

Компьютерная графика

Лекция 3. Удаление невидимых граней. Алгоритм Z-буфера

к.ф.-м.н. Филонов Павел Владимирович
filonovpv@gmail.com

Московский Государственный Технический Университет
Гражданской Авиации

18 марта 2017 г.

Задача удаления невидимых граней

Ситуации, приводящие необходимости удаления невидимых граней:

- На сцене находится несколько объектов и один загораживает другой;
- Объект представляет собой невыпуклый многогранник;
- Объекты взаимно перекрывают друг друга;

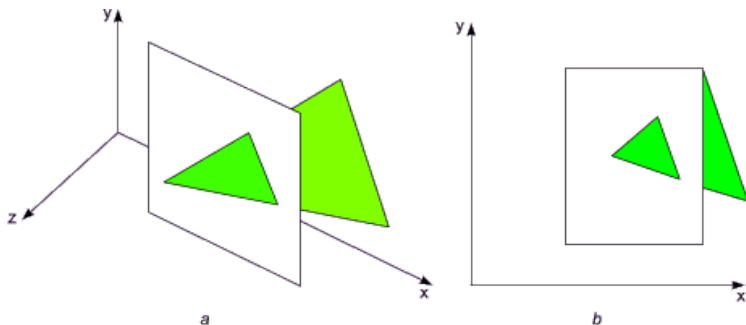
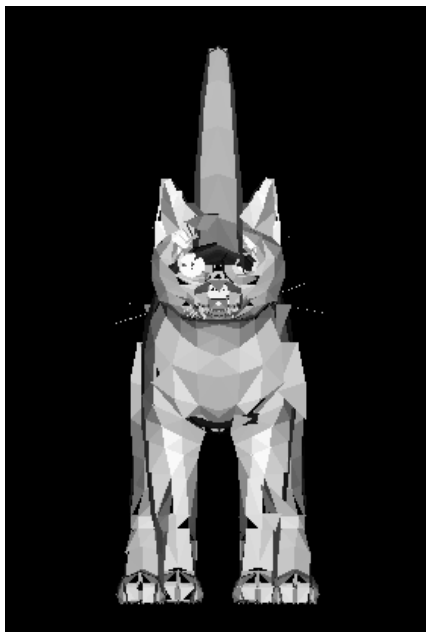


рис. 21.1

Ограничения алгоритма удаления нелицевых граней

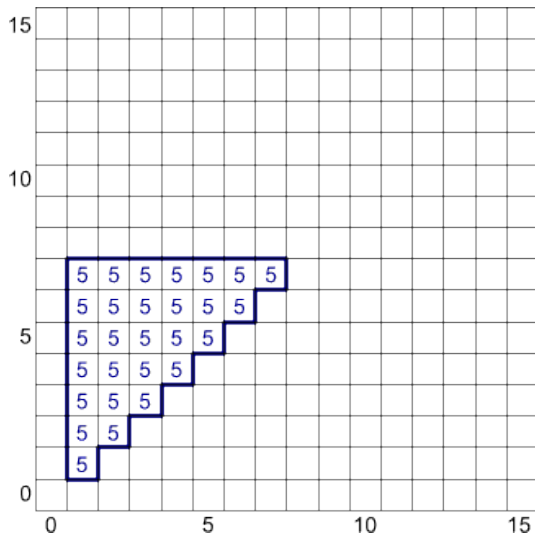


Алгоритмы уаления невидимых граней

- **Алгоритм Робертса** — вычисляет границы пересечения областей видимости;
- **Алгоритм художника** — объекты сортируются по Z координате и визуализируются последовательно; в случае необходимости разрезаются на части;
- **Двоичное разбиение пространства** — грани упорядочиваются по Z координате с помощью специальной структуры данных;
- **Алгоритм трассировки лучей** — цвет каждого пиксела рассчитывается как пересечение исходящего из него луча света с одним из объектов на сцене.
- **Алгоритм Z -буфера** — используется дополнительный массив глубины для каждого пикселя.

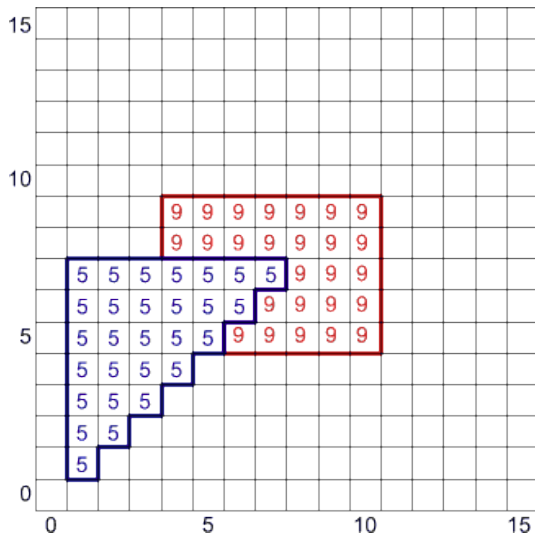
Алгоритм Z-буфера

Первый многоугольник с координатами (1, 1, 5), (7, 7, 5), (1, 7, 5)



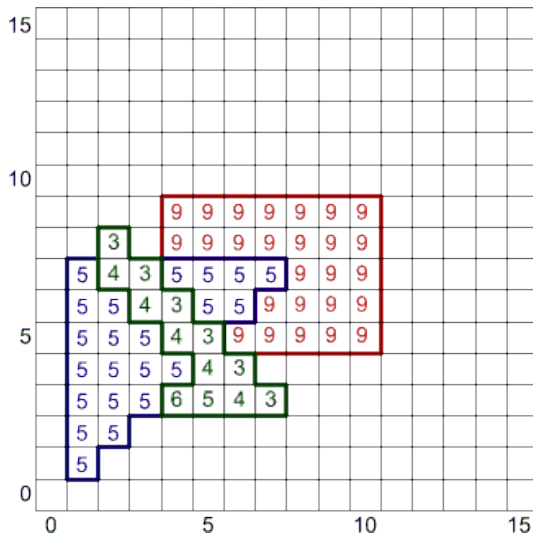
Алгоритм Z-буфера

Второй многоугольник с координатами (3, 5, 9), (10, 5, 9), (10, 9, 9), (3, 9, 9)



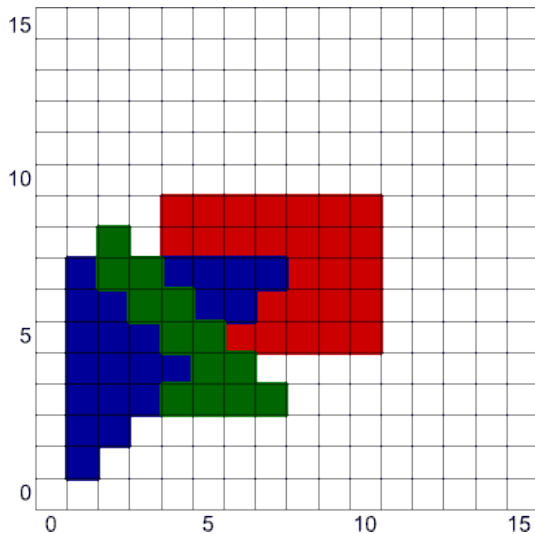
Алгоритм Z-буфера

Третий многоугольник с координатами (2, 6, 3), (2, 3, 8), (7, 3, 3)



Алгоритм Z-буфера

Результат работы алгоритма



Алгоритм Z-буфер

RENDER(m)

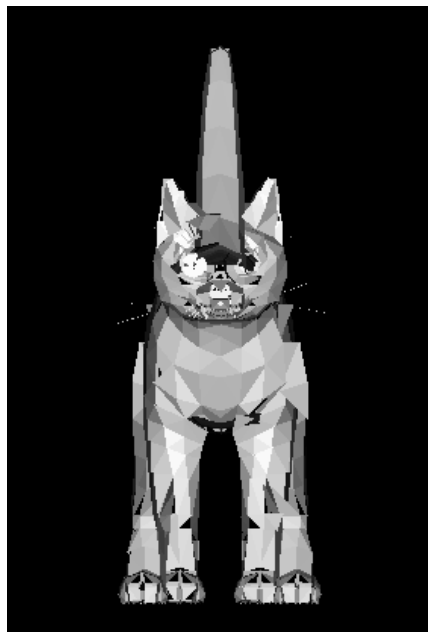
```
1   $zbuffer \leftarrow$  минимально возможные значения
2  for  $f \in$  полигоны  $m$ 
3      do  $\vec{n} \leftarrow$  нормаль к  $f$ 
4           $I = (\vec{n}, \vec{l}) \triangleright$  интенсивность
5          if  $I < 0$ 
6              then  $\triangleright$  грань видима
7                   $color = |I| \cdot (255, 255, 255) \triangleright$  цвет грани
8                   $\{(x_{ij}, y_{ij}, z_{ij}) | j = 0, 1, 2\} \leftarrow$  координаты  $f$ 
9                  TRIANGLE( $\{(x_{ij}, y_{ij}, z_{ij}) | j = 0, 1, 2\}, zbuffer$ )
```

Алгоритм Z-буфер

TRIANGLE($x_0, y_0, z_0, x_1, y_1, z_1, x_2, y_2, z_2, zbuffer$)

```
1  ▷ упорядочим вершины по координате  $y : y_0 \leq y_1 \leq y_2$ 
2  for  $y \leftarrow y_0$  to  $y_2$ 
3      do  $(x_a, z_a) \leftarrow$  координата пересечения
4          сканирующей строки с левой границей
5           $(x_b, z_b) \leftarrow$  координата пересечения
6          сканирующей строки с правой границей
7      for  $x \leftarrow x_a$  to  $x_b$ 
8          do  $z \leftarrow$  соответствующая точке  $(x, y)$ 
9              if  $z > zbuffer[x, y]$ 
10                  then  $zbuffer[x, y] \leftarrow z$ 
11                      PUTPIXEL( $x, y$ )
```

Результат работы алгоритма алгоритма Z-буфера



Результат работы алгоритма алгоритма Z-буфера

