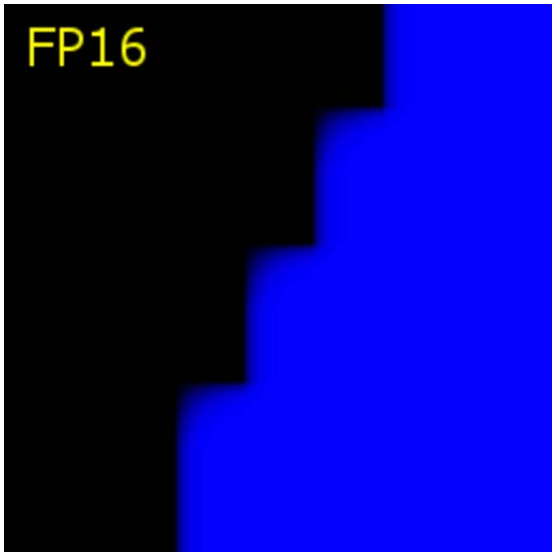


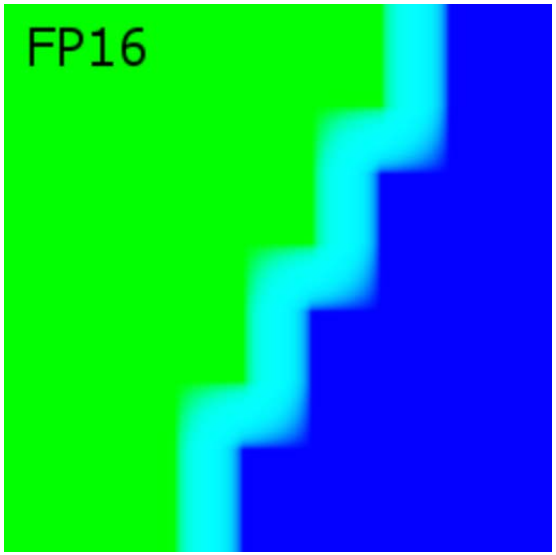
Как насчет фильтрации?

- Эти методы не соответствуют билинейке
- … но они достигают такой же цели – сглаживают изображение
- Положительные черты
  - Хорошо сглаживают изображение
  - В случае с «HDR» нелинейность фильтрации не обязательно плоха
- Недостатки
  - Возможно образование ореолов при резком изменении значений (редко проблематично)
  - Возможно придется поиграться с диапазоном и методами для наилучшего изображения

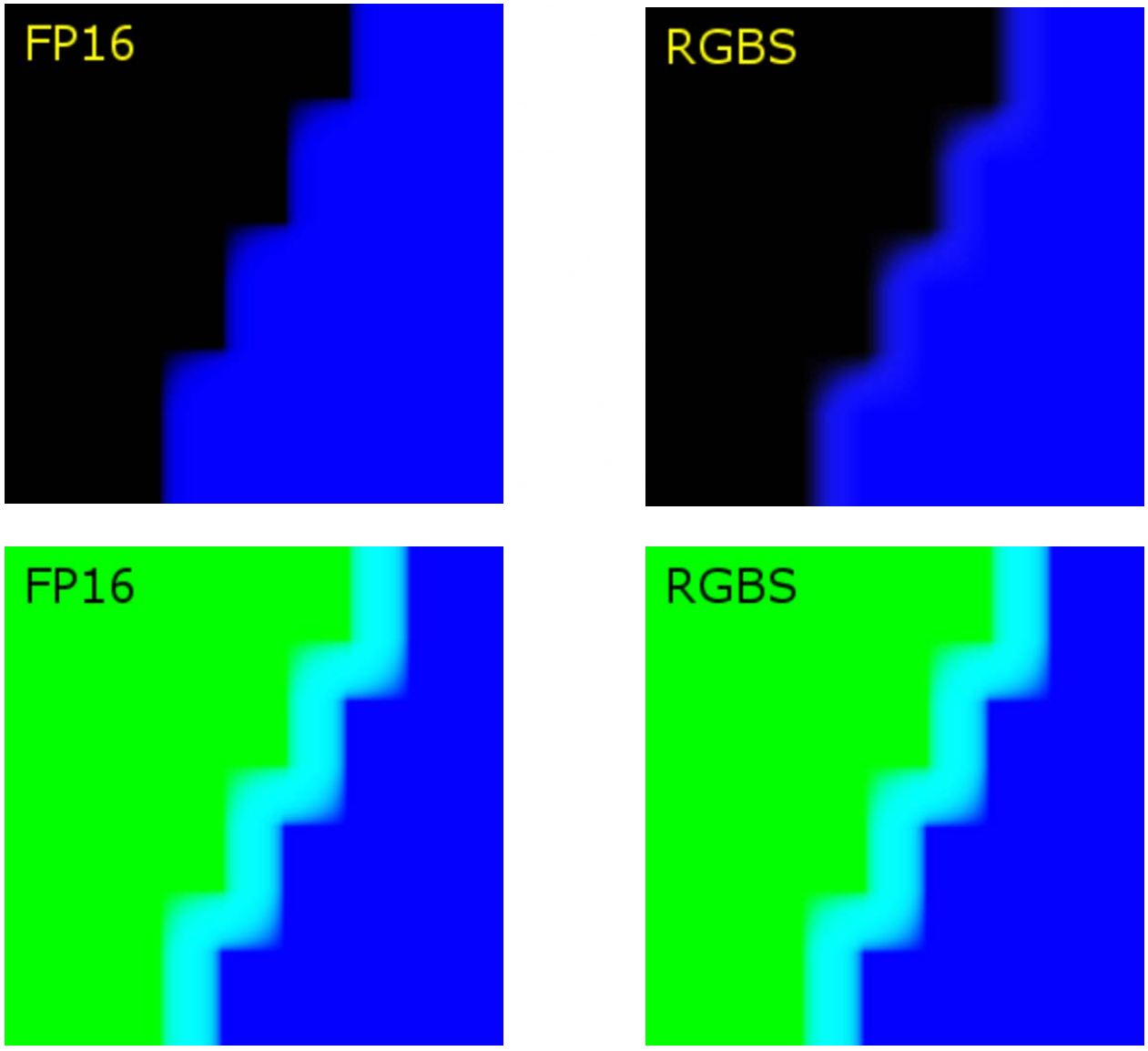
Сравнение фильтрации

- Это экстремально плохие случаи
  - Увеличенные изображения
  - Контраст между соседними пикселами 1000
  - Одно- и многоканальные случаи
- Эталонные изображения

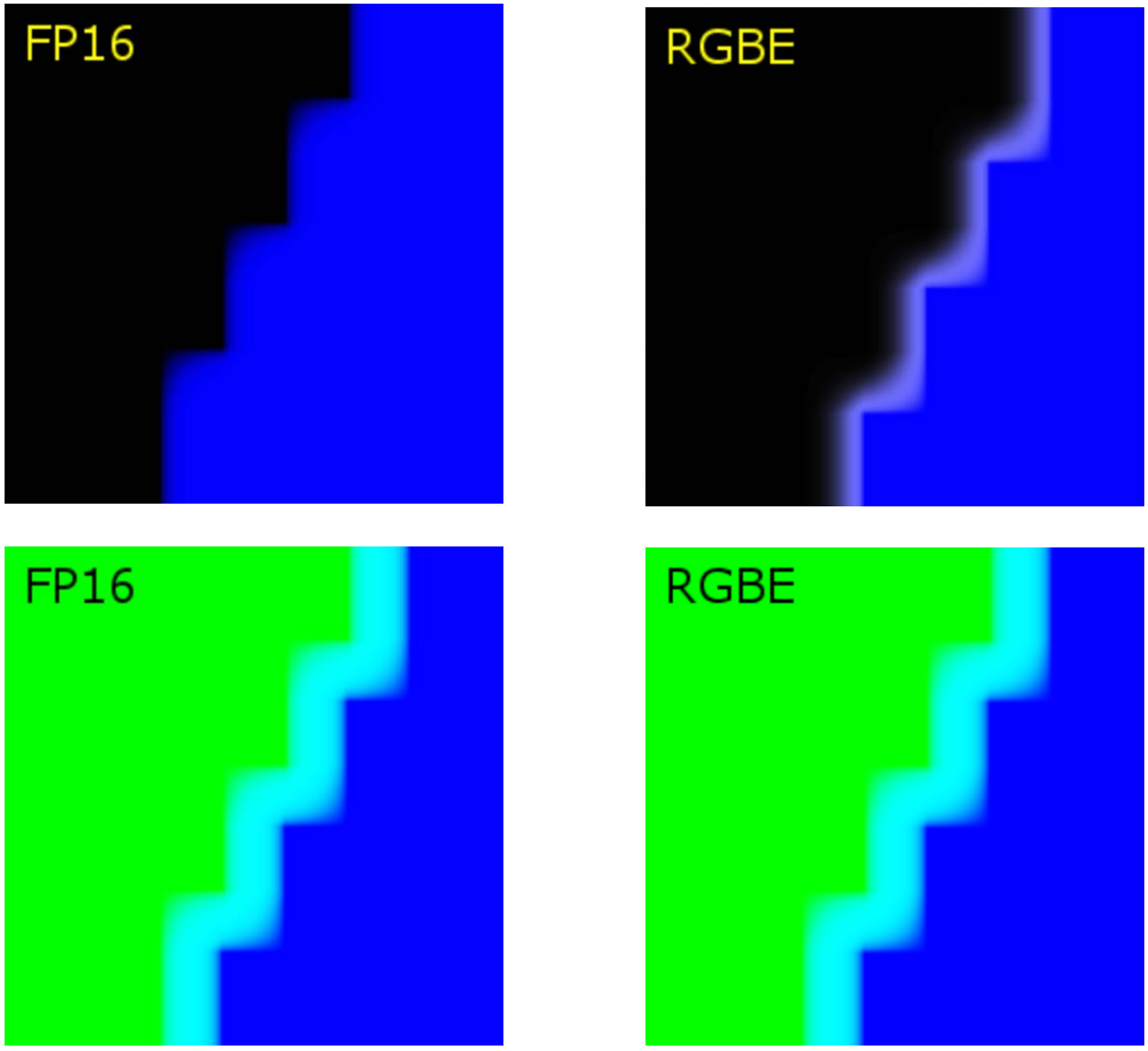




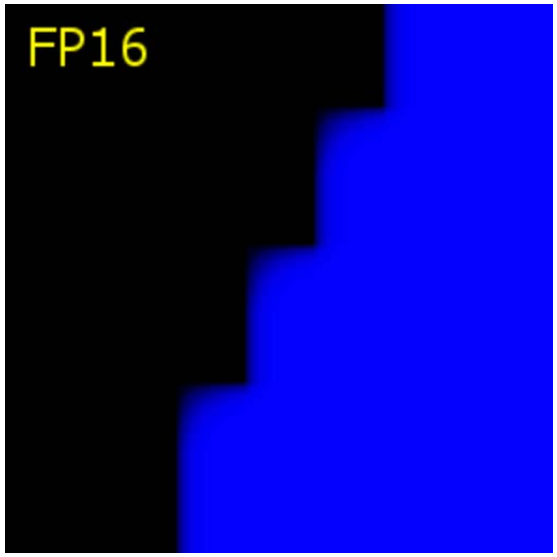
Сравнение фильтрации (RGBS)

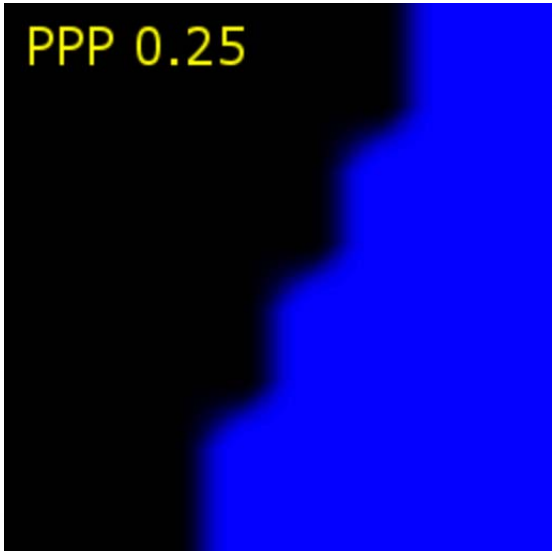


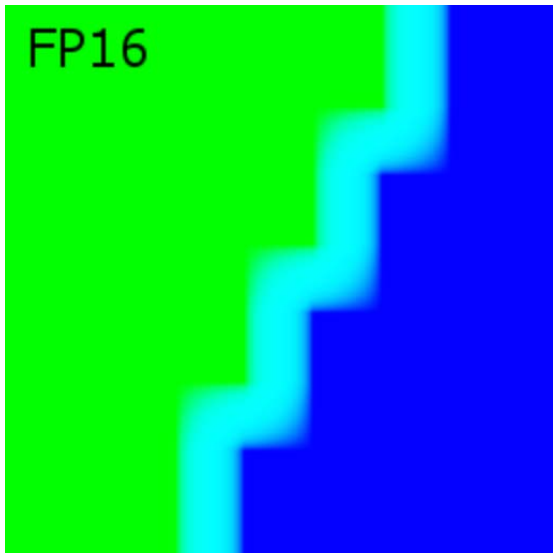
Сравнение фильтрации (RGBE)

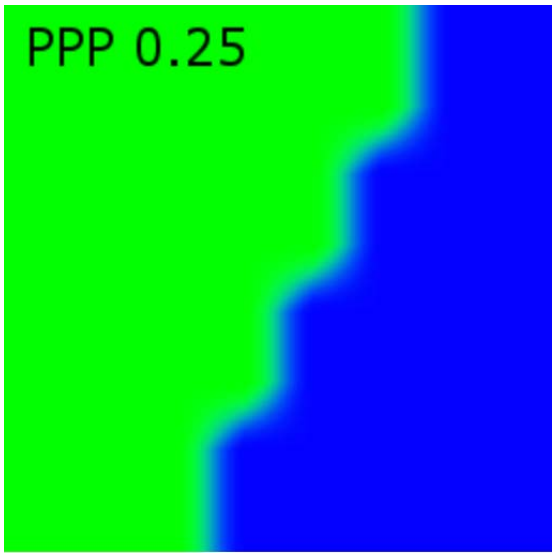


Сравнение фильтрации (PPP)

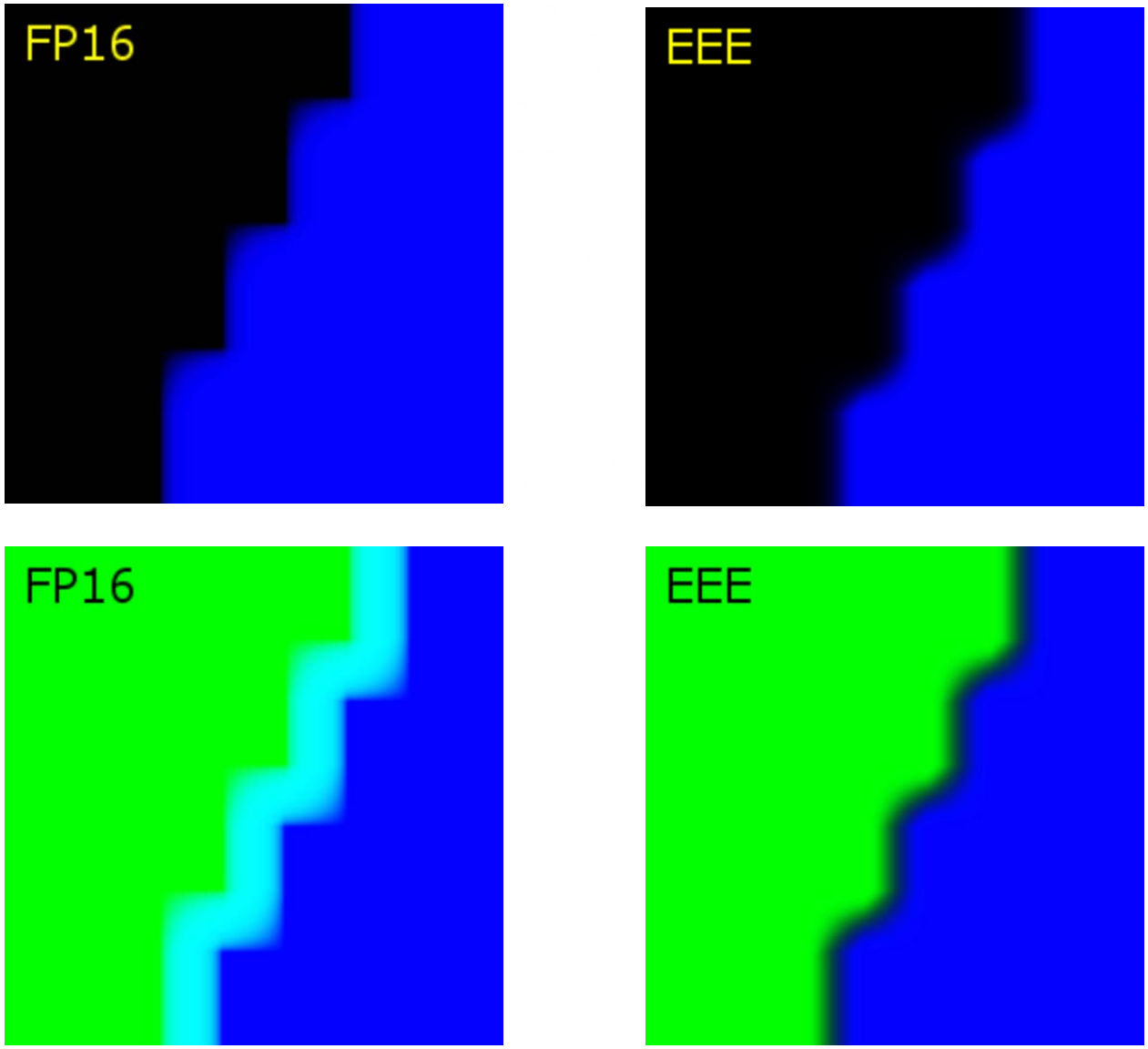








Сравнение фильтрации (EEE)



Обзор методов

- Выберите наилучший для данной ситуации способ
  - Компромис между качеством и производительностью

Method	Encod. instr.	Decod. Instr.	Free Alpha	Range	Filter quality	Overall quality
Fixed scale	1	1	Yes	Low	+++	+
RGBS	5-9	2	No	Med	++	+
RGBE	5-6	3	No	High	+	+++
PPP	5/8	2/8	Yes	High	+++	++
RGBS+PPP	10-14	3	No	High	++	+++
EEE	4	4	Yes	High	+	++

Вопросы