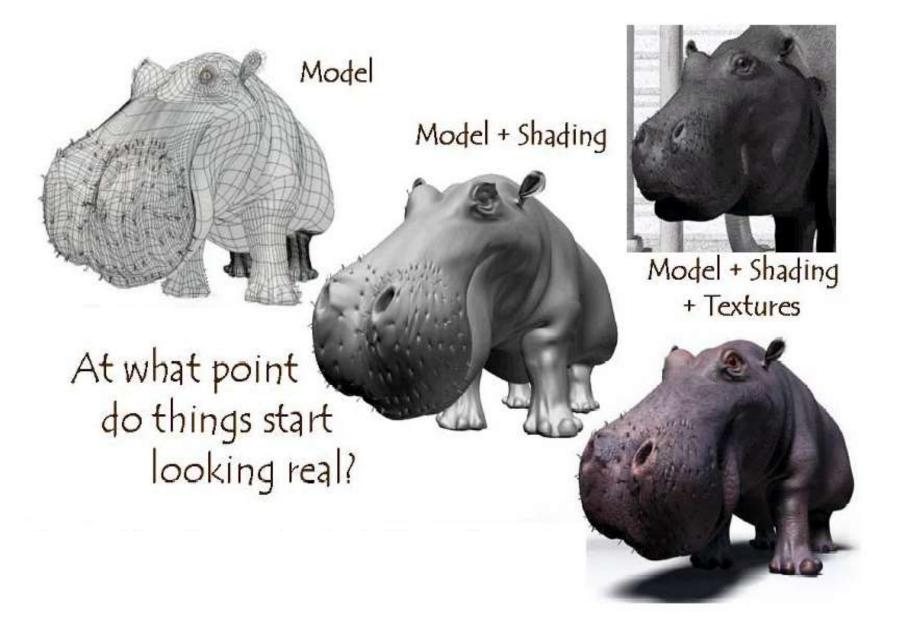
# Компьютерная графика и визуализация в реальном времени

#### Текстурирование

Алексей Романов

# The Quest for Visual Realism

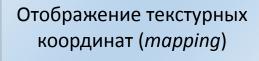


#### Цель – реалистичная картинка

- ▶ 3D модель должна как можно ближе быть к оригиналу
- Поверхность модели сложная структура с большой детализацией
- Необходимо задавать/моделировать эти детали
  - ▶ Процедурный подход, е.g. фракталы на шейдере
  - Моделирование всех деталей

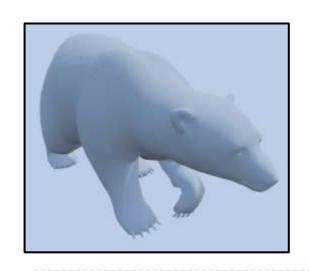
# Текстурирование



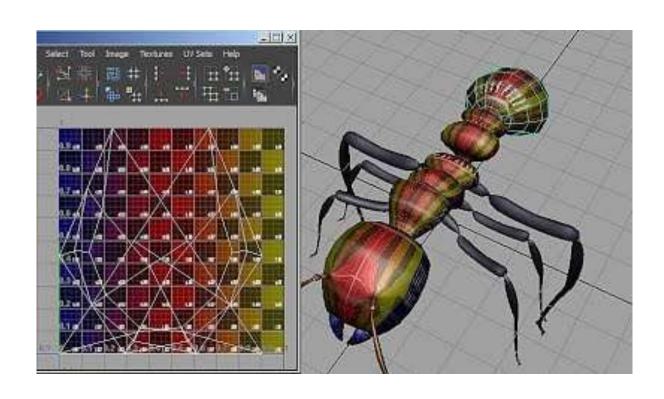


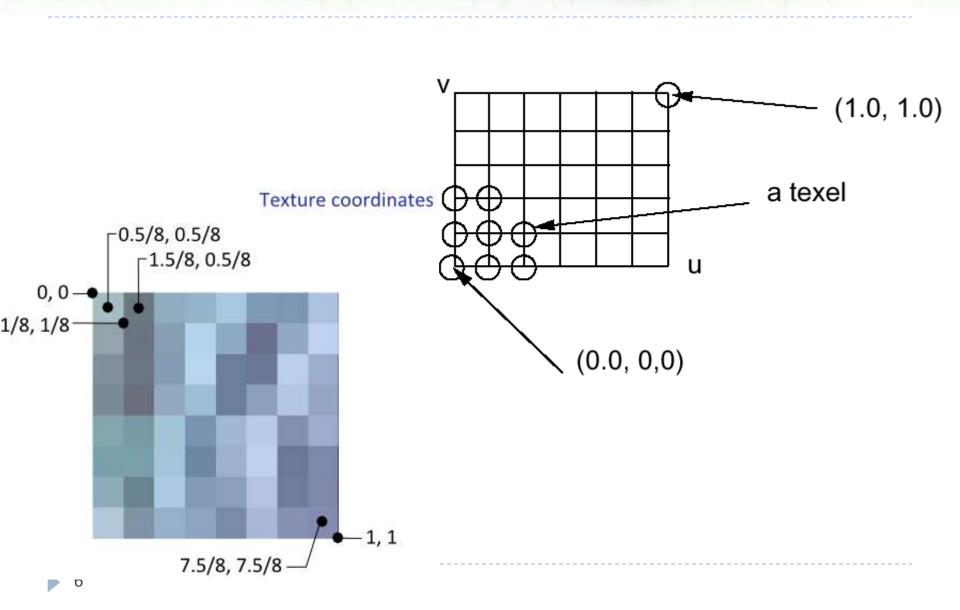


Текстура







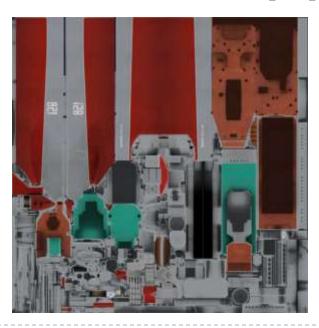


#### Виды текстур

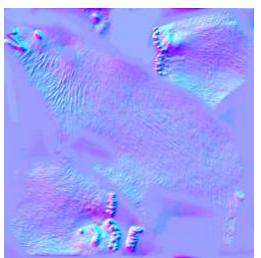
- ▶ Двумерные 2D, Rect
- ▶ Одномерные 1D
- ▶ Трехмерные volume, 2d array, cube

#### 2D текстуры

- > Задание свойств поверхности
  - ▶ Отражающая способность/цвет/albedo
  - ▶ Степень бликовости
  - Карта неровностей/карта нормалей
- Rect специальные тип текстур с адресацией вне диапазона [0;1]

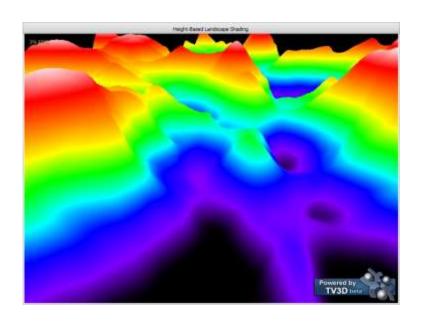






## 1D текстуры

- ▶ LUT фильтрация
- ▶ Кодирование функций
- Высотная раскраска

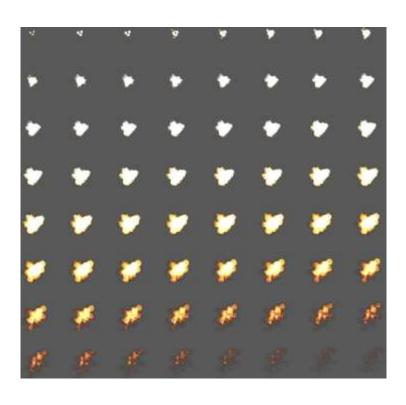


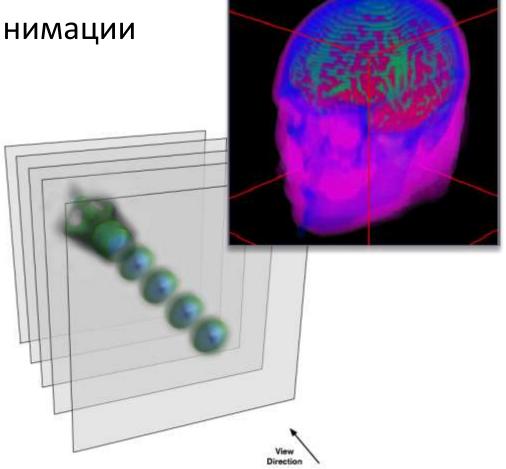


## 3D/volume текстуры

Задание воксельных моделей

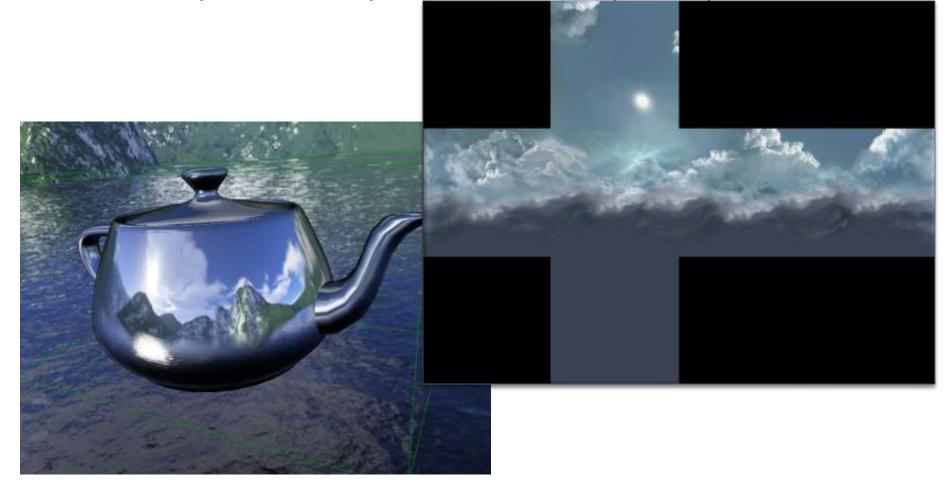
> Хранение покадровой анимации





# Cubemap текстура

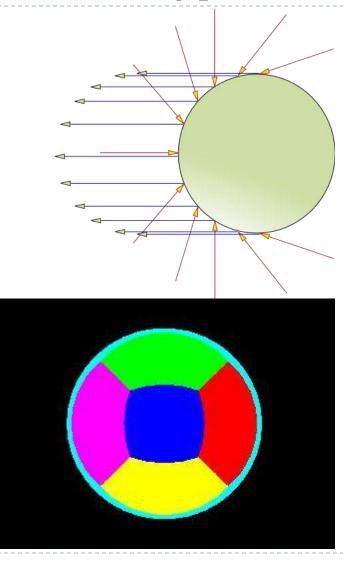
Используются для представления карты отражений



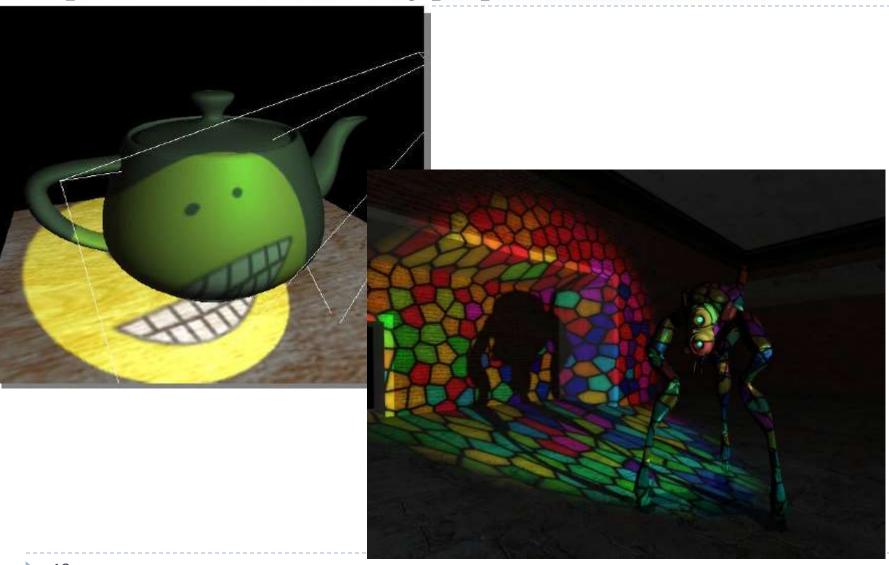
#### Ещё примеры использования текстур

Сферические карты отражений



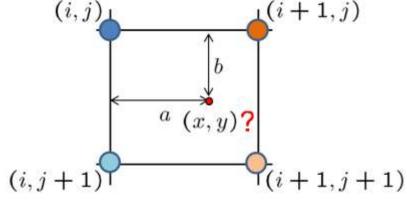


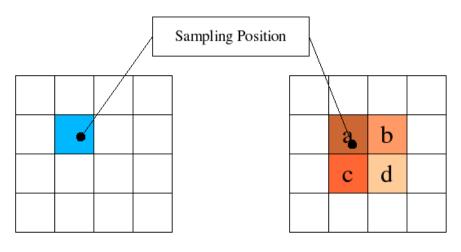
# Проективное текстурирование



#### Сэмплирование

- Вычисление значения текстуры по нецелым текстурным координатам
- Nearest ближайший
- ▶ Bilinear взвешенная сумма соседей

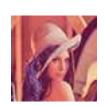


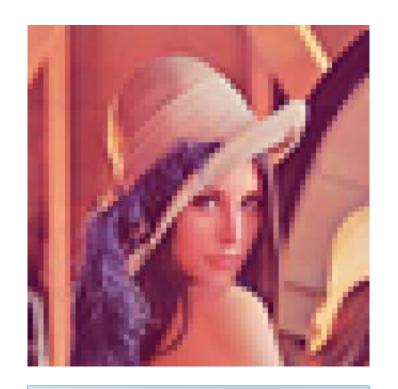


$$f(x,y) = (1-a)(1-b) f[i,j] +a(1-b) f[i+1,j] +ab f[i+1,j+1] +(1-a)b f[i,j+1]$$

Bilinear interpolation

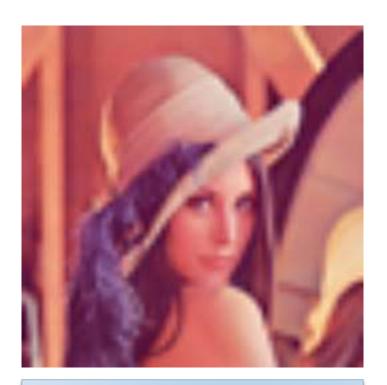
# Сэмплирование, nearest vs bilinear





Исходное изображение

Nearest фильтрация



Билинейная фильтрация

# Сэмплирование минифицированного изображения

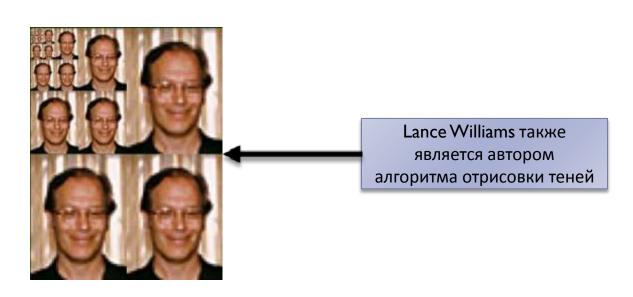
- Алиасинг, если пикселю на экране соответствует больше одного текселя из текстуры
- ► MIP(multum in parvo) фильтрация
  - Дополнительное хранение до log(n) слоев на текстуру,  $Resolution(level) = \frac{Resolution(level-1)}{2}$
  - Уровень выбирается в зависимости от «размера» текселя в экране
  - Размер текселя аппроксимируется выражением:

$$\rho(x,y) = \max\left(\sqrt{\left(\frac{\partial s}{\partial x}\right)^2 + \left(\frac{\partial s}{\partial y}\right)^2}, \sqrt{\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial y}\right)^2}\right)$$

Дополнительные  $\log(n)$  уровней

#### Размер МІР текстуры

- $S = S_0 = 2^a \times 2^a$  размер текстуры/размер нулевого уровня
- $S_i = \frac{S_{i-1}}{4}$  размер
- $S_{MIP} = \sum_{i=0}^{a} S_i = \frac{4}{3} S_0$



#### NB! Условное сэмплирование на шейдере

Недетерминированное поведение

```
vec3 color;

if (condition)
color = texture(sampler1, st);

else
color = texture(sampler2, st);
```

Некорректные расчет MIP уровня

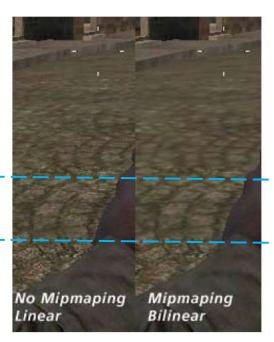
```
vec3 color;

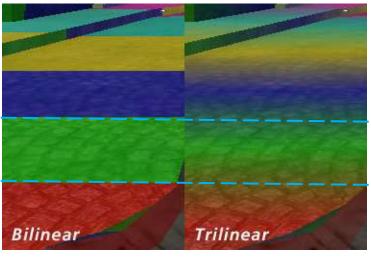
if (condition)
color = texture(sampler1, st1);

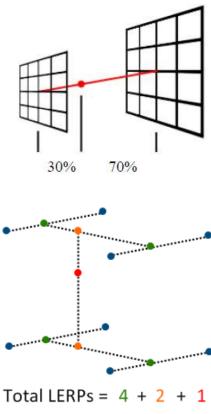
else
color = texture(sampler1, st2);
```

# Trilinear фильтрация

- Разрывы между соседними уровнями
- Линейная фильтрация между билинейноинтерполированными значениями уровней

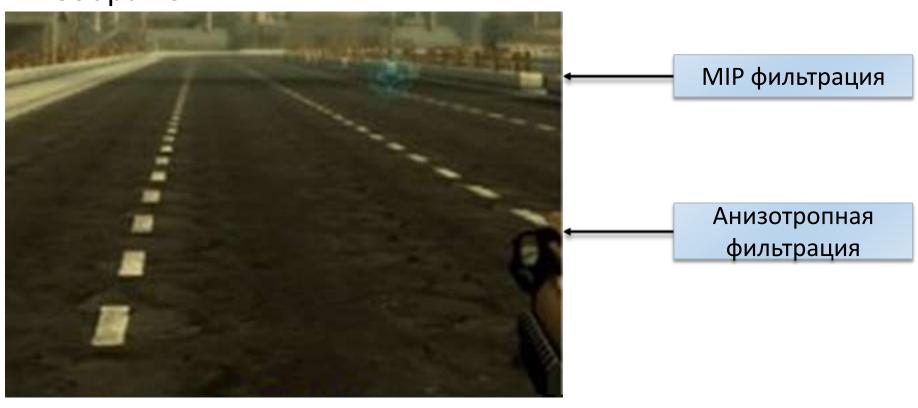






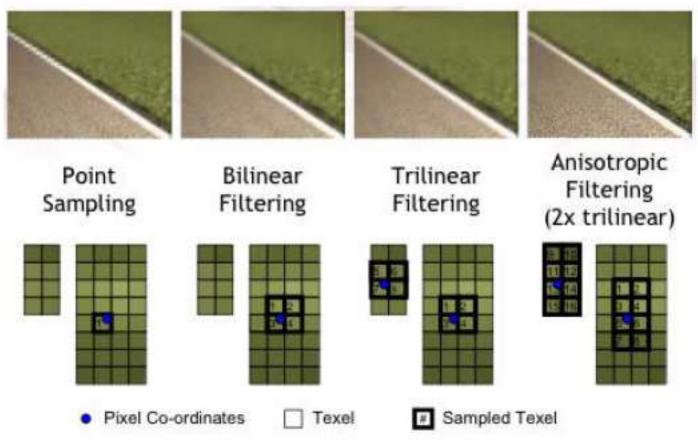
#### Анизотропная фильтрация

MIP фильтрация приводит к «замыливанию» изображения

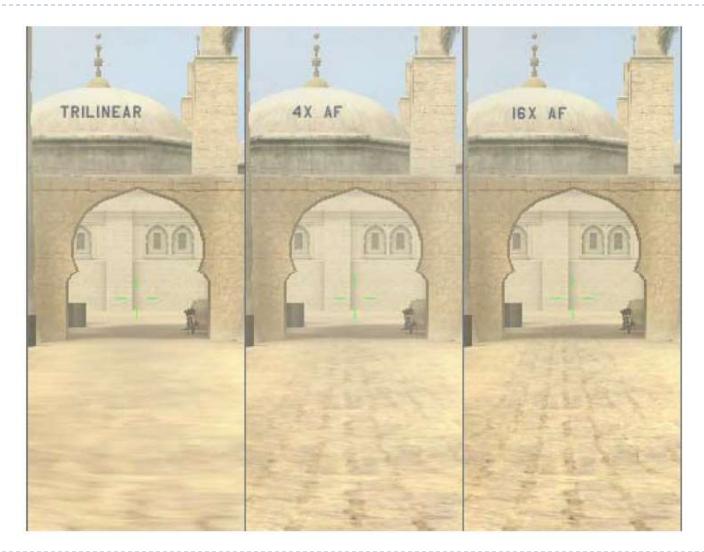


#### Анизотропное сэмплирование

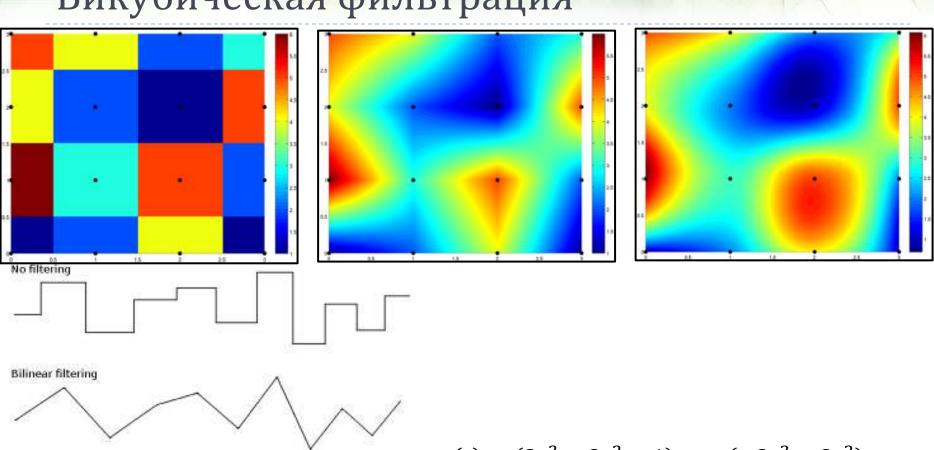
- Проекция пикселя в текстуру
- Аппроксимация по большему количество сэмплов

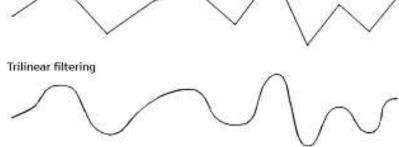


# Сравнение трилинейной и анизотропной фильтрации



# Бикубическая фильтрация



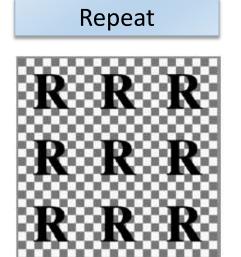


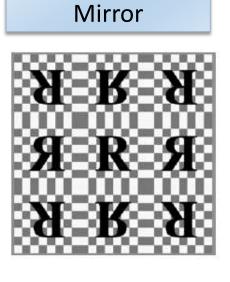
$$p(t) = (2t^3 - 3t^2 + 1)p_0 + (-2t^3 + 3t^2)p_1$$

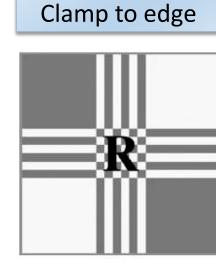
Интерполяция сплайном Эрмита с нулевыми производными

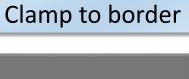
# Texture wrapping

#### Аппаратно-поддерживаемые режимы





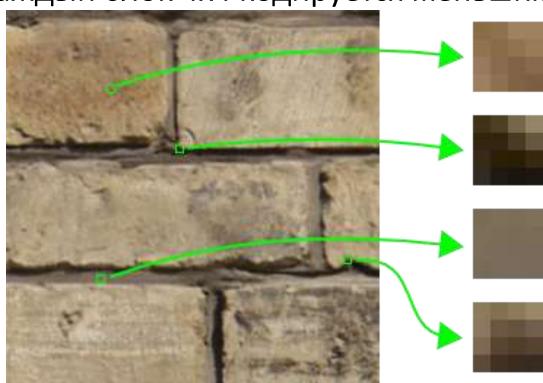






#### Аппаратное сжатие текстур

- ▶ BC Block compression
- Используется факт однородности соседних текселей
- Каждый блок 4х4 кодируется меньшим кол-вом бит



#### BC1

2 Bytes

MSB LSB color 0 R[15:11] G[10:5] B[4:0] color\_1

1 Byte

$$color_0 = 00$$
  
 $color_1 = 01$   
 $color_2 = 10$   
 $color_3 = 11$ 

#### Uncompressed

8 Байт на блок из 16 RGB пикселей вместо 48







 $color_{2} = \frac{2}{3}color_{0} + \frac{1}{3}color_{1}$  $color_{3} = \frac{1}{3}color_{0} + \frac{2}{3}color_{1}$ 





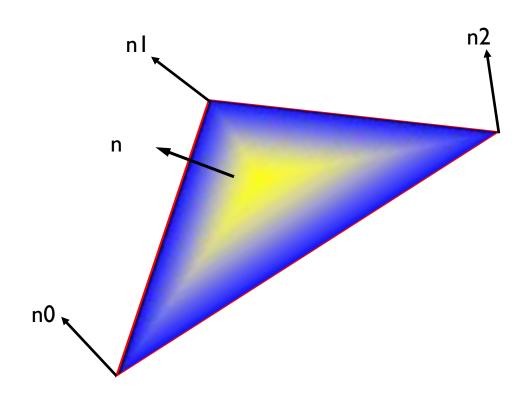


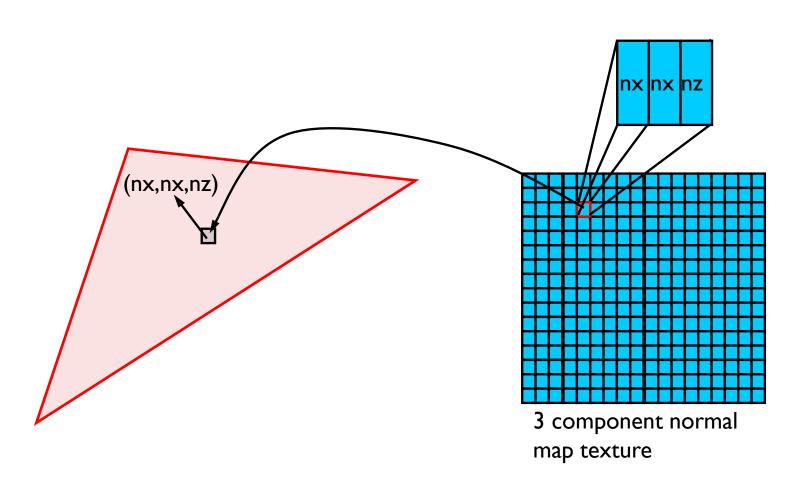




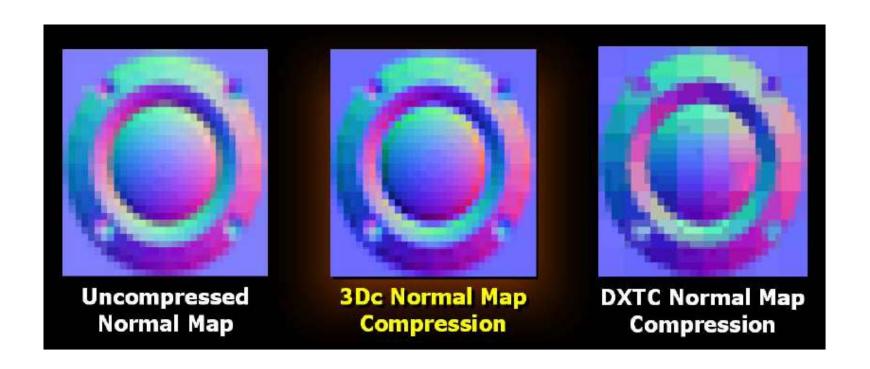


# Текстура нормалей





8b x 8b z 8b x	8b - 8b - 8b y	8b x 8b z 8b x	8b y 8b - 8b y	8b x 8b z 8b x	8b y 8b - 8b y	8b x 8b z 8b x	8b y 8b - 8b y			X <sub>mi</sub>	+	1011	+ 1. N. C. S.	- 1/16°	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Solly T X <sub>max</sub>		
8b z	8b -	8b z	8b -	8b z														
8b x 8b z	8b y 8b -	8b x 8b z	8b y 8b	8b x 8b z	8b y 8b -	8b x 8b z	8b y 8b -		d	⊢ y <sub>mi</sub>	-				<del>   </del>	y <sub>max</sub>		
8b x 8b z	8b y 8b -	8b x 8b z	8b y 8b -	8b x 8b z	8b y 8b -	8b x 8b z	8b y 8b -	3b ×			3b 3	10.10.00	3b У	х	3b y	8b x <sub>min</sub>	8b y <sub>min</sub>	
					,			3b ×	S. com	seesa B	3b У	20000	¦ 3Ь ¦ У	342.92	3b . y	8b x <sub>max</sub>	8b	
				X				3b X		2002	3b y	1.02.00	3b   y	3b X	3b y			
		¥		3				3b X			3b 3		¦ 3b ¦ y		3b y			



# Вопросы

