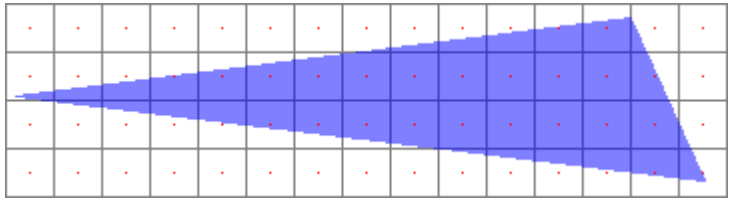
Компьютерная графика и визуализация в реальном времени

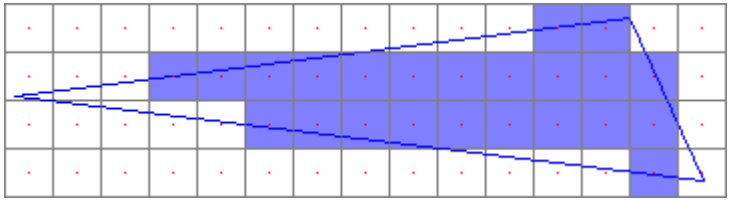
Anti-aliasing, relief mapping

Алексей Романов

Антиалиасинг при растеризации

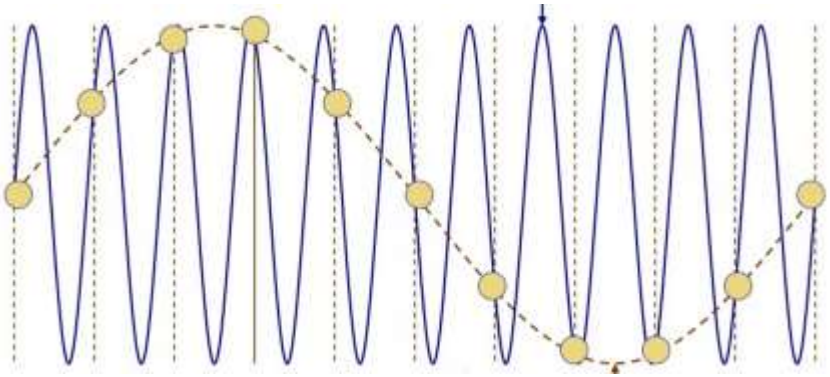
Растеризация по центрам пикселей



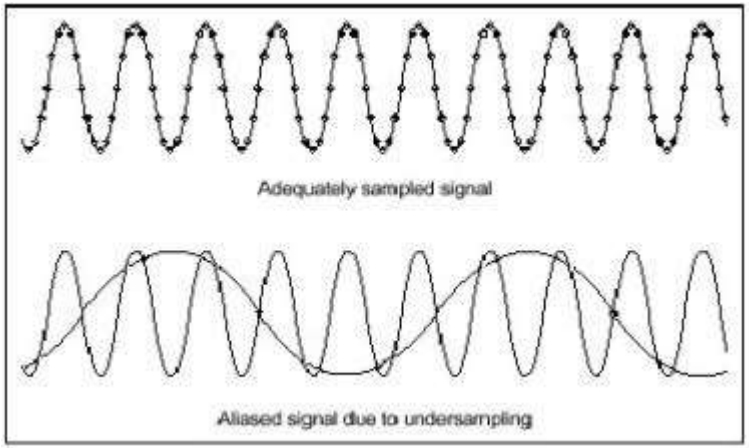


Сэмплирование

- Алиасинг являеся проявлением некорректного сэмплирования
- Условие корректного сэмплирования сигнала s(x): $N > 2f_{max}, f_{max}$ - масимальная частота s(x) при спектральном представлении



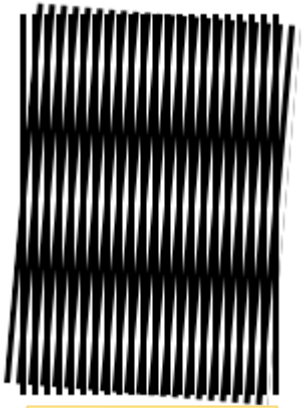
Некорректное восстановление сигнала



Сэмплирование

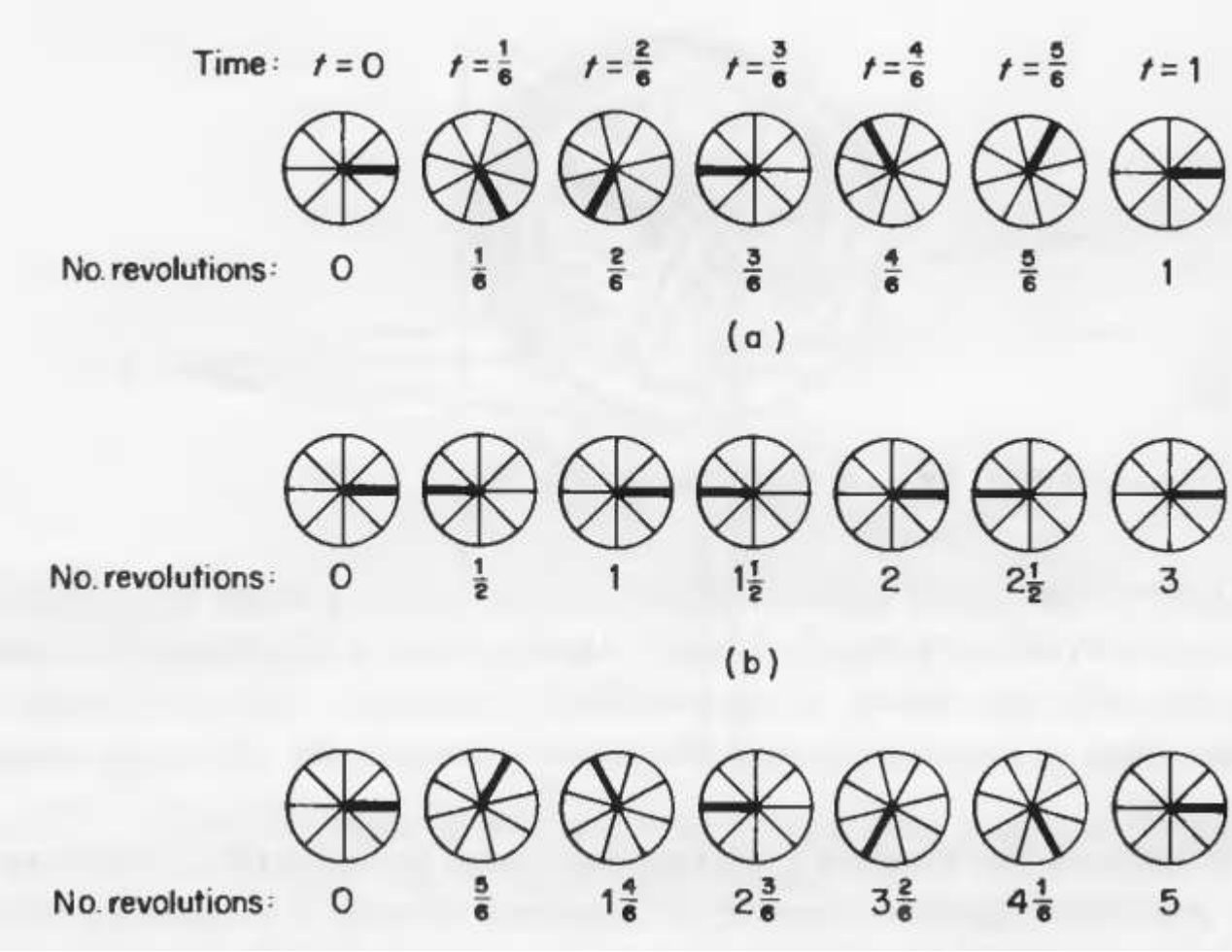
Некорректное сэмплирование

Сэмплирование с корректной частотой

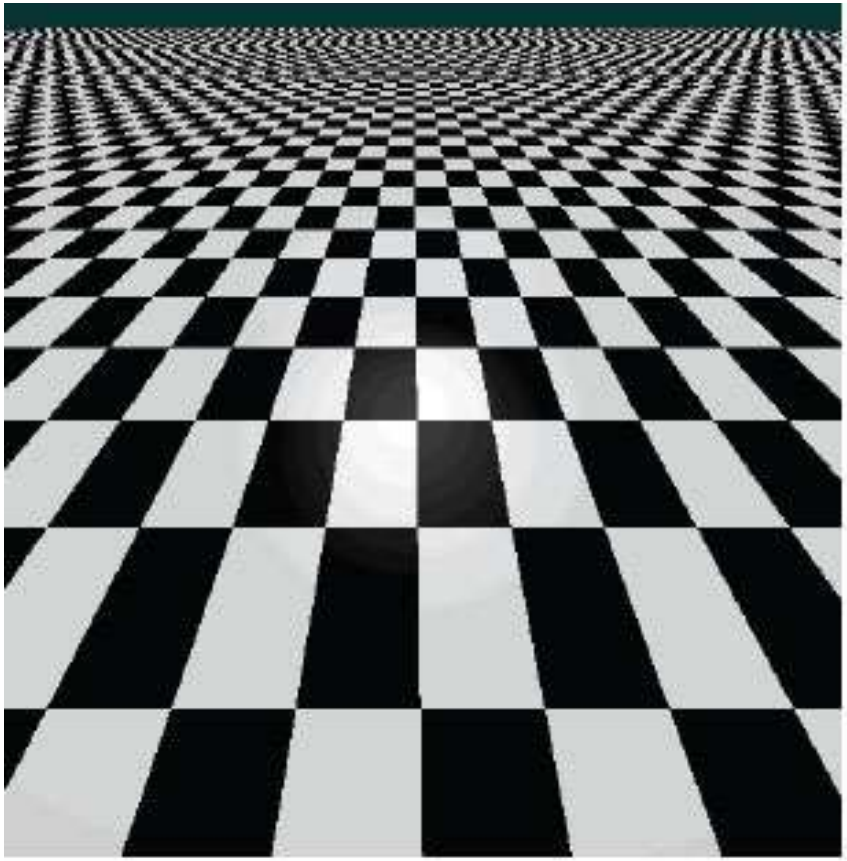


Муар

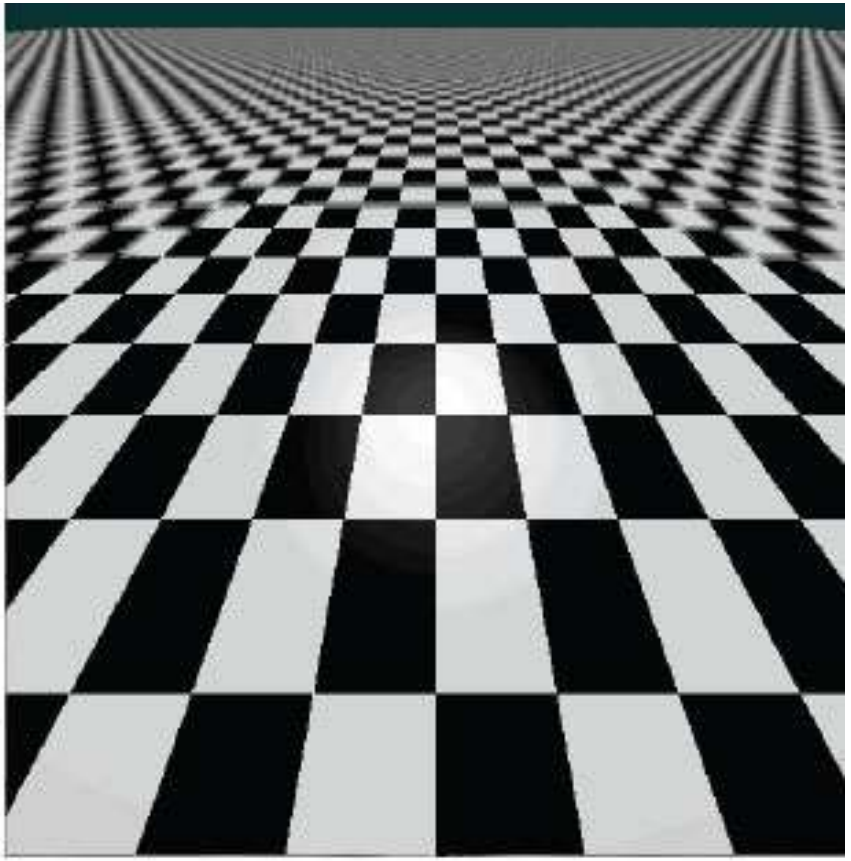
Временной алиасинг



Пространственный алиасинг



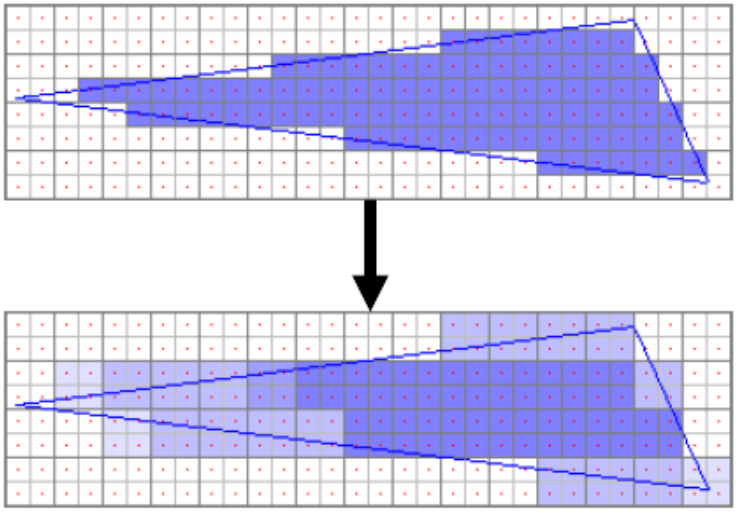
aliasing effects



anti-aliasing by over-sampling

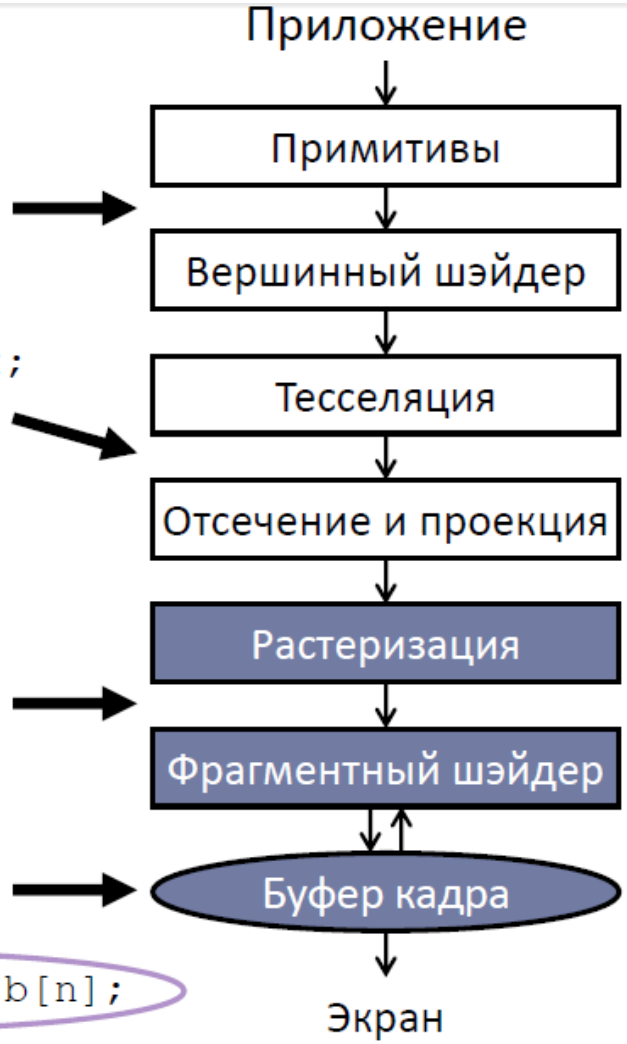
Supersampling

- Повышение частоты сэмплирования
- Поддерживается аппаратно
- Большая нагрузка на растеризатор



Multisampling

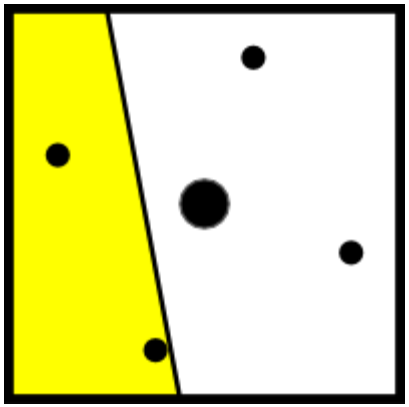
```
struct {
  float x,y,z,w;
  float r,g,b,a;
} vertex;

struct {
  vertex v0,v1,v2;
} triangle;

struct {
  short x,y;
  bool mask[n];
  float depth[n];
  float r,g,b,a;
} fragment;

struct {
  short R,G,B;
  int depth[n];
  byte r[n],g[n],b[n];
} pixel;
```



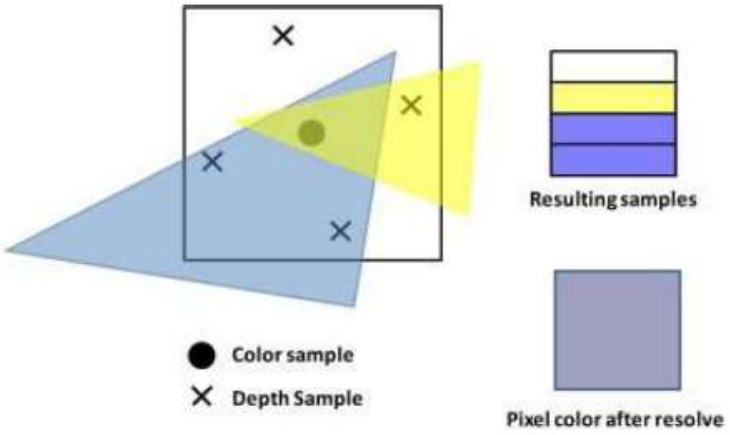
Точечная локализация по локальным позициям сэмплов устанавливает маски активных и вычисляет глубины. Цвет берётся из самого центрального сэмпла.

Фрагментарные сэмплы сливаются по глубине в буфер глубины, затем усредняются в буфер цвета кадра.

Centroid интерполянты



```
vertex shader:
centroid out color;

fragment shader:
centroid in color;
```

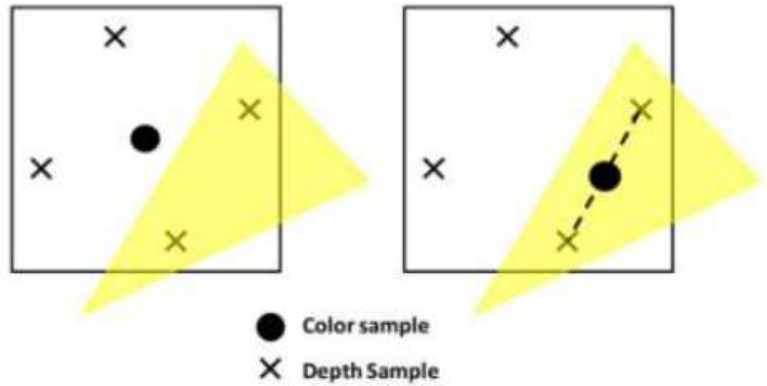


No centroid sampling

```
struct PS_INPUT
{
  float2 vTex : TEXCOORD1;
};
```

Centroid sampling

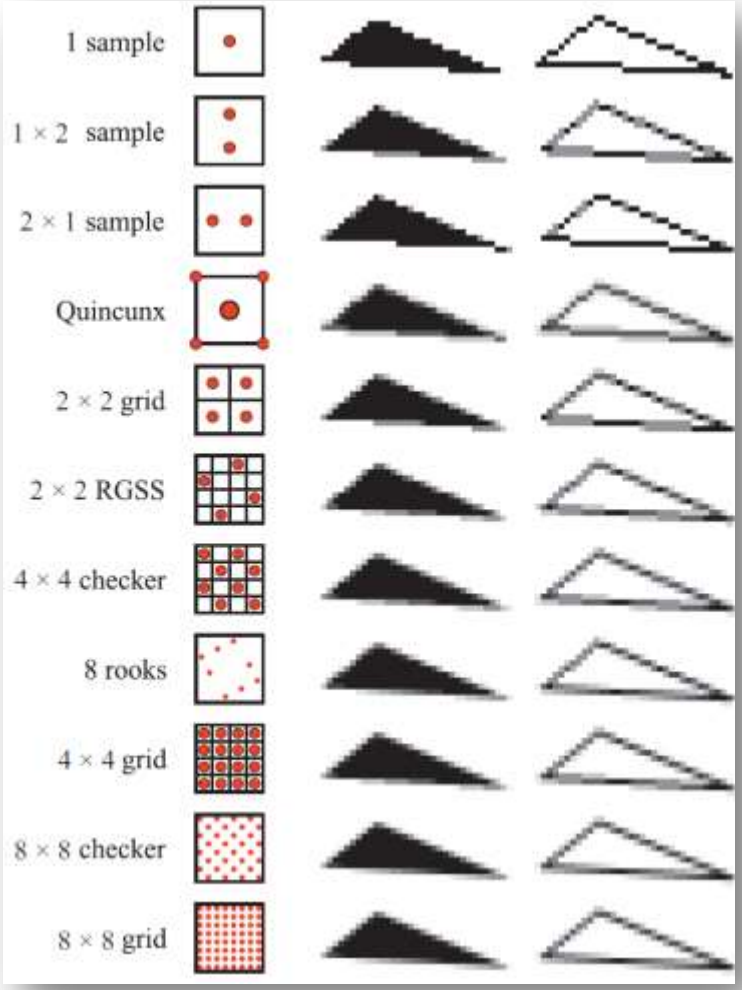
```
struct PS_INPUT
{
  float2 vTex : TEXCOORD1_CENTROID;
};
```



Per-sample shading

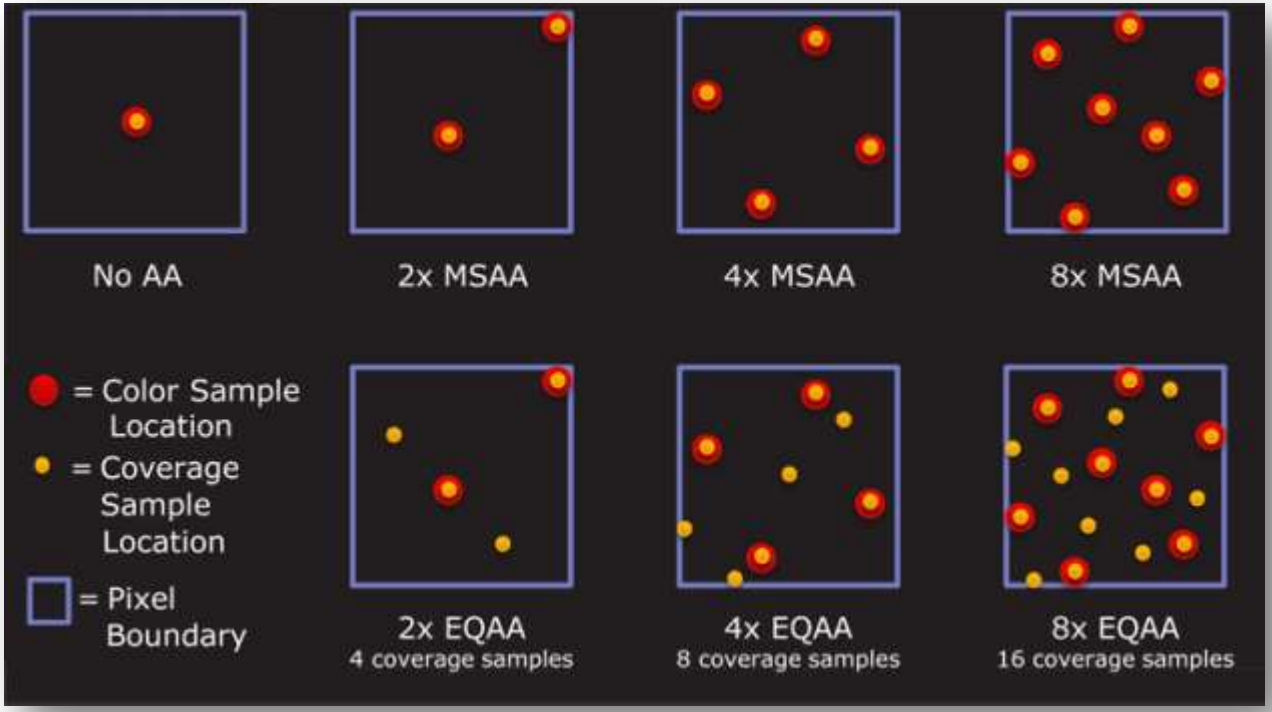
```
sample in variable;
```

Режимы MSAA


1 sample
1 × 2 sample
2 × 1 sample
Quincunx
2 × 2 grid
2 × 2 RGSS
4 × 4 checker
8 rooks
4 × 4 grid
8 × 8 checker
8 × 8 grid

CSAA (Coverage Sample Anti-Aliasing)

- Отделение color от coverage sample'ов



Supersampling

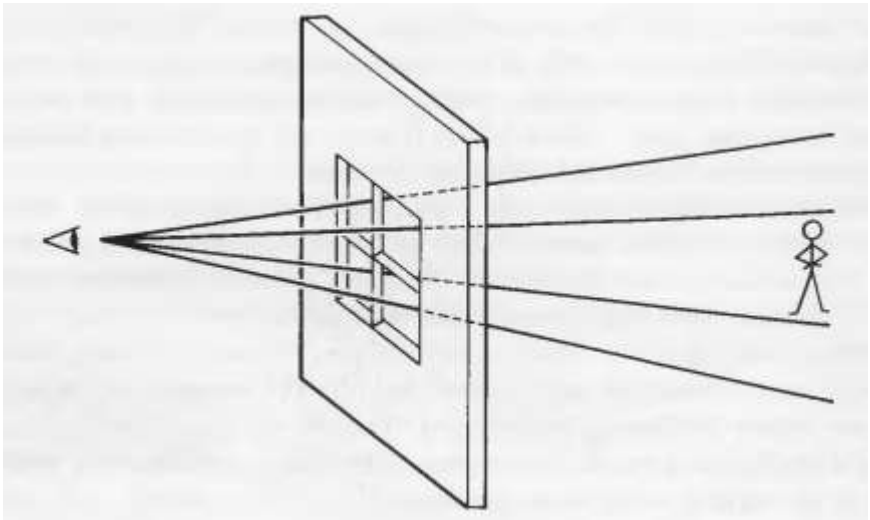
MSAA/CSAA неприменим для полупрозрачных объектов

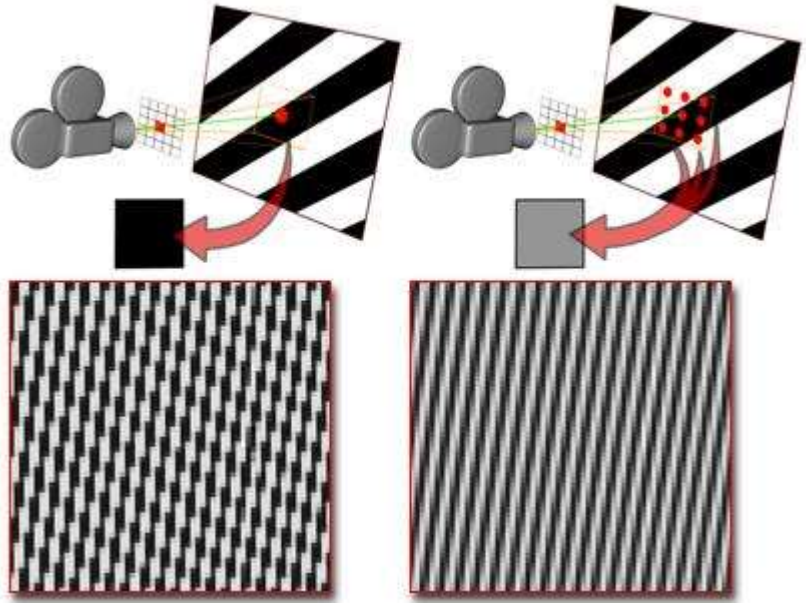
Уменьшение алиасинга

- *Supersampling*

  Сэмплирование большего количества фрагментов на один пиксель экрана
- Адаптивный *supersampling*
- Данный подход, к сожалению, не идеален

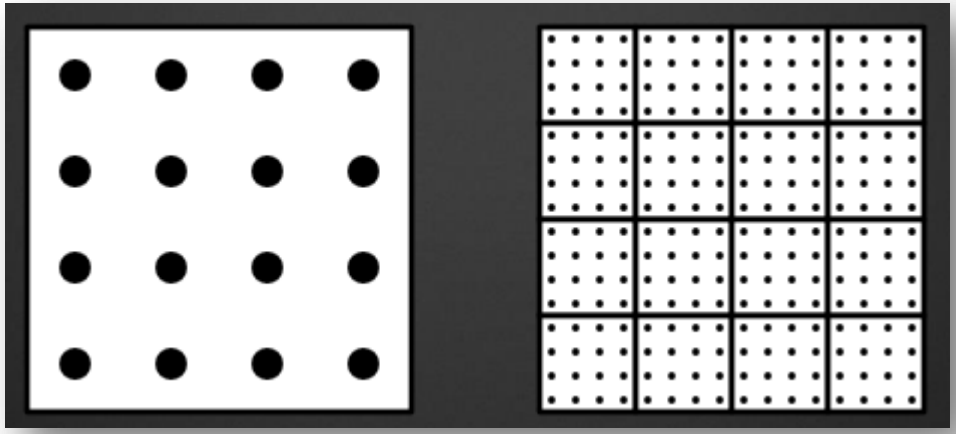
Pixel with sample postions





Jittering

- Не Real-time метод
- Подходит для генерации скриншотов/видео максимального качества
- Позволяет настраивать сколь-угодно большой уровень сглаживания
- Не требует дополнительной памяти (в отличие от supersampling)

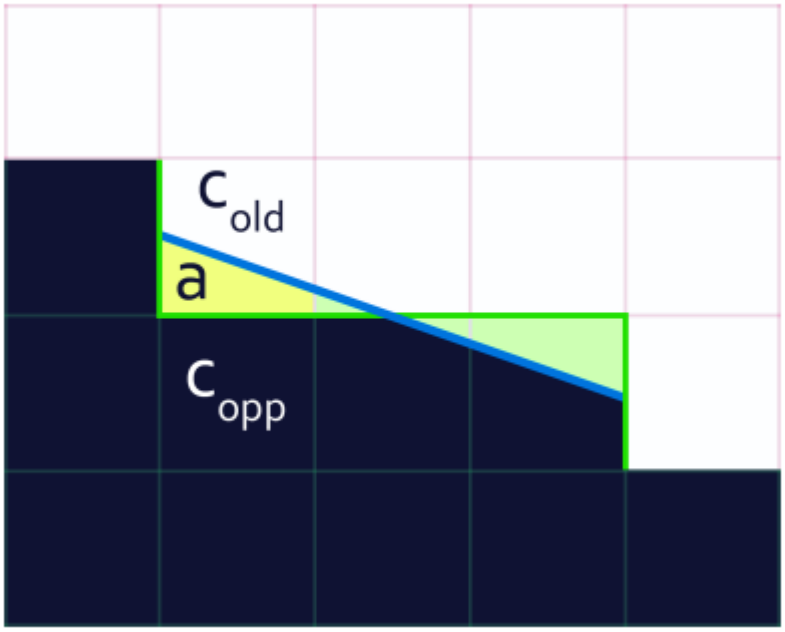


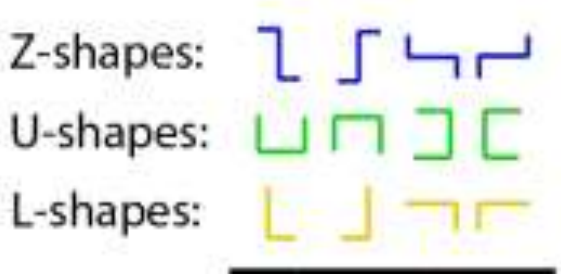
$$P' = M_{frustum}P = \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix}$$

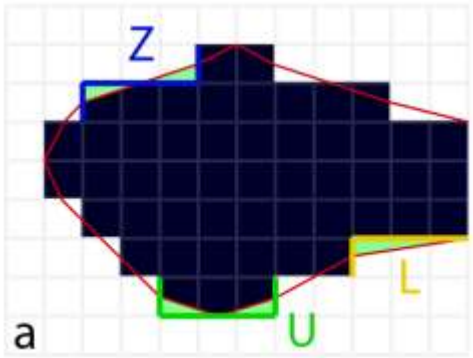
- $r^x = r + \frac{x}{N * width}, l^x = l + \frac{x}{N * width}$
- $t^y = t + \frac{i}{N * width}, b^y = b + \frac{y}{N * height}$

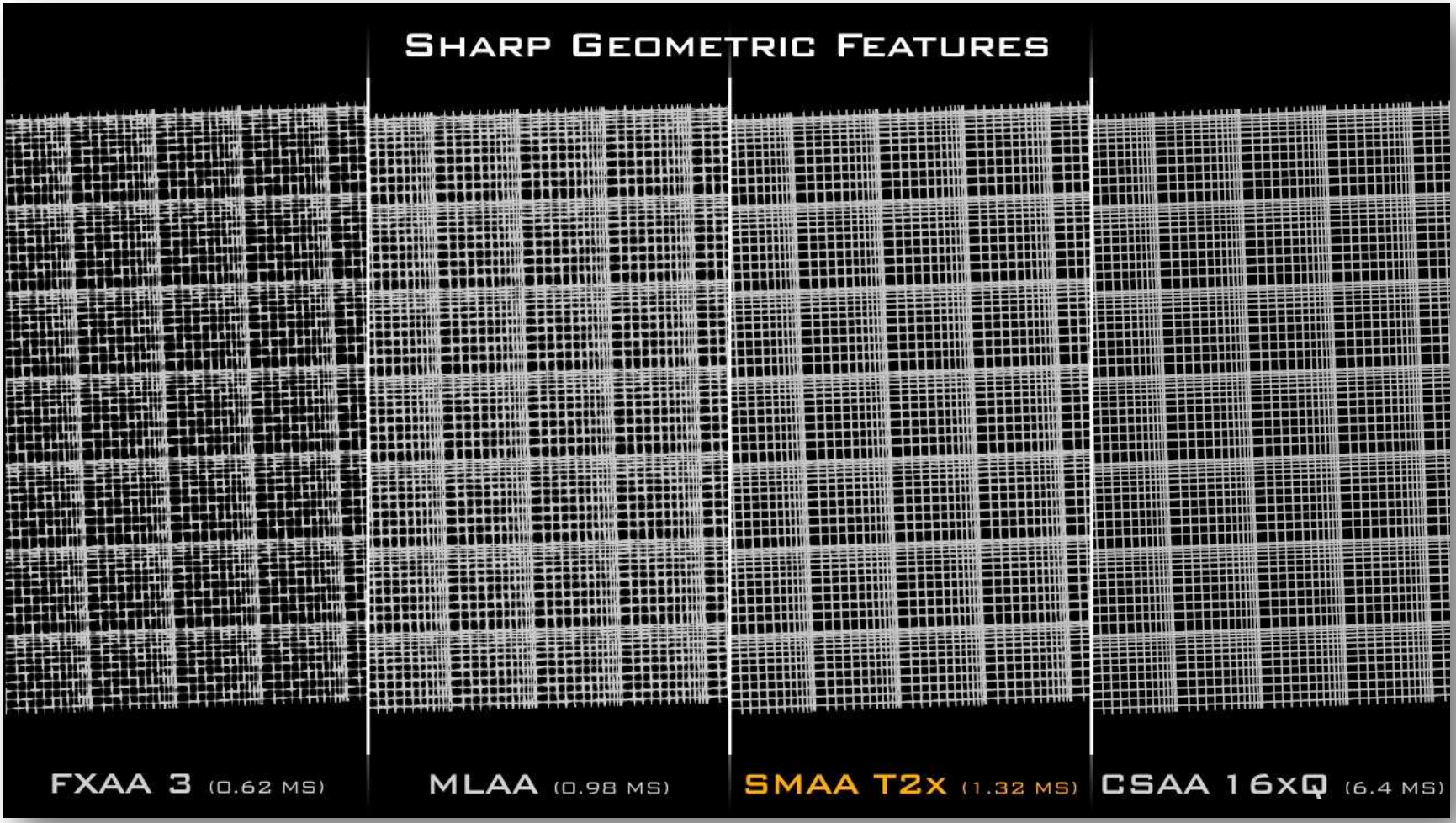
SMAA, MLAA, FXAA

- Subpixel, Morphological, Fast approXimate
- Применяются в качестве post-effect'а, после отрисовки сцены





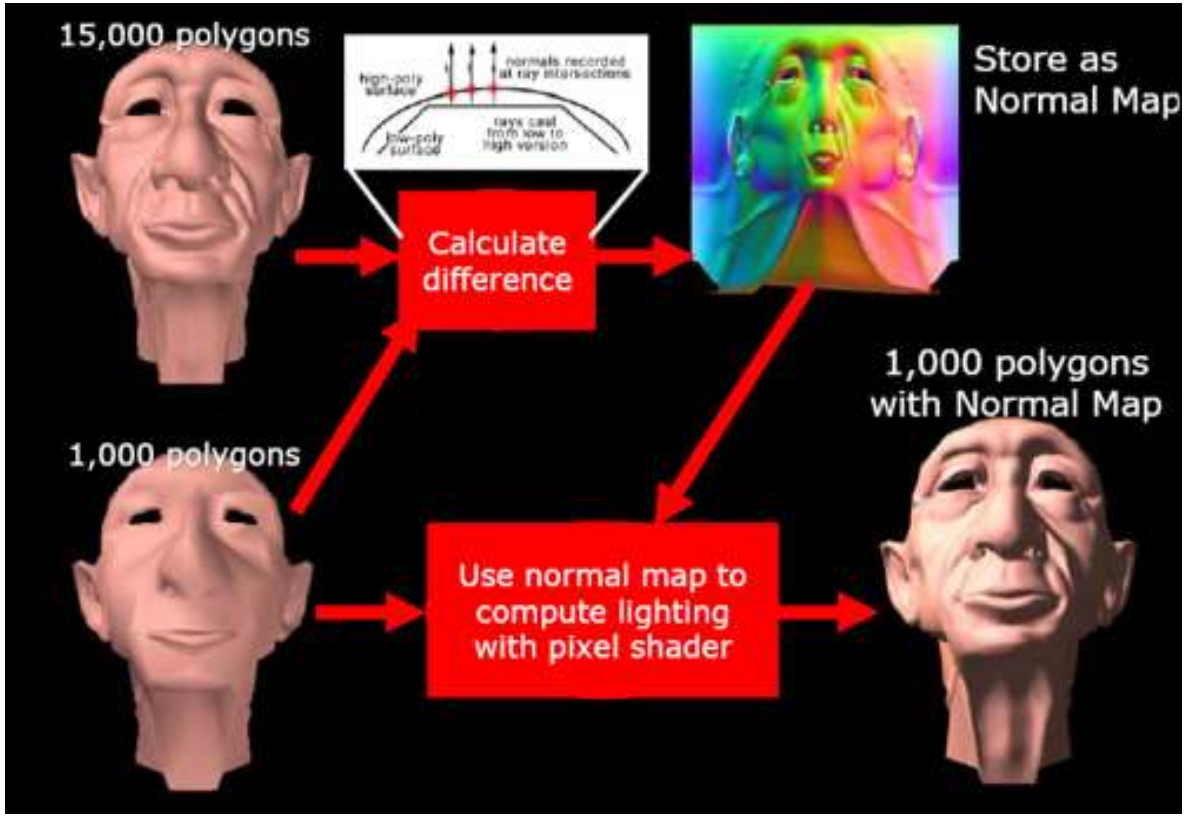



SHARP GEOMETRIC FEATURES
FXAA 3 (0.62 MS)
MLAA (0.98 MS)
SMAA T2x (1.32 MS)
CSAA 16xQ (6.4 MS)

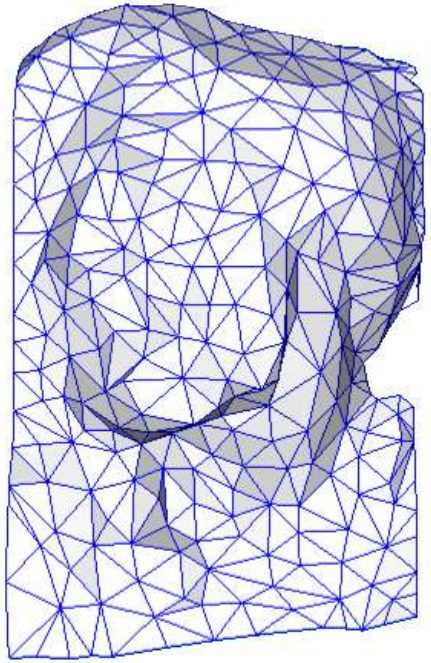
Итог

- SSAA
  - поддерживается аппаратно
  - нагружает растеризатор и память
- MSAA, CSAA
  - Поддерживается аппаратно
  - Требователен к памяти
  - Работает только на границах геометрии
- SMAA, MLAA, FXAA, …
  - Применяются пост-эффектом
  - Не работают для объектов меньше пикселя

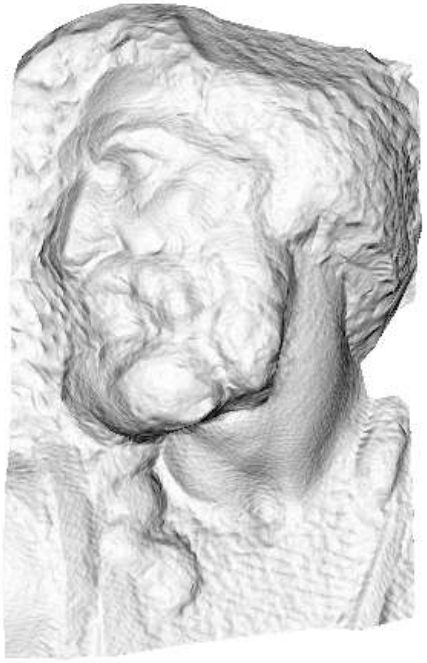
Bump mapping



original mesh
4M triangles

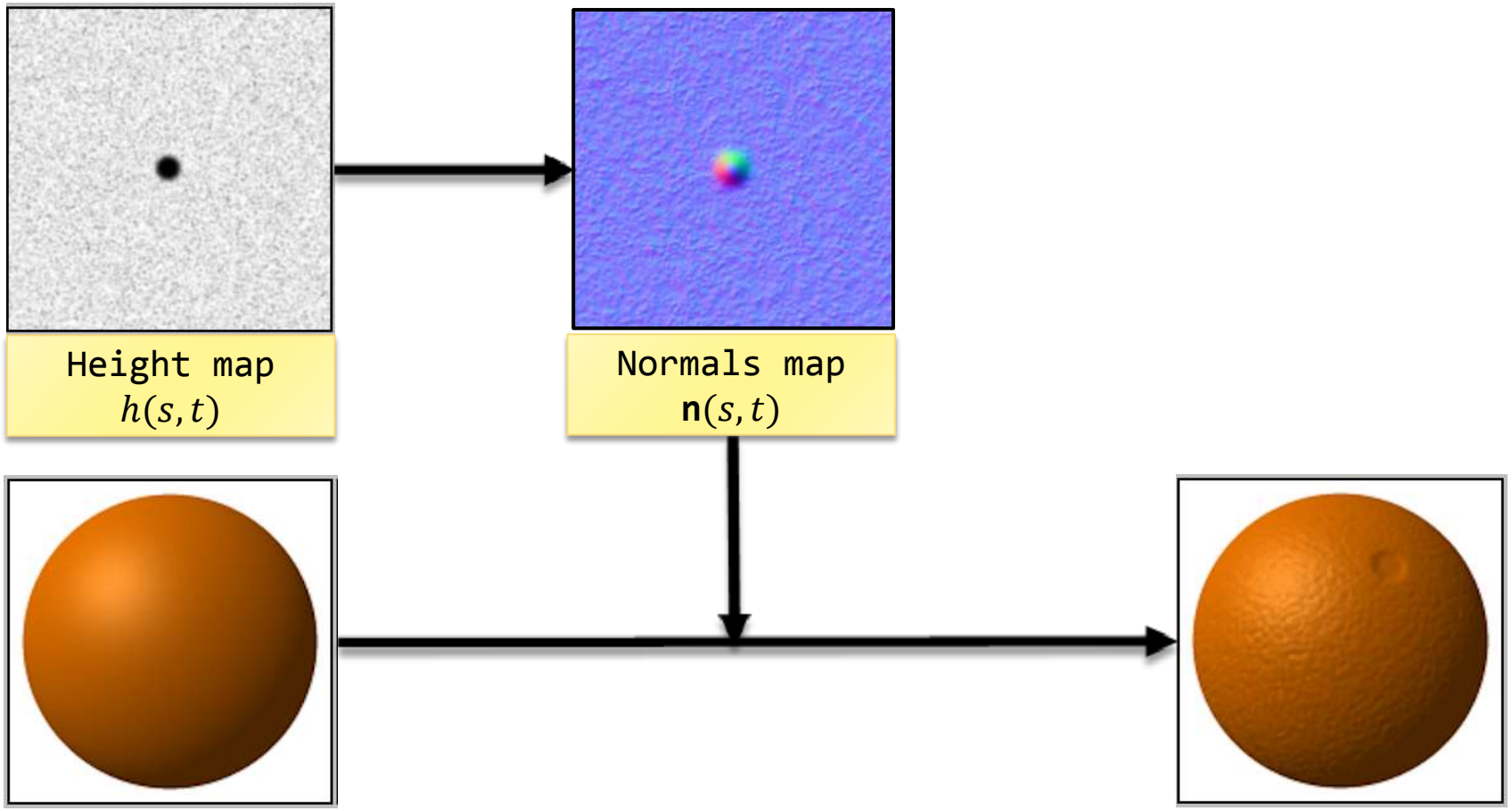


simplified mesh
500 triangles

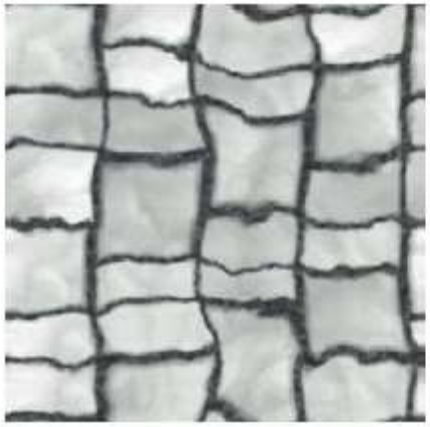


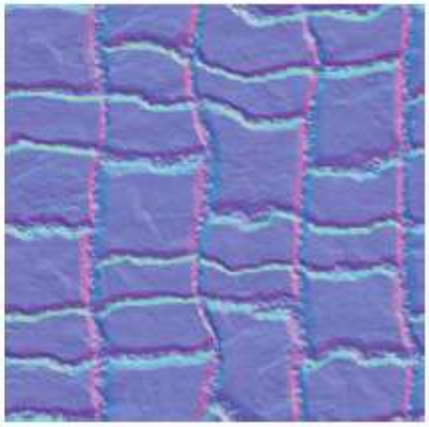
simplified mesh
and normal mapping
500 triangles

Bump mapping



Вычисление нормали

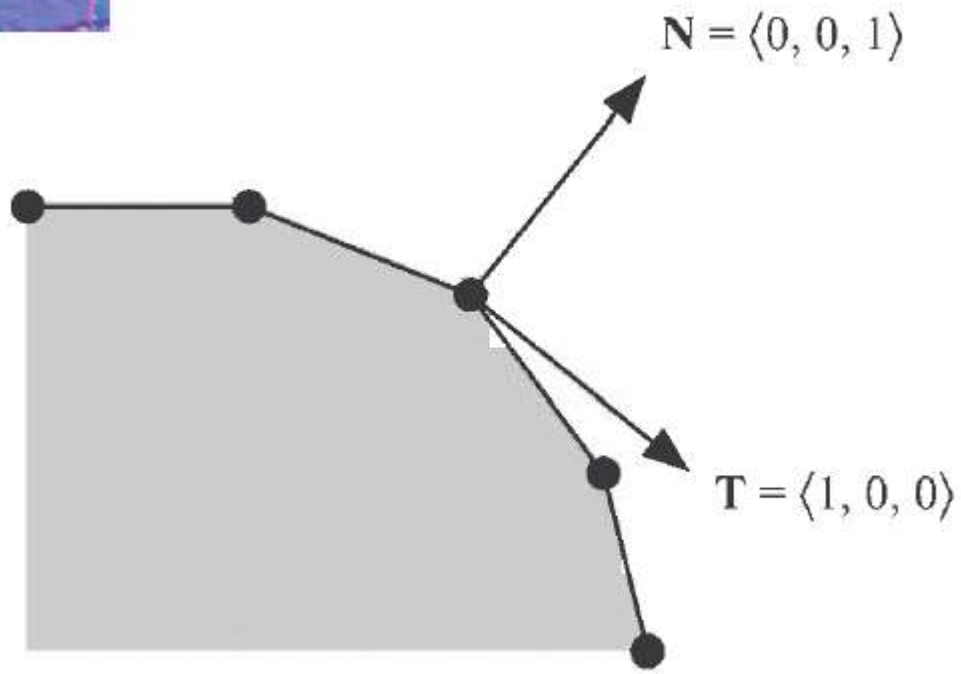




$$\mathbf{S}(i,j) = \langle 1, 0, aH(i+1,j) - aH(i-1,j) \rangle$$

$$\mathbf{T}(i,j) = \langle 0, 1, aH(i,j+1) - aH(i,j-1) \rangle$$

$$\mathbf{N}(i,j) = \frac{\mathbf{S}(i,j) \times \mathbf{T}(i,j)}{\|\mathbf{S}(i,j) \times \mathbf{T}(i,j)\|} = \frac{\langle -S_z, -T_z, 1 \rangle}{\sqrt{S_z^2 + T_z^2 + 1}}$$



Построение касательного базиса

$$\mathbf{Q} - \mathbf{P}_0 = (s - s_0)\mathbf{T} + (t - t_0)\mathbf{B}$$

$$\mathbf{T}' = \mathbf{T} - (\mathbf{N}\cdot\mathbf{T})\mathbf{N}$$
$$\mathbf{B}' = \mathbf{B} - (\mathbf{N}\cdot\mathbf{B})\mathbf{N} - (\mathbf{T}'\cdot\mathbf{B})\mathbf{T}'.$$

$$\mathbf{Q}_1 = \mathbf{P}_1 - \mathbf{P}_0$$
$$\mathbf{Q}_2 = \mathbf{P}_2 - \mathbf{P}_0$$

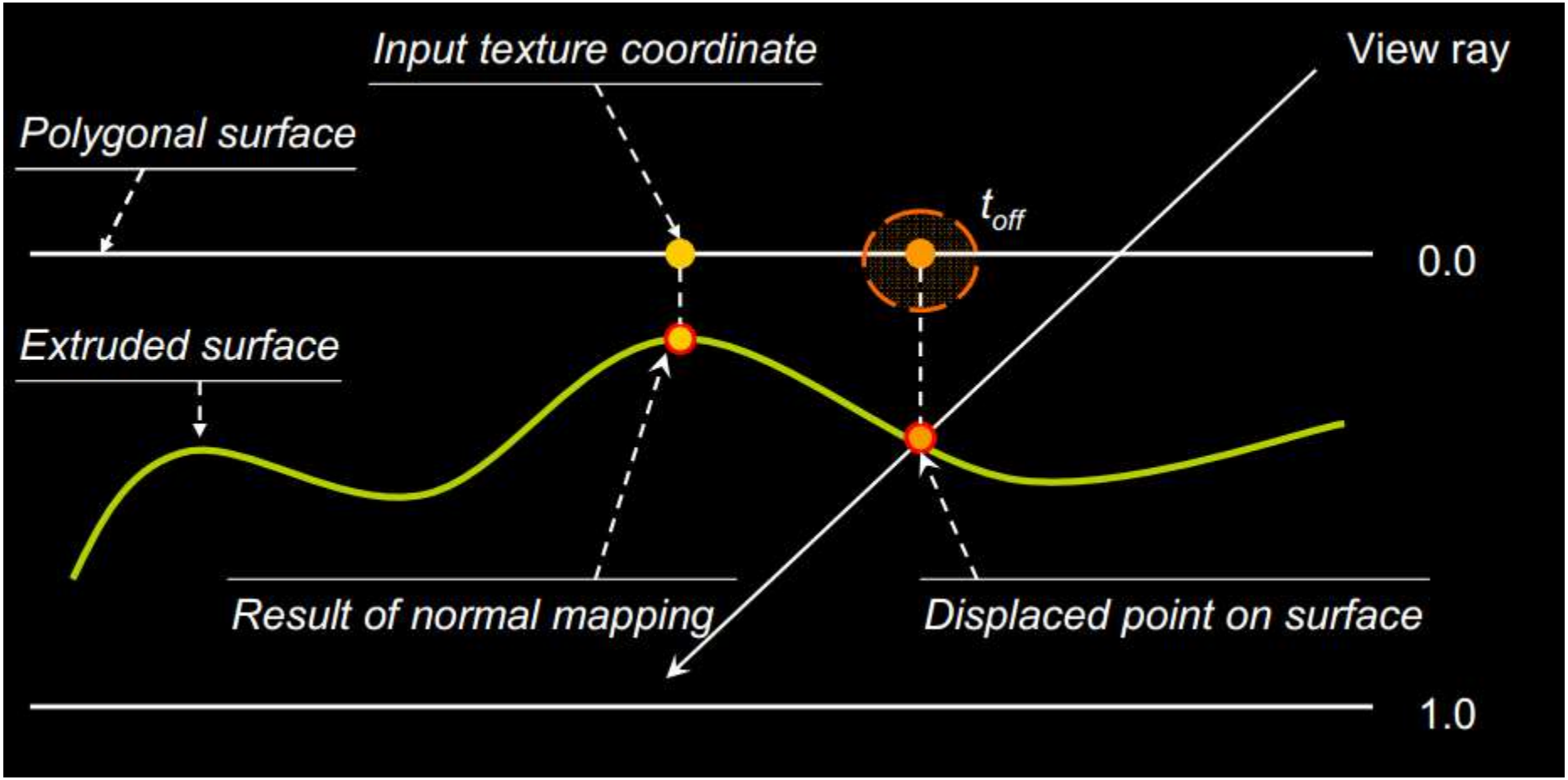
$$\langle s_1, t_1\rangle = \langle s_1 - s_0, t_1 - t_0\rangle$$
$$\langle s_2, t_2\rangle = \langle s_2 - s_0, t_2 - t_0\rangle$$

$$\mathbf{Q}_1 = s_1\mathbf{T} + t_1\mathbf{B}$$
$$\mathbf{Q}_2 = s_2\mathbf{T} + t_2\mathbf{B}$$

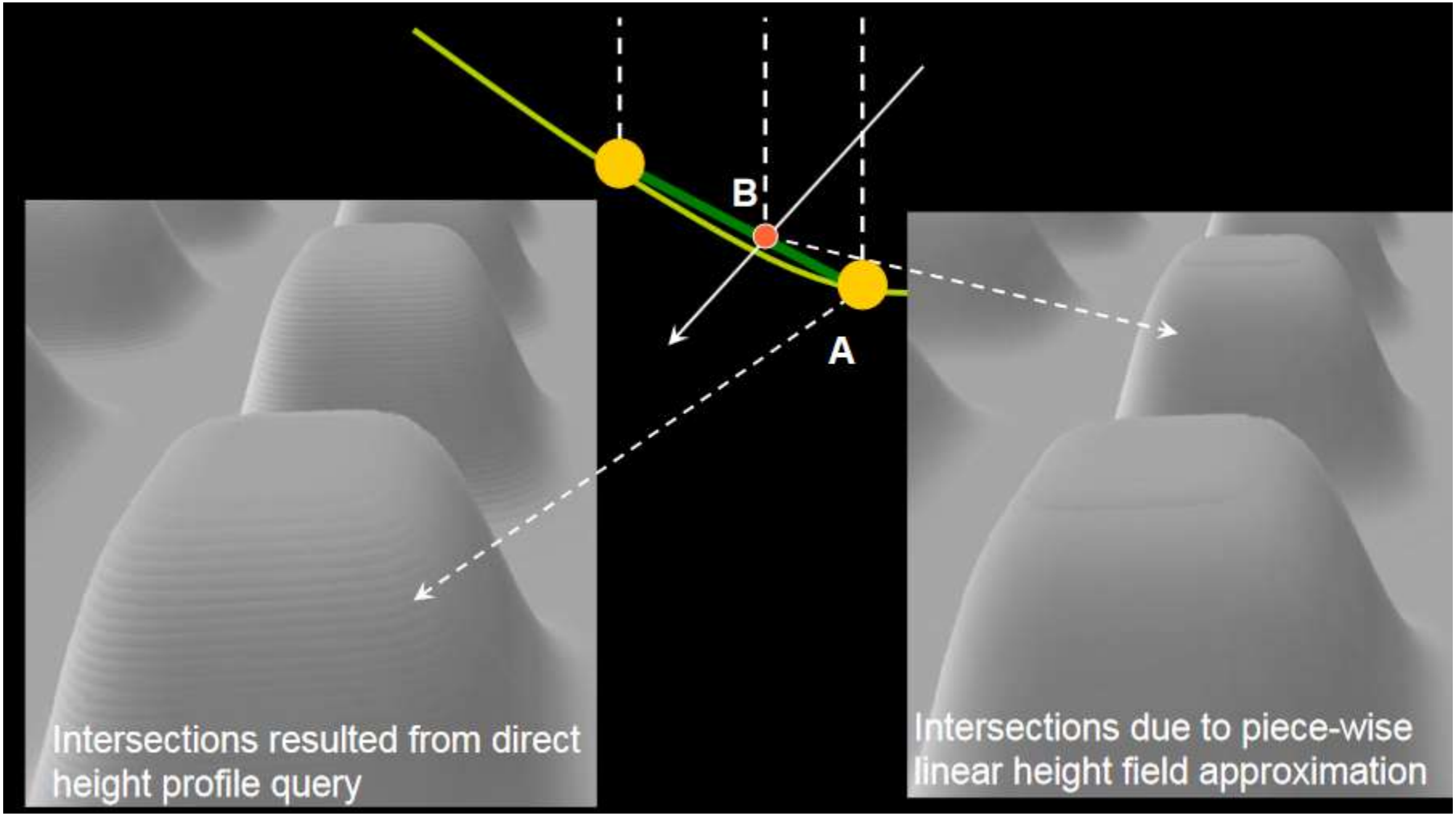
$$\begin{bmatrix} (\mathbf{Q}_1)_x & (\mathbf{Q}_1)_y & (\mathbf{Q}_1)_z \\ (\mathbf{Q}_2)_x & (\mathbf{Q}_2)_y & (\mathbf{Q}_2)_z \end{bmatrix} = \begin{bmatrix} s_1 & t_1 \\ s_2 & t_2 \end{bmatrix} \begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \end{bmatrix}$$

$$\begin{bmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \end{bmatrix} = \frac{1}{s_1 t_2 - s_2 t_1} \begin{bmatrix} t_2 & -t_1 \\ -s_2 & s_1 \end{bmatrix} \begin{bmatrix} (\mathbf{Q}_1)_x & (\mathbf{Q}_1)_y & (\mathbf{Q}_1)_z \\ (\mathbf{Q}_2)_x & (\mathbf{Q}_2)_y & (\mathbf{Q}_2)_z \end{bmatrix}.$$

Parallax occlusion mapping



Вычисление пересечения с картой высот



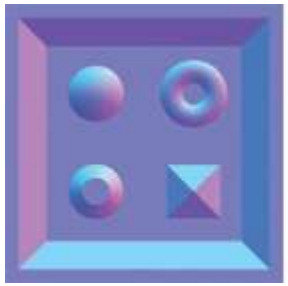
Parallax occlusion mapping

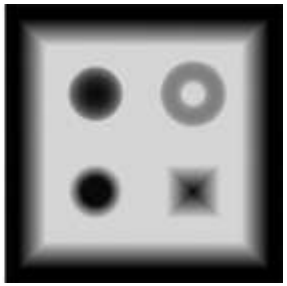
Parallax occlusion mapping, ~1 500 polygons

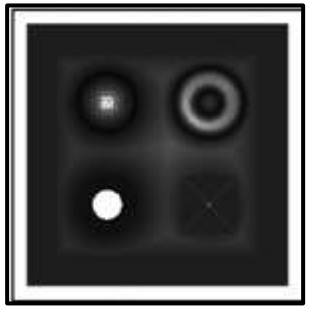
Diffuse lighting, ~1 500 000 polygons

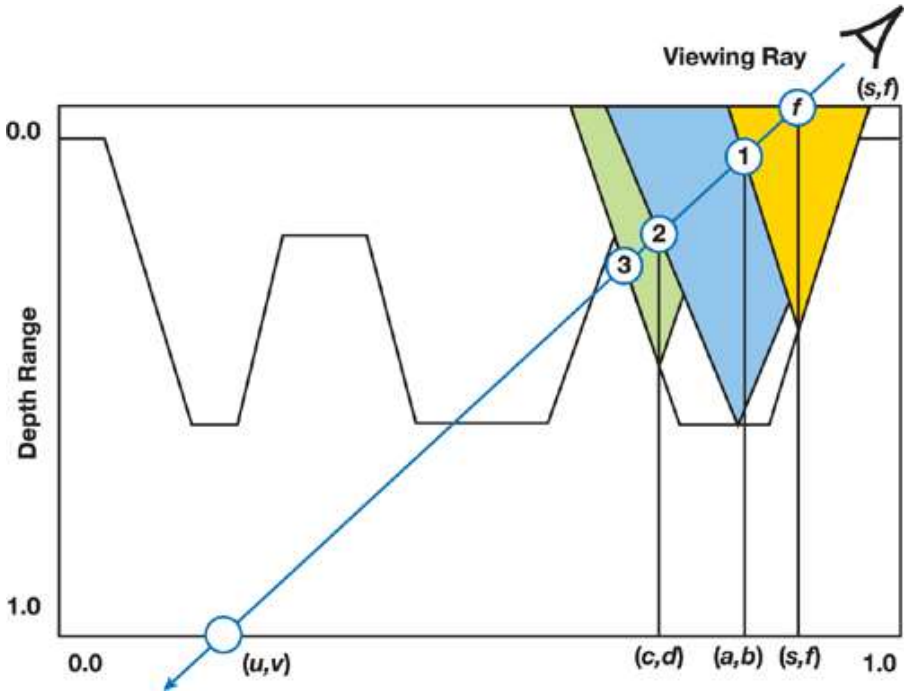
Cone mapping

- Parallax occlusion mapping требует большого количества операций
- Сокращение за счет предобработки
- Неприменимость для динамической карты высот



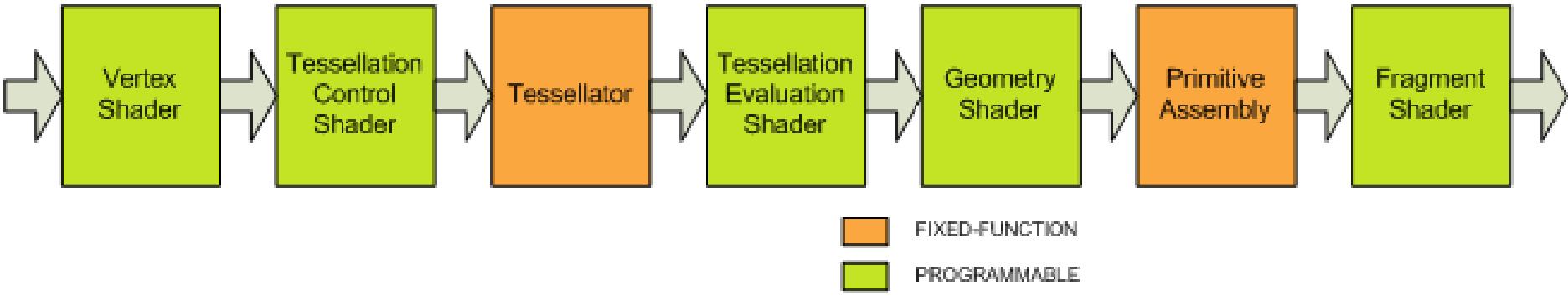






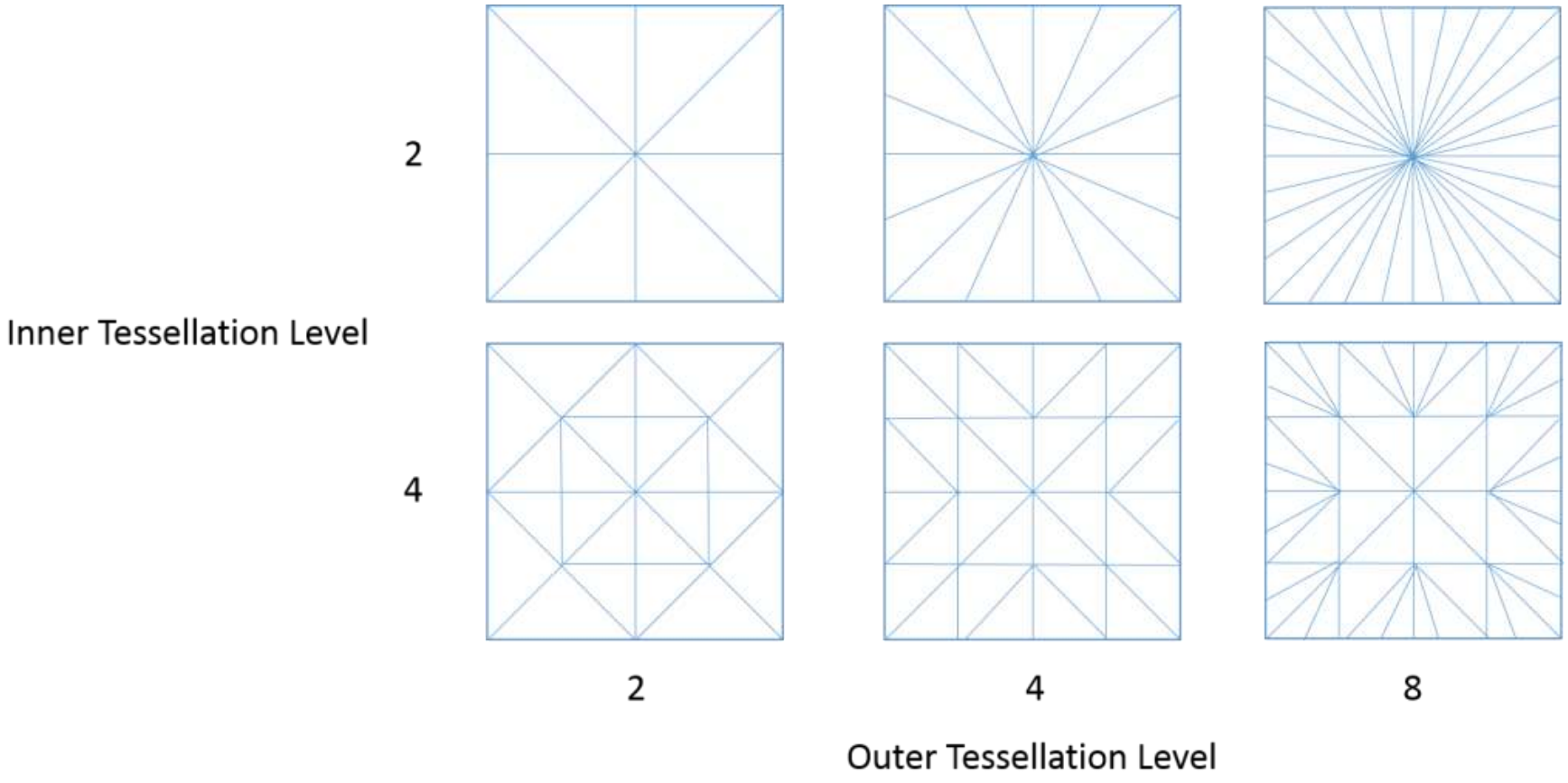
Tessellation

- Возможность динамически задавать разбиение примитивов
- Реализуется с помощью двух дополнительных этапов графического конвейера



Tessellation

- Control shader
  - In: Патчи (геометрия)
  - Out: факторы тесселяции
    - Inner
    - Outer
- Evaluation shader
  - In: барицентрические координаты вершины
  - Out: итоговая позиция вершины


Inner Tessellation Level
2
4
2
4
8
Outer Tessellation Level

Вопросы?