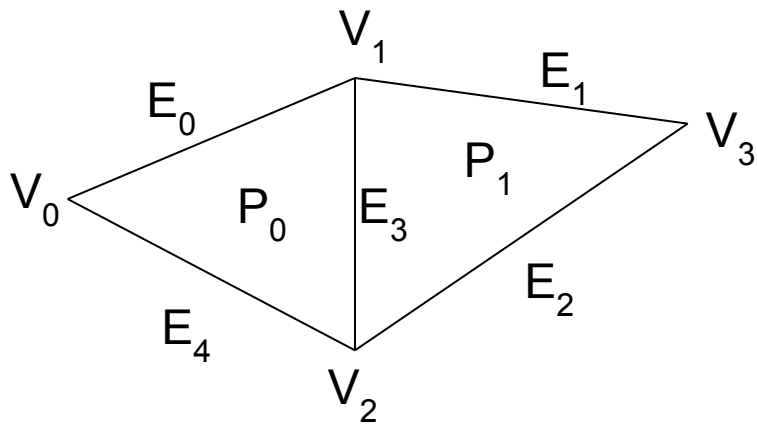


Визуализация в 3D

URL: <http://www.school30.spb.ru/cgsg/cgc/>

E-mail: CGSG@yandex.ru



$$V = \{V_0, V_1, V_2, V_3\} = \{(x_0, y_0, z_0), (x_1, y_1, z_1), \dots\}$$

$$E^* = \{\#V^*1, \#V^*2, \#P^*1, \#P^*2\}$$

$$E_0 = \{0, 1, 0, -1\}$$

$$E_1 = \{1, 3, 1, -1\}$$

$$E_2 = \{3, 2, 1, -1\}$$

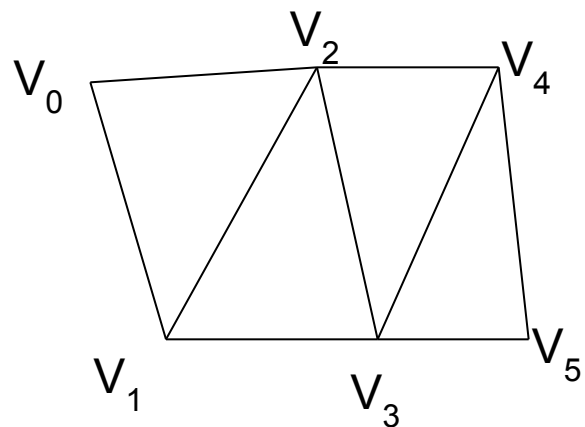
$$E_3 = \{1, 2, 0, 1\}$$

$$E_4 = \{2, 0, 0, -1\}$$

$$P^* = \{\#V^*1, \#V^*2, \#V^*3, \#E^*1, \#E^*2, \#E^*3\}$$

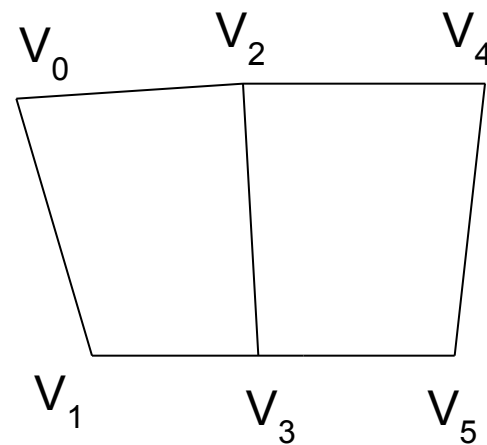
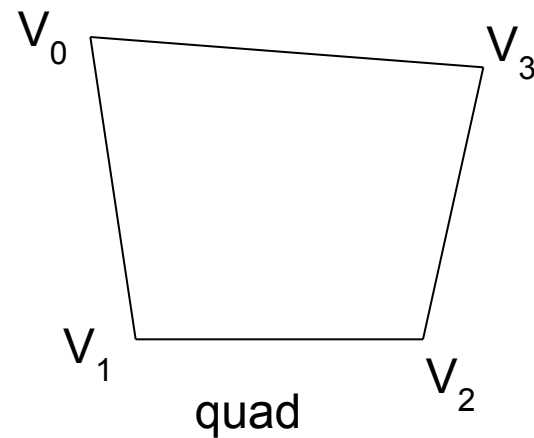
$$P_0 = \{0, 1, 2, 0, 3, 4\}$$

$$P_1 = \{1, 3, 2, 1, 2, 3\}$$

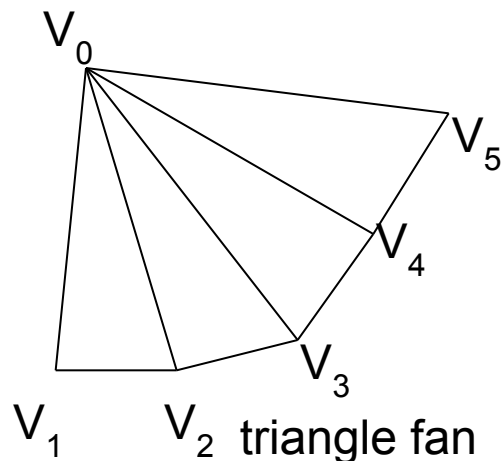


triangle strip

$(0, 1, 2), (1, 2, 3), (2, 3, 4), (3, 4, 5)$



quad strip



triangle fan

$(0, 1, 2), (0, 2, 3), (0, 3, 4), (0, 4, 5)$

Работа в «контексте»:

```

openStructure (Id1) ;
    setLocalTransformation (rotateX (...)) ;
    addPolygon (... ) ;
    setLocalTransformation (rotateX (...)) ;
    openStructure (Id2) ;
    
```

```

setLocalTransformation (rotateX (...)) ;
    addPolygon (... ) ;
    ...
closeStructure () ;
openStructure (Id2) ;
    
```

```

setLocalTransformation (rotateX (...)) ;
    addPolygon (... ) ;
    ...
closeStructure () ;
    ...
closeStructure () ;
    
```

функции:

```
openStructure (Id) ;
closeStructure () ;
```

```
setPolygonAttribute (... ) ;
addPolygon (... ) ;
```

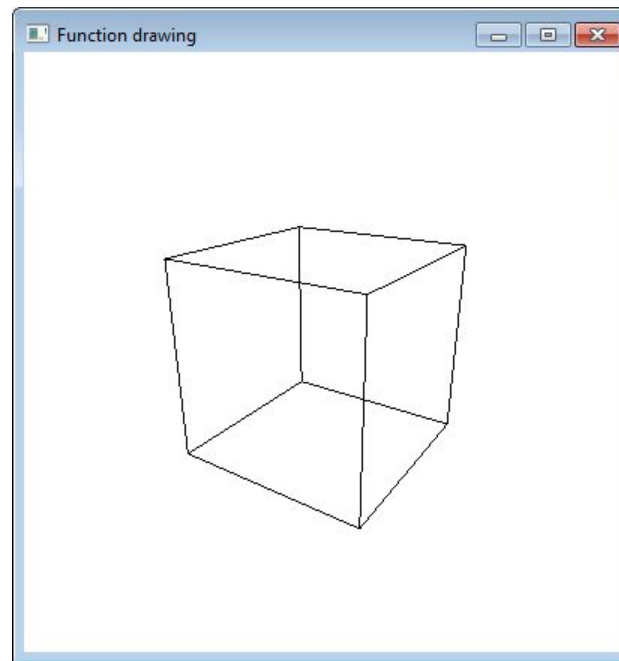
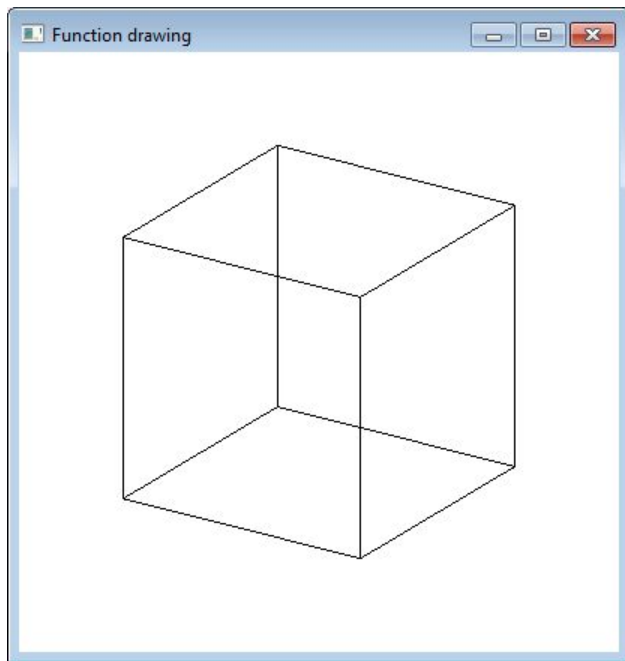
```
setLocalTransformation (matrix_4x4)
```

```
;
```

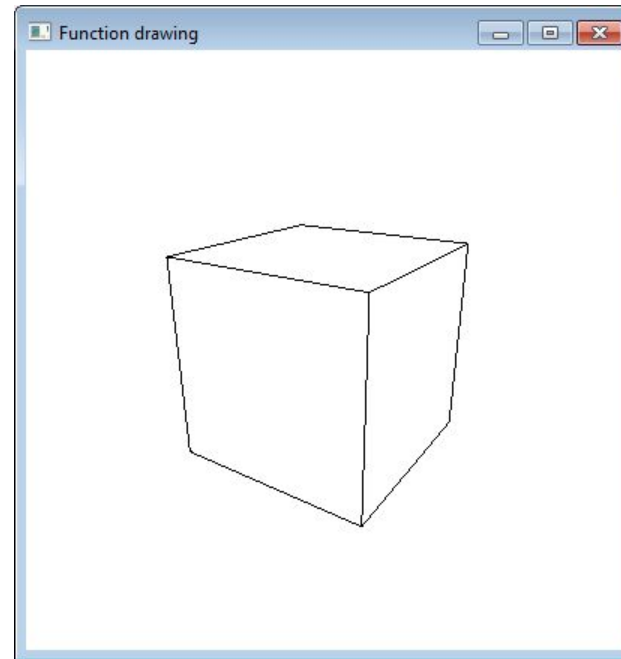
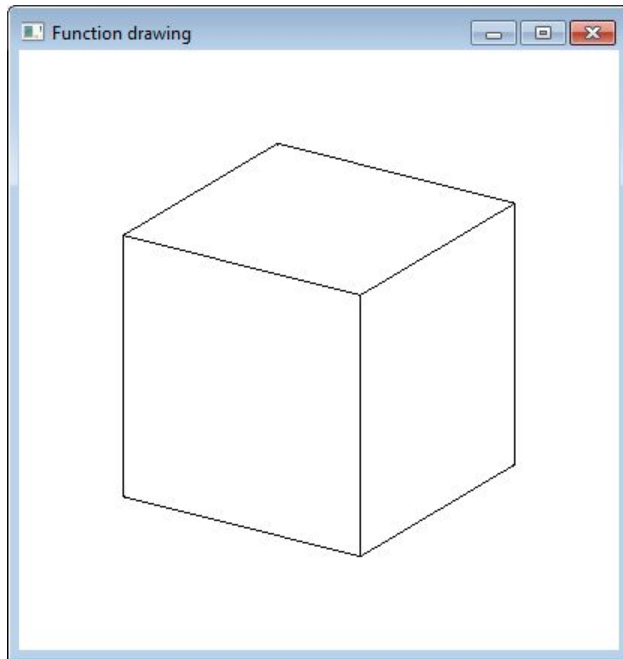
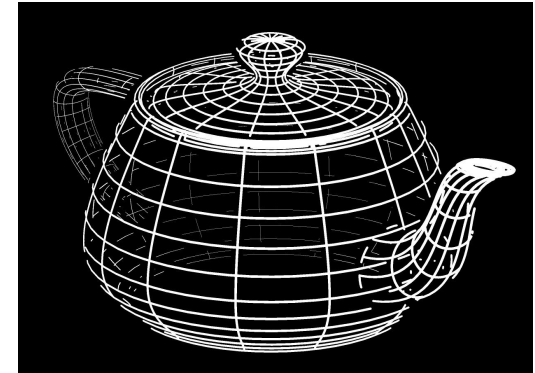
```
matrix_4x4 rotateX (angle) ;
matrix_4x4 rotateY (angle) ;
matrix_4x4 rotateZ (angle) ;
matrix_4x4 scale (sx, sy, sz) ;
matrix_4x4 translate (dx, dy, dz) ;
```

```
postRoot () ;
```

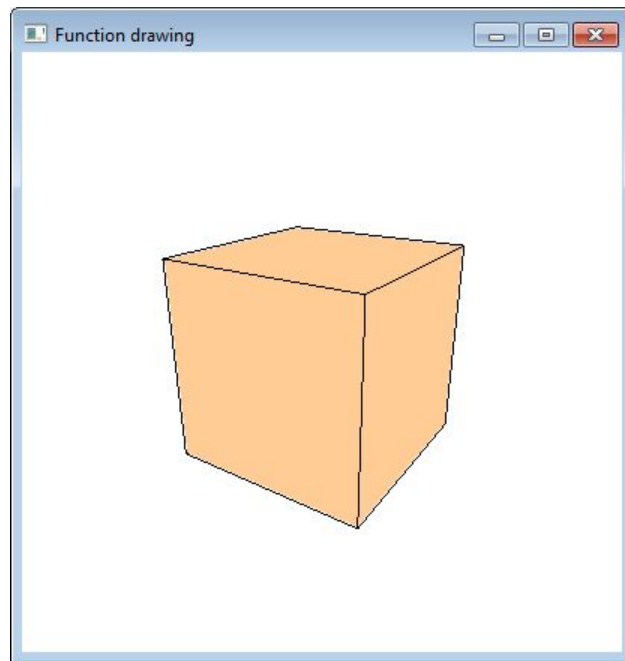
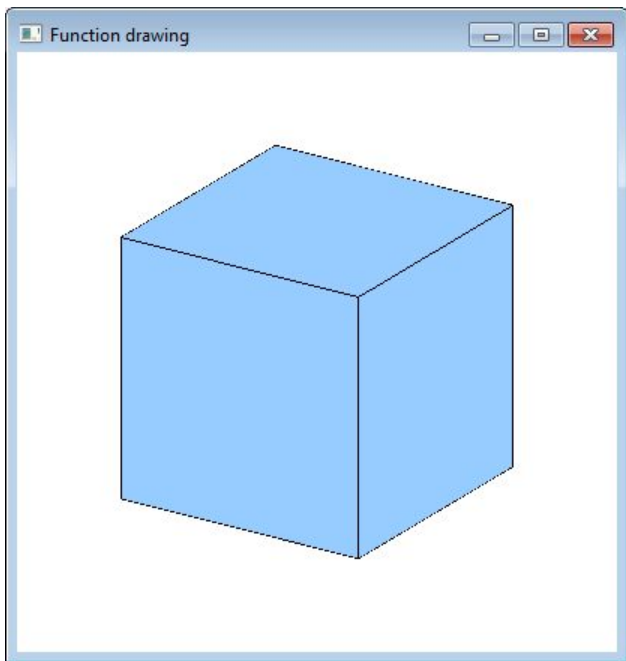
- Визуальный реализм
 - перспектива:



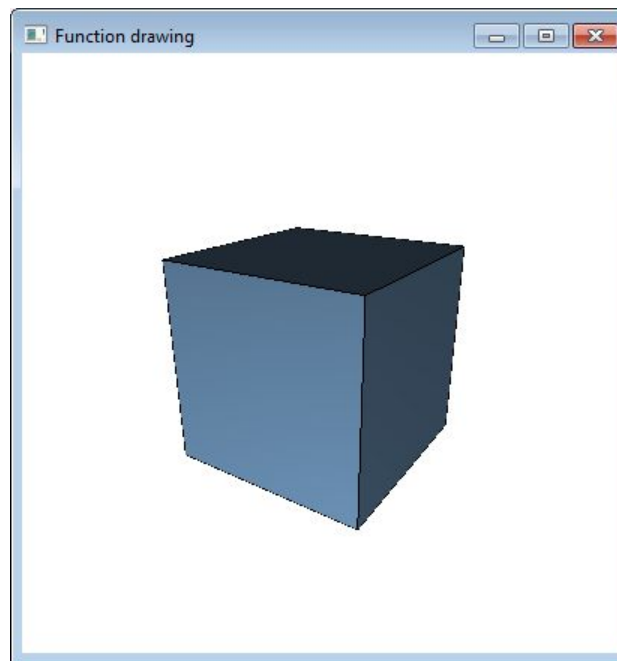
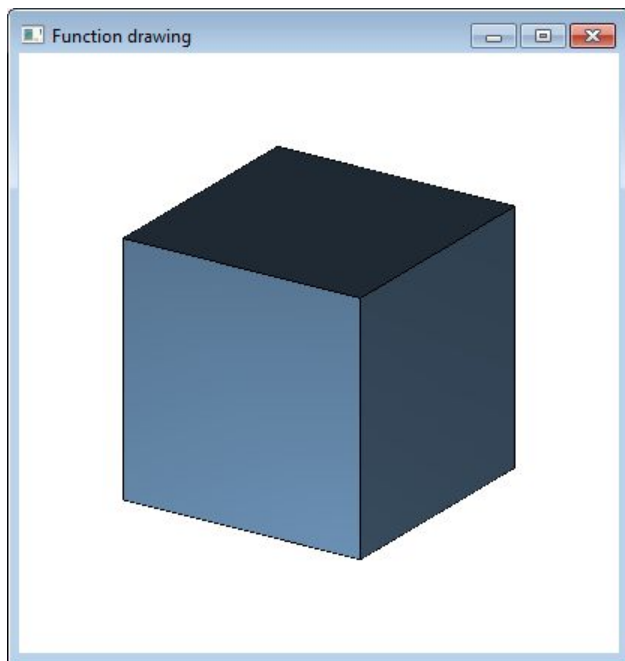
- Визуальный реализм
 - depth cueing (изображение глубины):
 - удаление невидимых линий



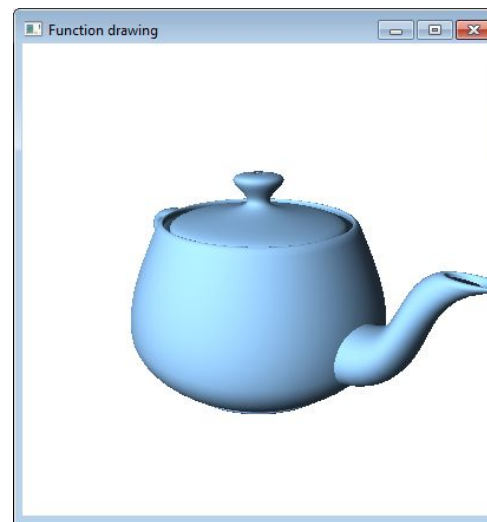
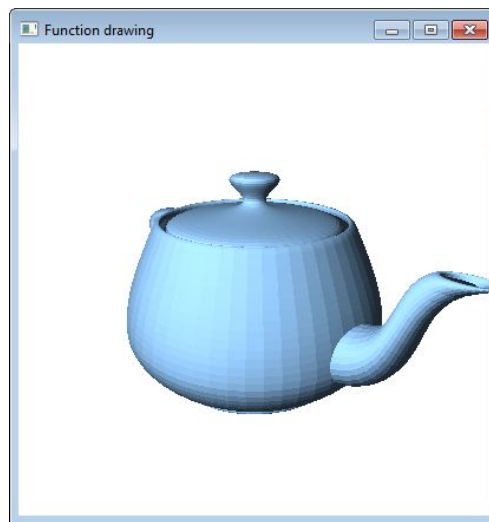
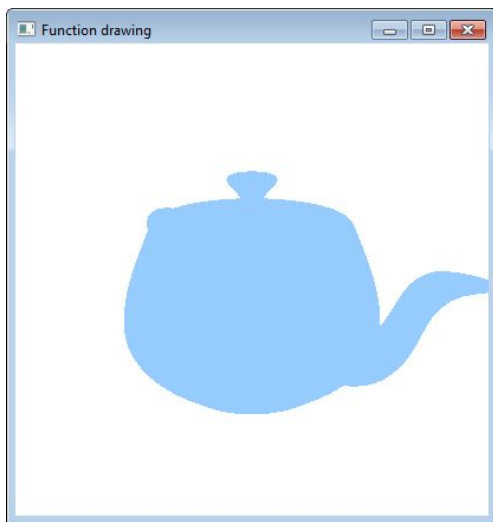
- Визуальный реализм
 - удаление невидимых поверхностей
 - цвет



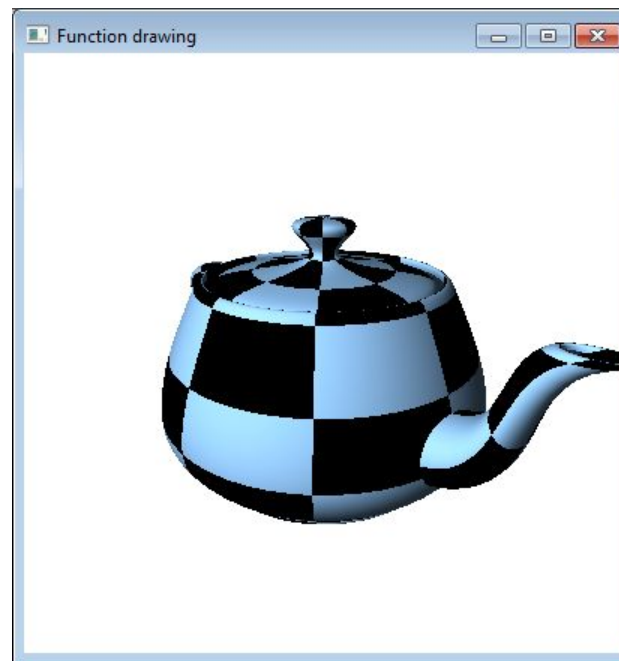
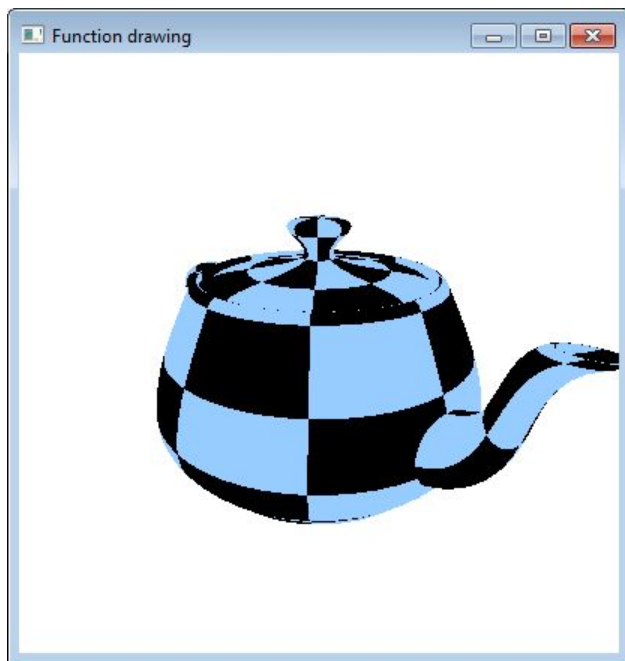
- Визуальный реализм
 - освещение

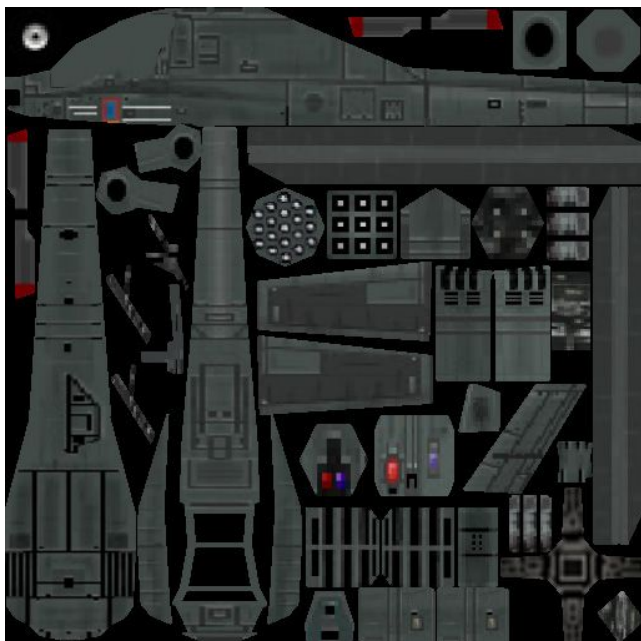
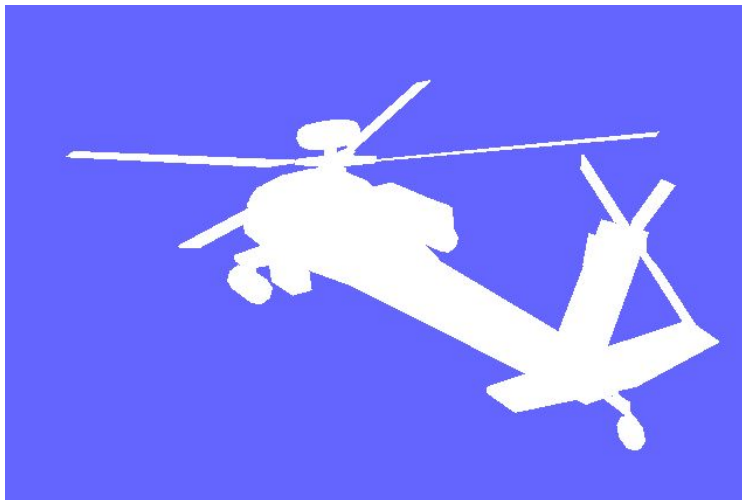


- Визуальный реализм
 - закразка и интерполяция

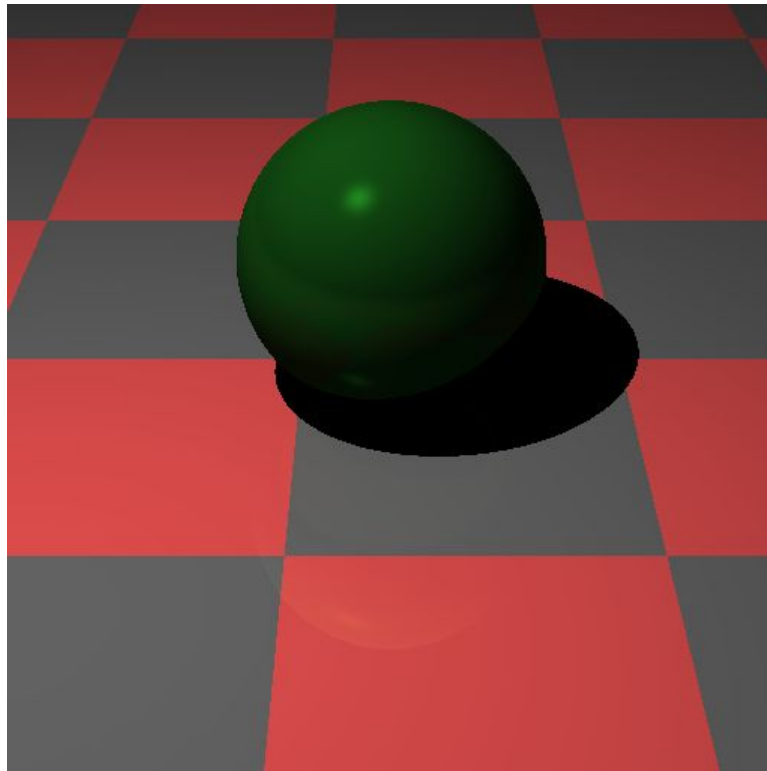


- Визуальный реализм
 - текстурирование

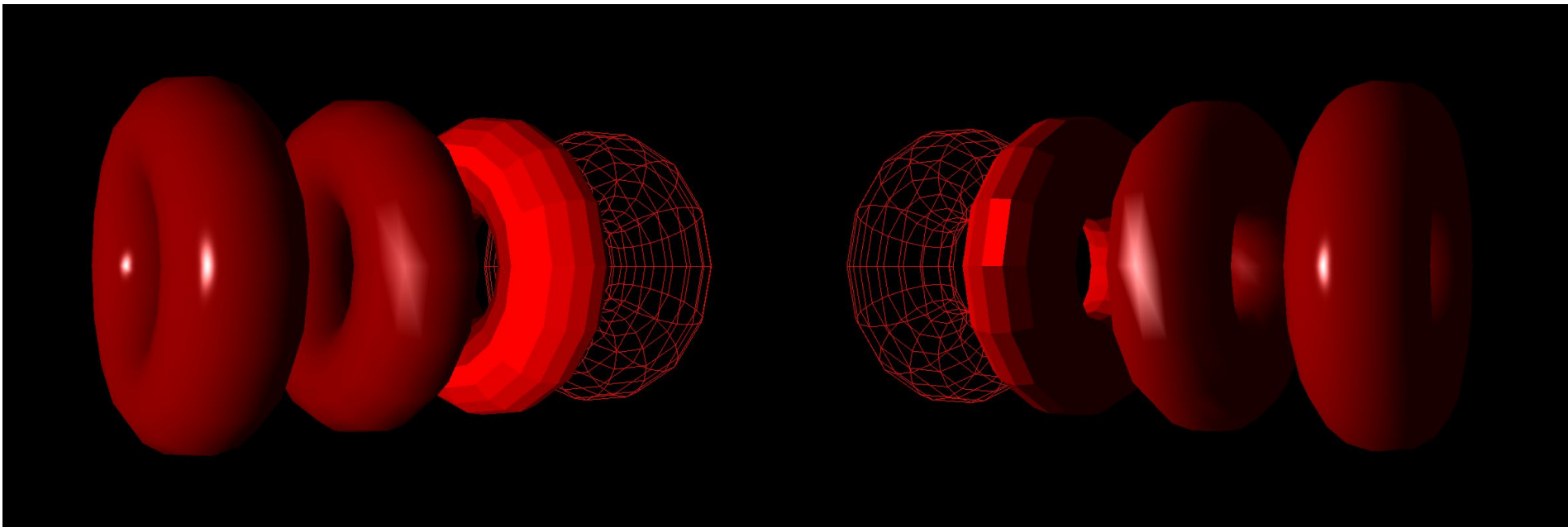
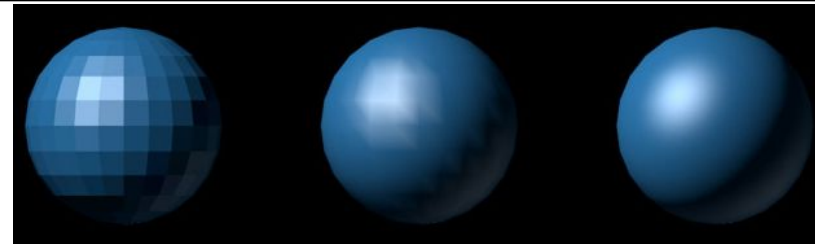


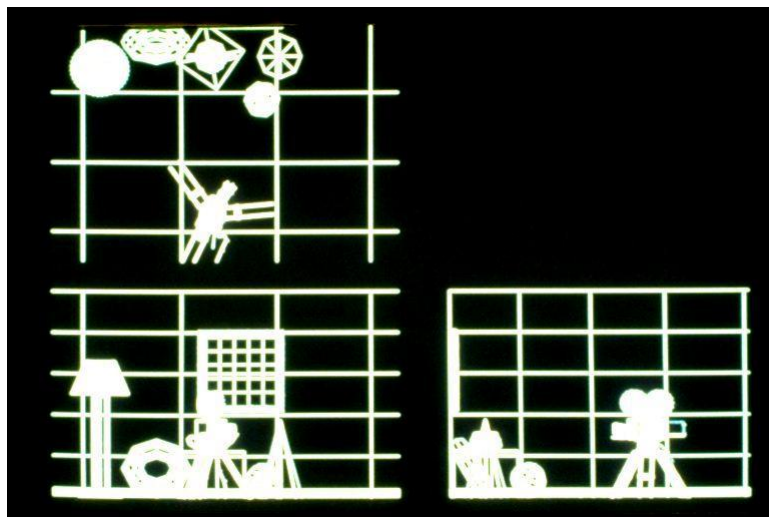


- Визуальный реализм
 - тени



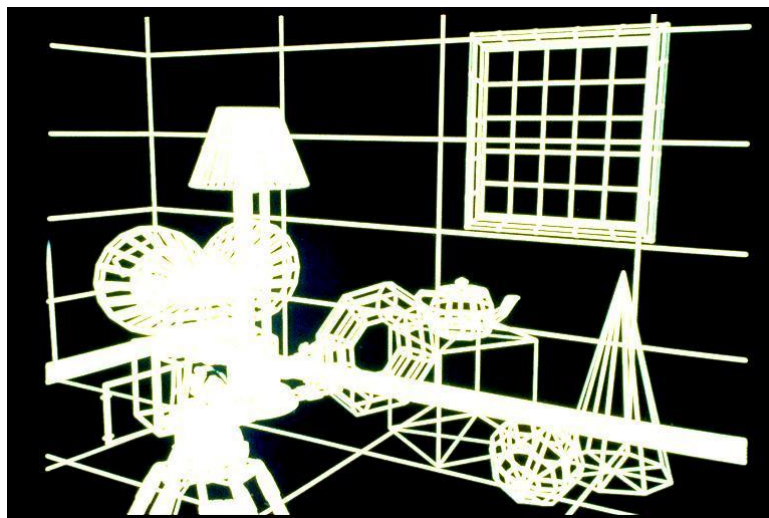
- каркасная визуализация (wireframe)
- плоская закрапка (flat shading)
- интерполяция цвета – закрапка Гуро (Gouraud shading)
- интерполяция нормали – закрапка Фонга (Phong shading)



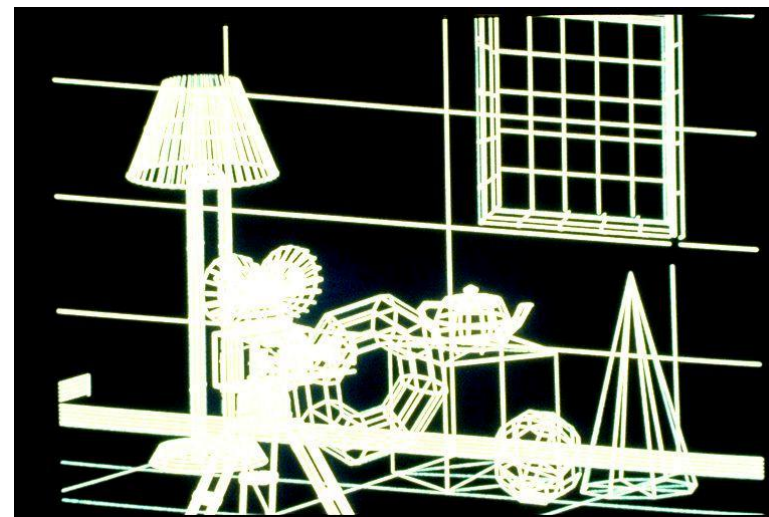


3 Orthographic views

Perspective View (no hidden lines)

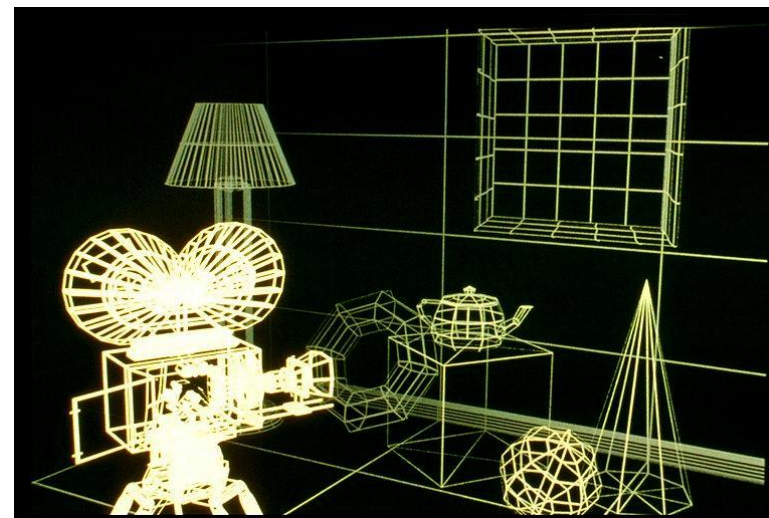


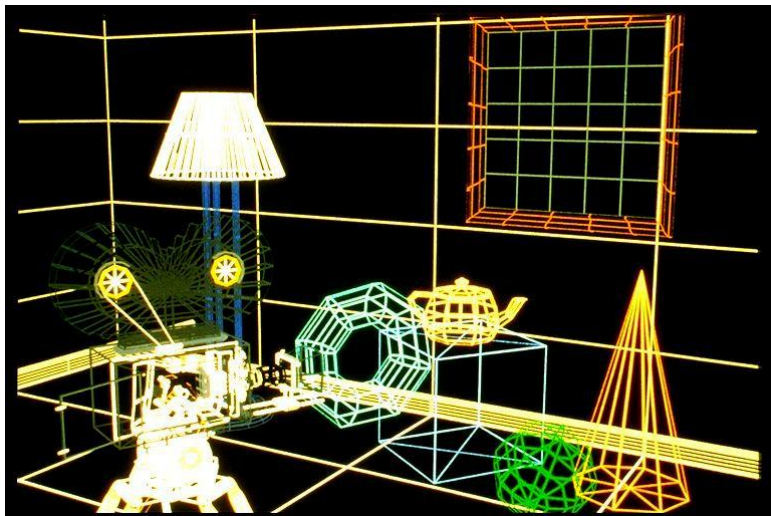
<http://www.siggraph.org/education/materials/HyperGraph/shutbug.ht>



Parallel projection

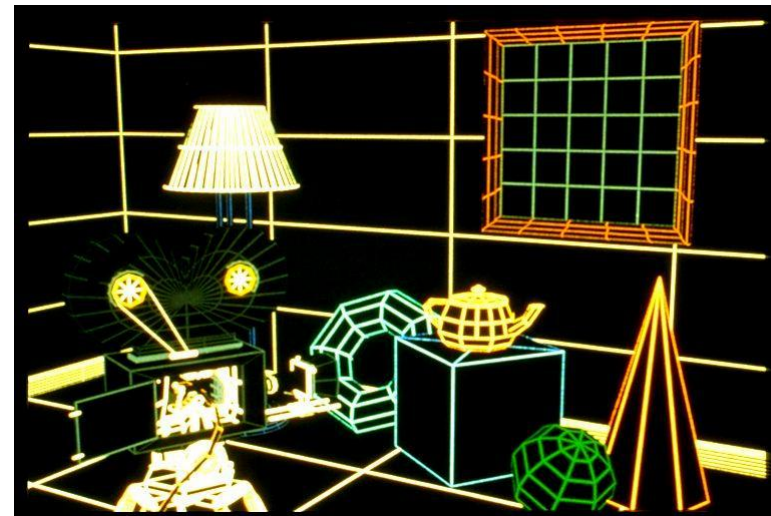
Depth cuing (hidden lines)





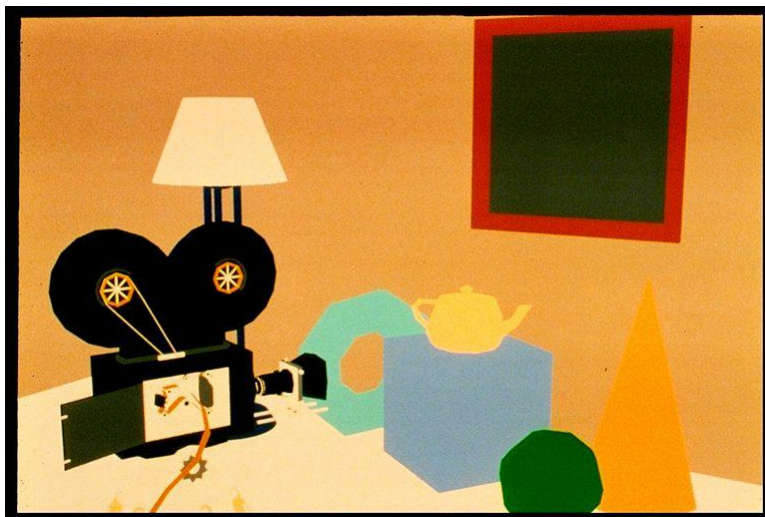
Colored

Visible surface determination (constant shading)

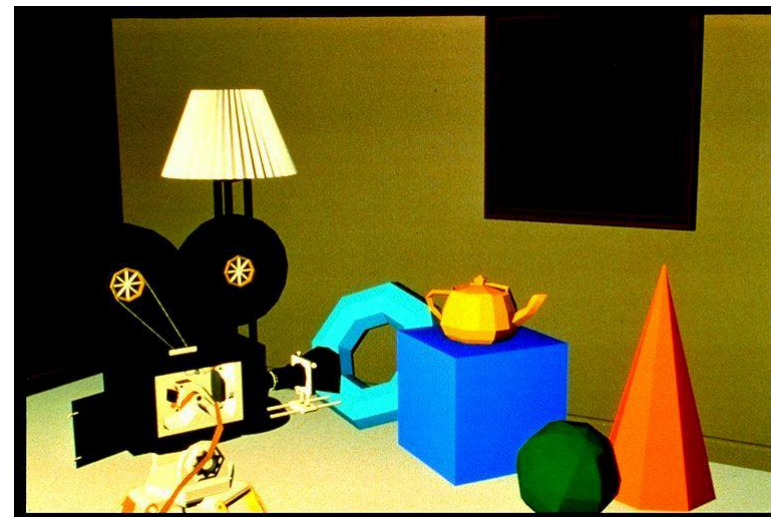


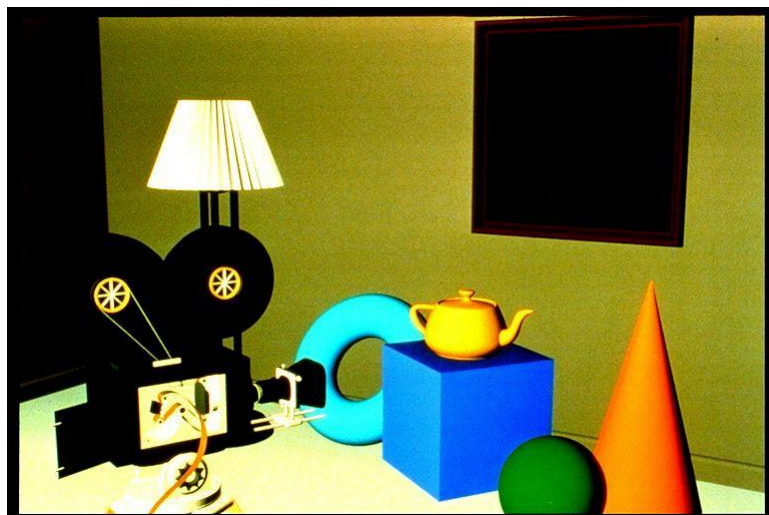
Visible line determination (with facets)

Shaded by facets



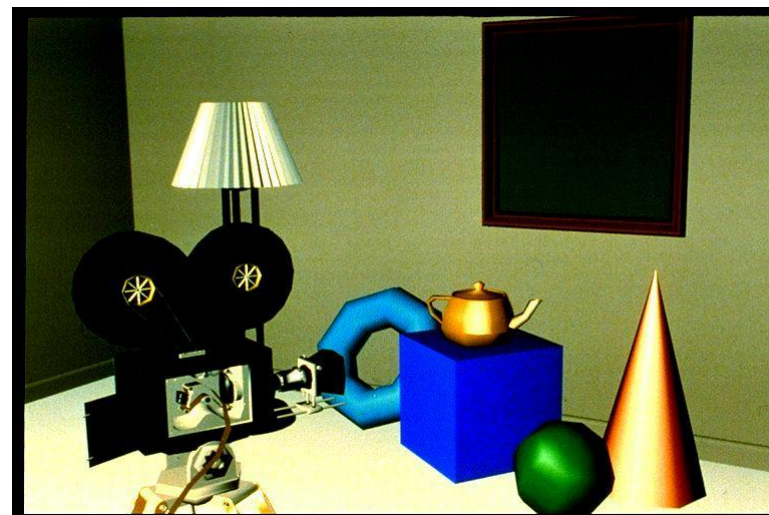
<http://www.siggraph.org/education/materials/HyperGraph/shutbug.ht>





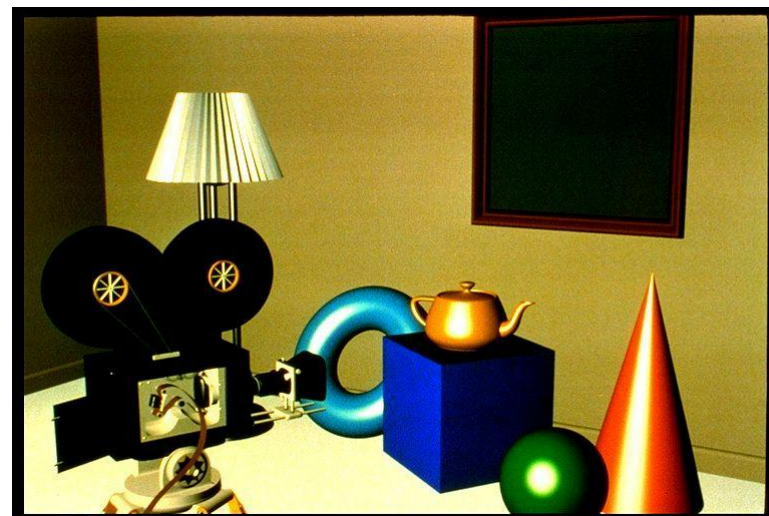
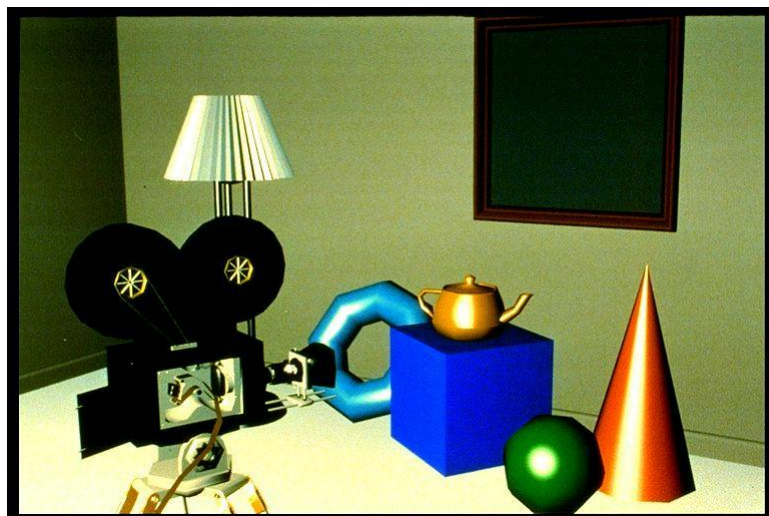
Gouraud shaded (diffuse)

Phong shaded (polygon based)



Gouraud shaded (specular)

Phong shaded (curved surfaces based)



<http://www.siggraph.org/education/materials/HyperGraph/shutbug.ht>



Multiple lights

Bump mapping and shadow

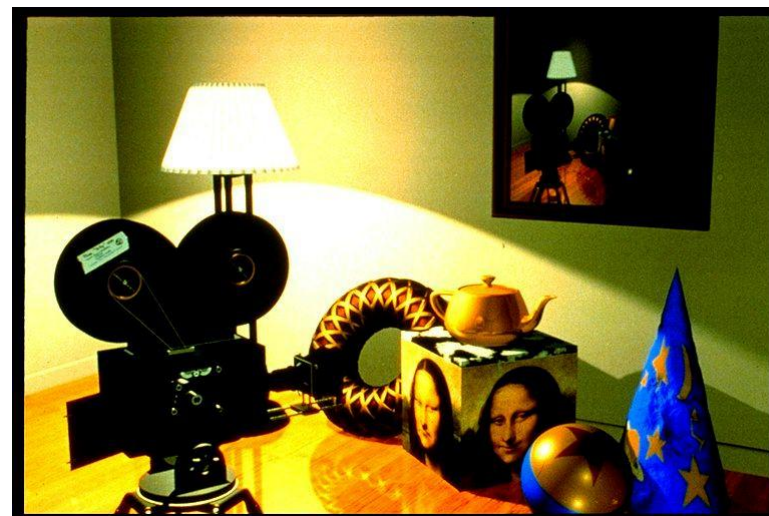


Texture mapping

Reflection mapping



<http://www.siggraph.org/education/materials/HyperGraph/shutbug.ht>



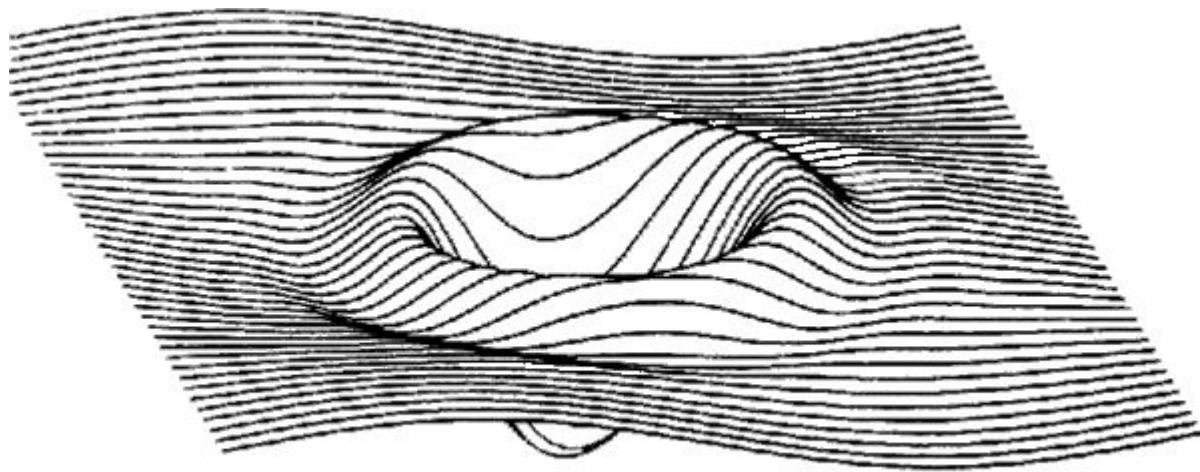
глобальные массивы:

Up[W] и Down[W]

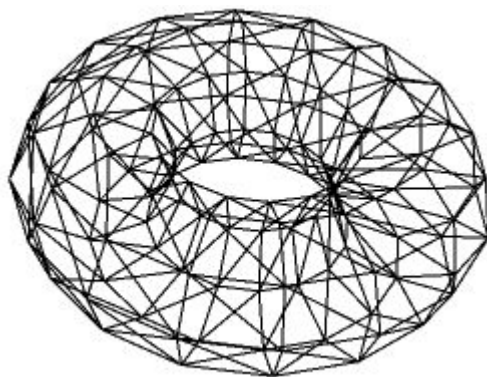
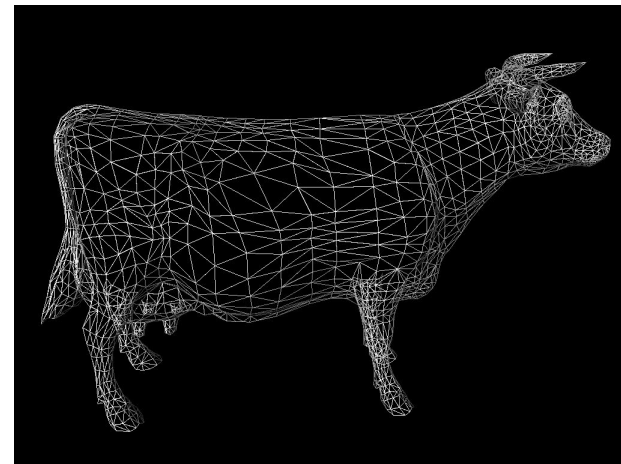
Функция точки(x,y):

```

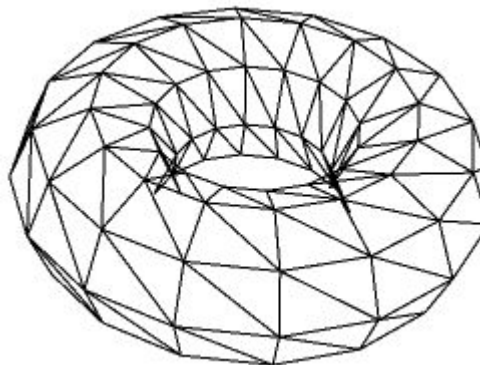
if (y > Up[x])
{
    SetPixel(x,y);
    Up[x] = y;
}
if (y < Down[x])
{
    SetPixel(x,y);
    Down[x] = y;
}
    
```



- отсечение «задних» поверхностей
- определение передних и задних многоугольников по обходу (по и против часовой стрелке – CW, CCW)

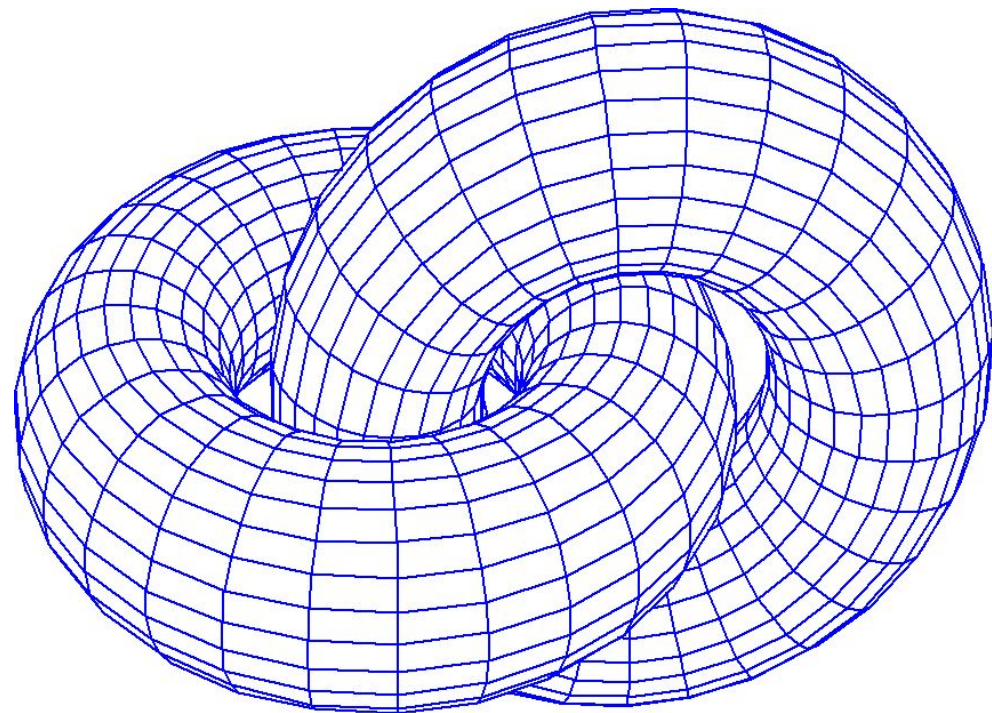


без отсечения



с отсечением

1. Все тела выпуклы (или приводим к такому виду)
2. Удаление самоперекрывающихся ребер и граней
3. Каждое из видимых ребер каждого тела сравнивается с каждым из оставшихся тел для определения перекрытия



для каждого мн-ка

для каждой точки мн-ка

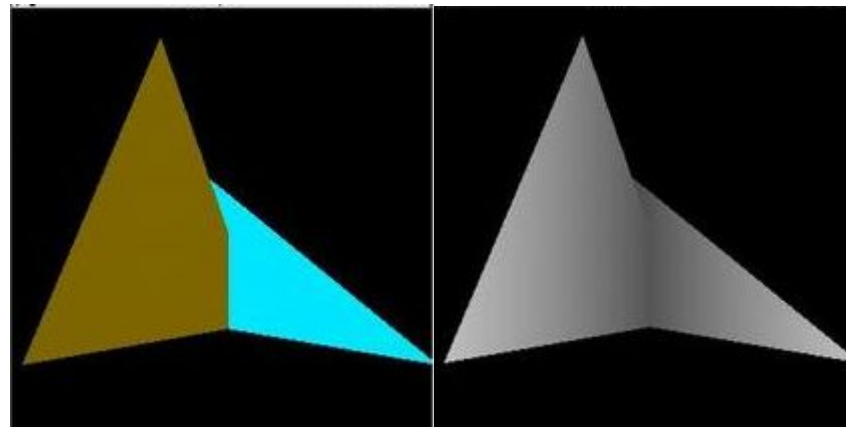
pz = значение z в (x, y)

если $pz \leq ZBuf[y][x]$

тогда

`SetPixel(x, y)`

`ZBuf[y][x] = pz`



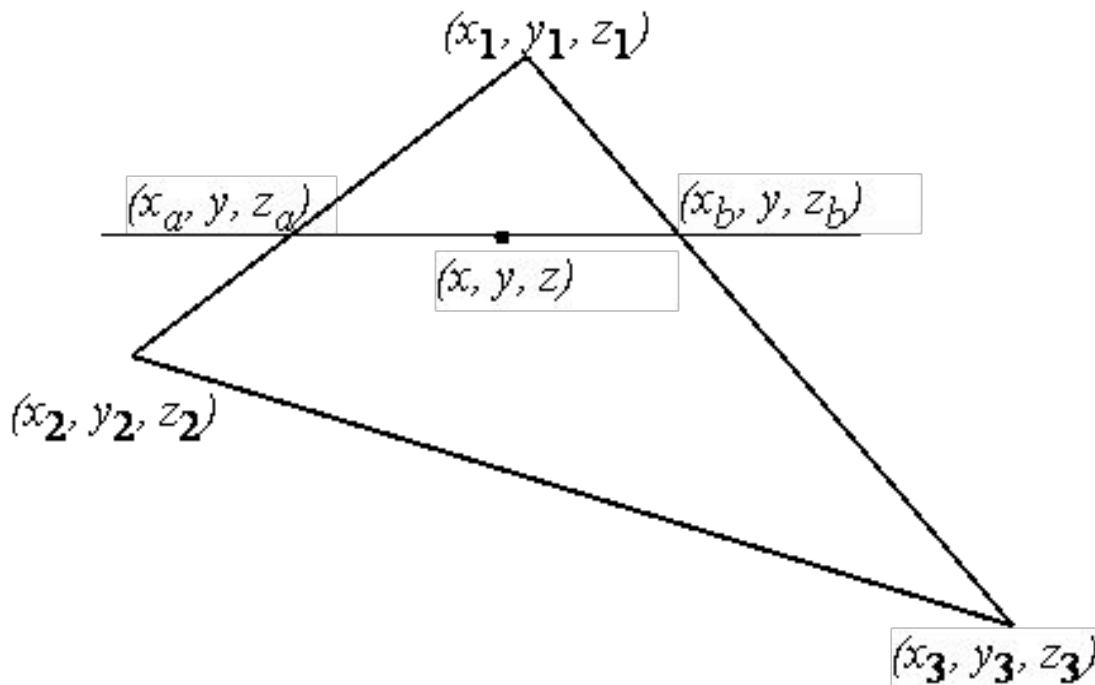
$$x_a = x_1 + (x_2 - x_1) \cdot \frac{y - y_1}{y_2 - y_1};$$

$$x_b = x_1 + (x_3 - x_1) \cdot \frac{y - y_1}{y_3 - y_1};$$

$$z_a = z_1 + (z_2 - z_1) \cdot \frac{y - y_1}{y_2 - y_1};$$

$$z_b = z_1 + (z_3 - z_1) \cdot \frac{y - y_1}{y_3 - y_1};$$

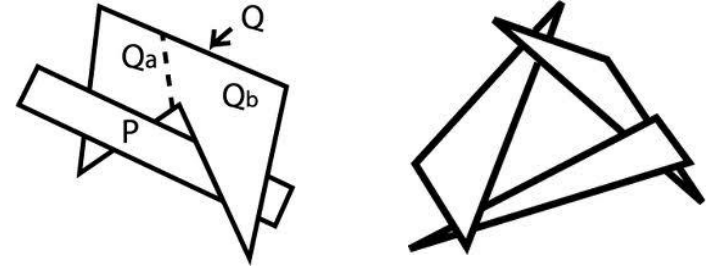
$$z = z_a + (z_b - z_a) \cdot \frac{x - x_a}{x_b - x_a}$$



Сортировка многоугольников по оси Z (по глубине) – ближней или дальней точке

Разрешение всех неоднозначностей при перекрытиях по глубине, «разрезание» при необходимости многоугольников

Построение всех многоугольников от дальнего по глубине к ближнему.



текущий (дальний) мн-к P:
по каждому мн-ку Q,
с которым P пересекается по z:

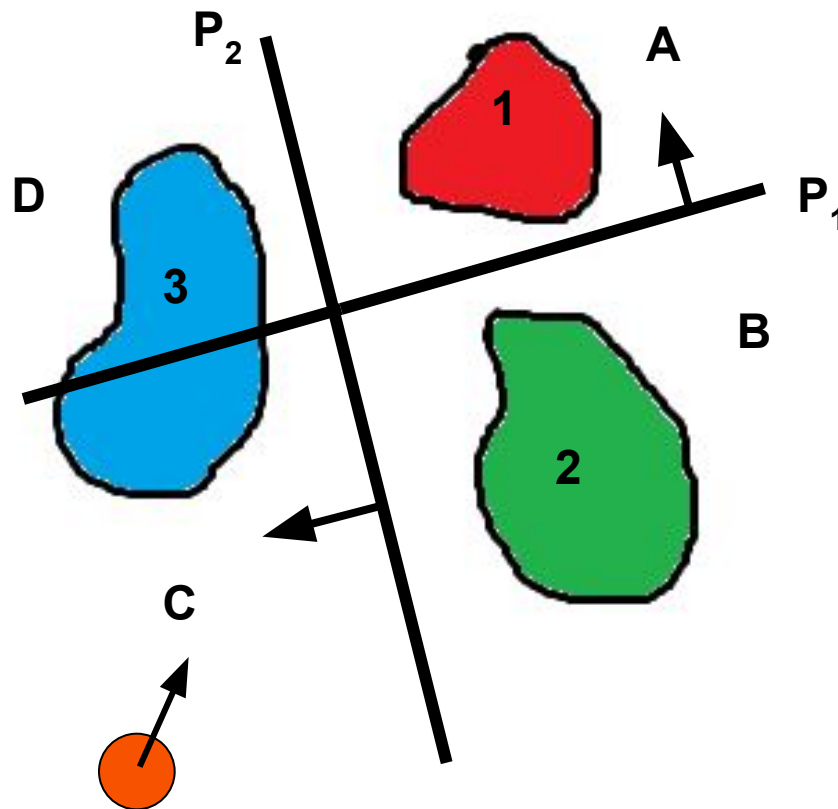
Оболочки P и Q не пересекаются по x?

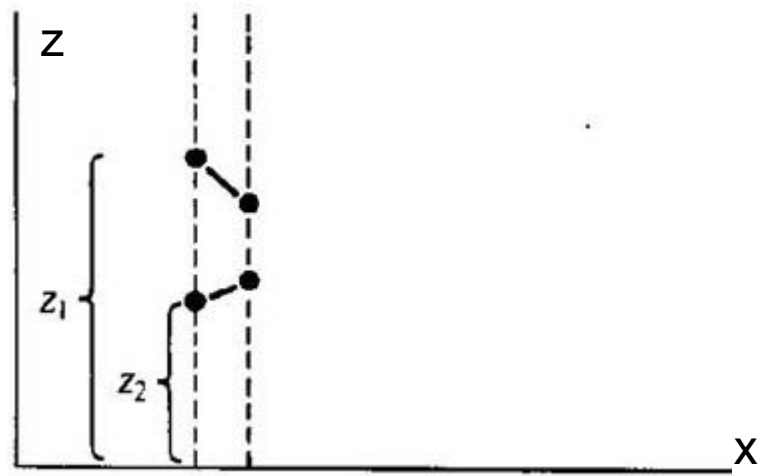
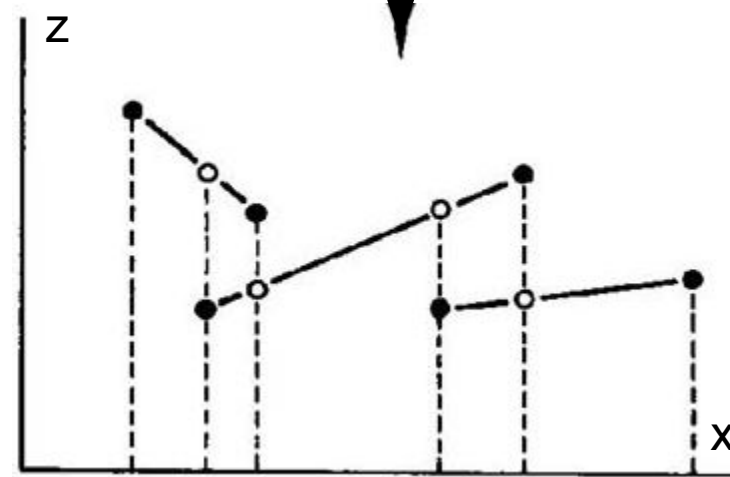
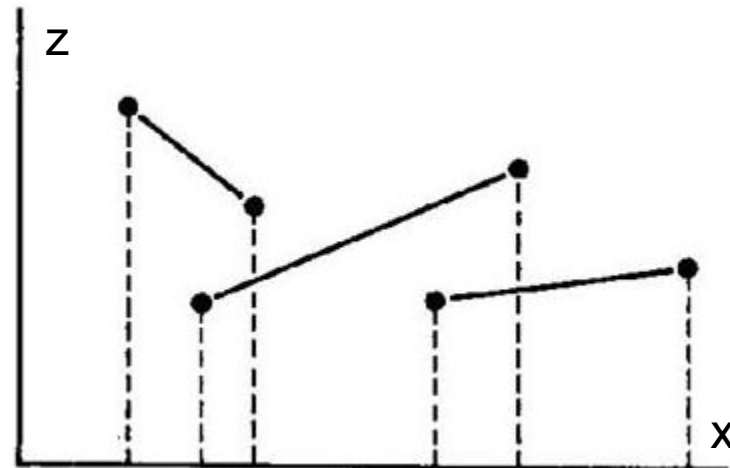
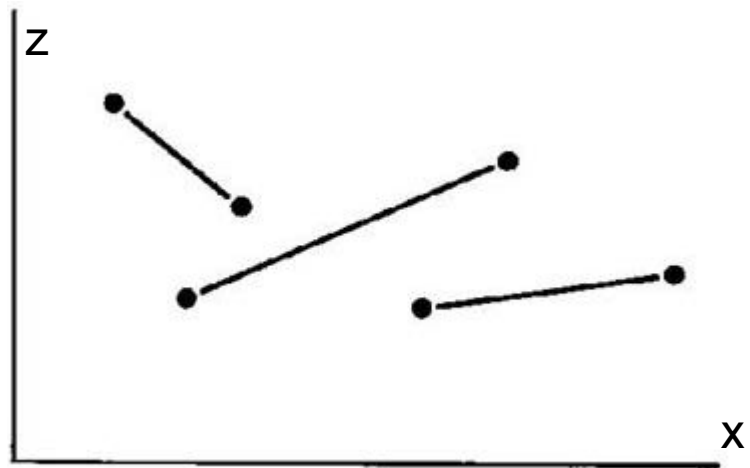
Оболочки P и Q не пересекаются по y?

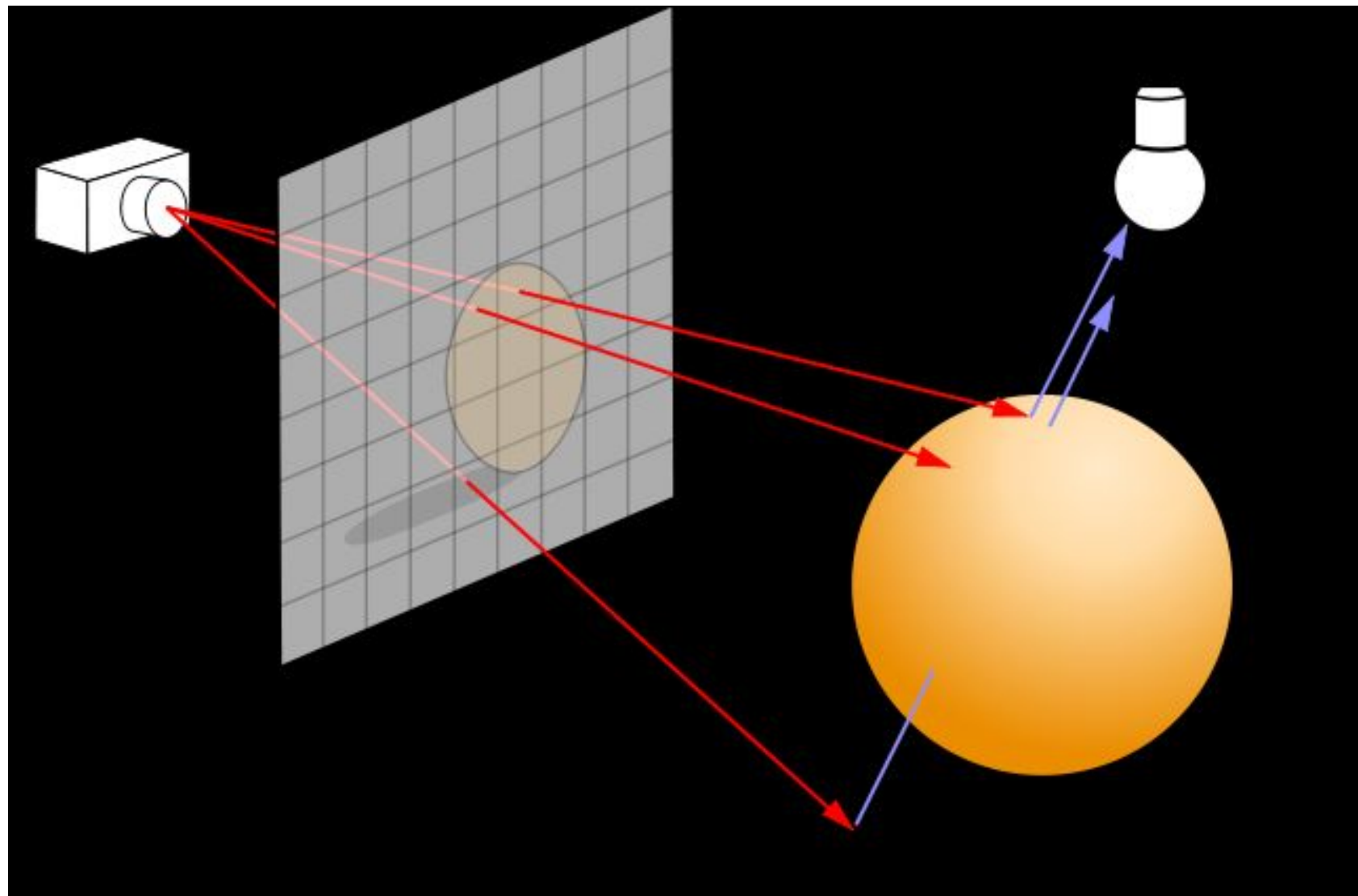
P целиком лежит по другую сторону от Q по отношению к наблюдателю?

Q целиком лежит по одну сторону от P по отношению к наблюдателю?

Проекции P и Q на плоскость (x,y) не пересекаются?







- **Практические задания (до 20.11.2011)**
 - Реализовать две программы построения простой 3D сцены в реальном времени (с использованием собственных функций построения в буфер кадра):
 - «проволочная» анимация - на выбор:
 - или график функции $y=f(x,z)$ с использованием алгоритма плавающего горизонта
 - или 2-3 выпуклых тела визуализируются с использованием алгоритма Робертса
 - визуализация сплошных тел – 2-3 выпуклых тела визуализируются с использованием алгоритма z-буфера с закраской Гуро, освещение вычисляется как косинус угла между нормалью к поверхности (к точке) и направлением на источник света.