Компьютерная графика. Часть 1.

Лектор – доц. Беляев Сергей Юрьевич.

Оглавление

Компьютерная графика. Часть 1. 1

Лектор – доц. Беляев Сергей Юрьевич. 1

Литература 1

1. Graphics pipeline 1
2. Geometry 5
3. Transforms 7
4. Lighting 12
5. Textures 15
6. Blending 18
7. Fog and antialiasing 20
8. Clipping 21

Литература

1. Tomas Akenine-Moller, Eric Haines. Real-Time Rendering, Third Edition. 2008.
2. DirectX документация.

1. Graphics pipeline

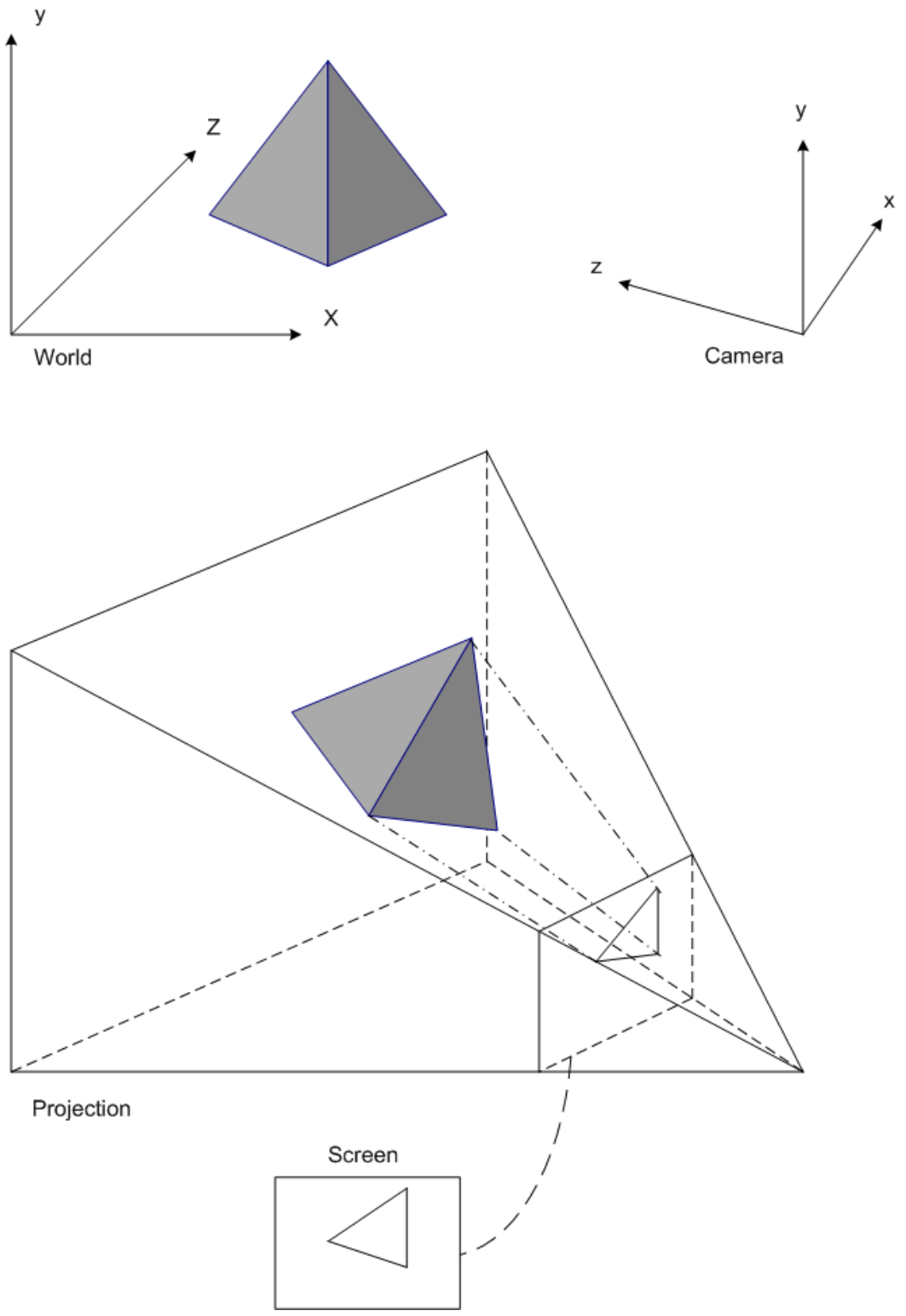


Рис.1.1

View frustum – near plane, far plane, угол обзора
View plane, view window, viewport
Центральное и параллельное проецирование

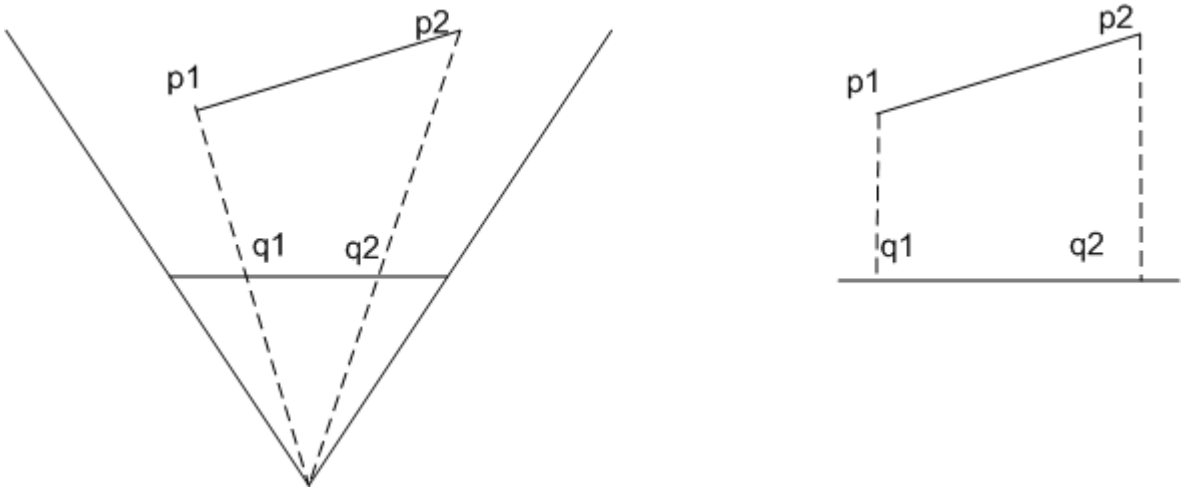


Рис. 1.2

Back face and hided face

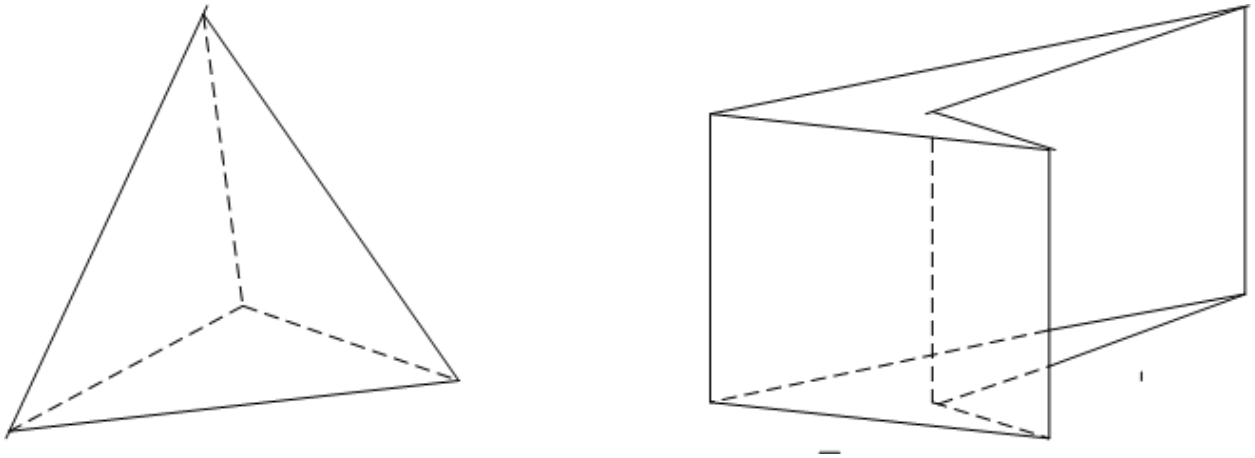


Рис. 1.3

Сортировка (топологическая), Z-buffer

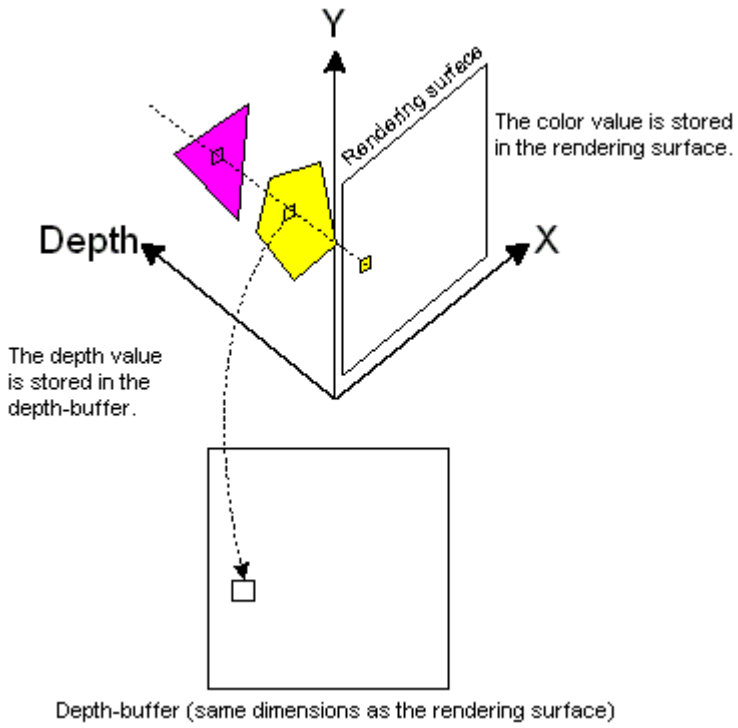


Рис. 1.4

Clipping and culling

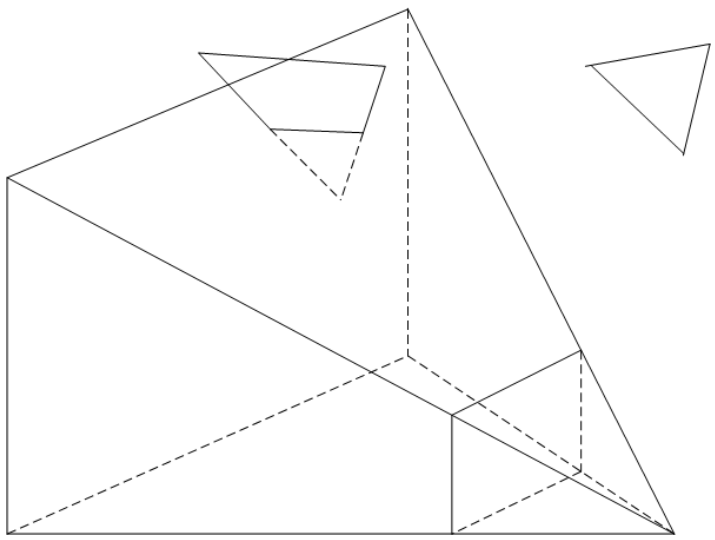


Рис. 1.5

Graphics pipeline

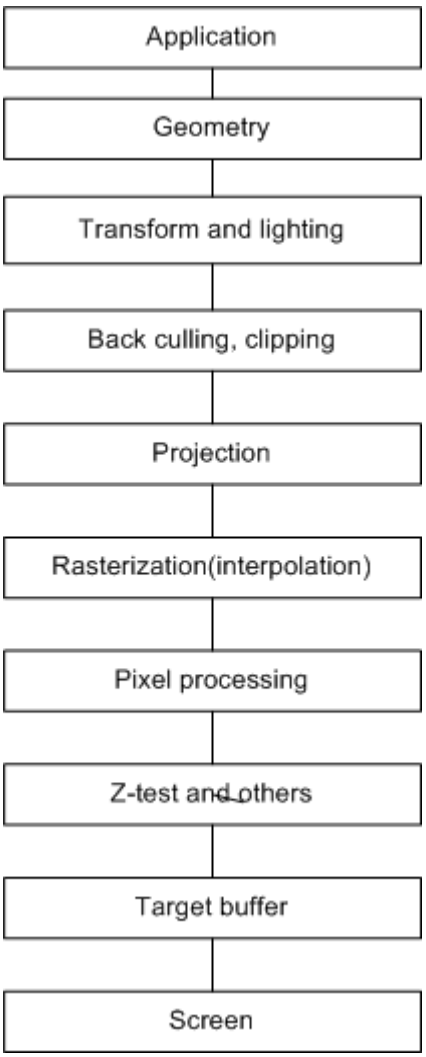


Рис. 1.6

2. Geometry

a. Графические примитивы

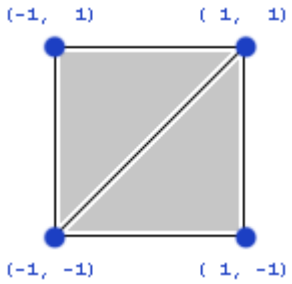


Рис. 2.1

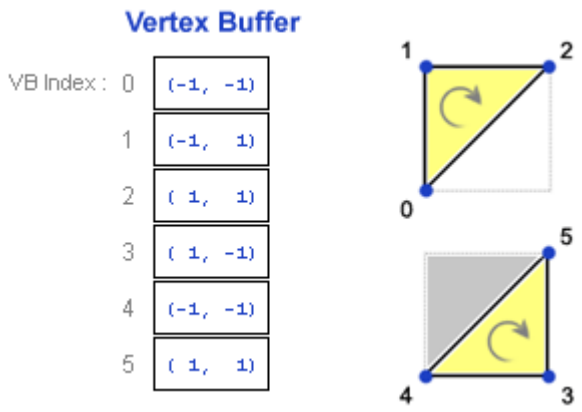


Рис. 2.2 Triangle List

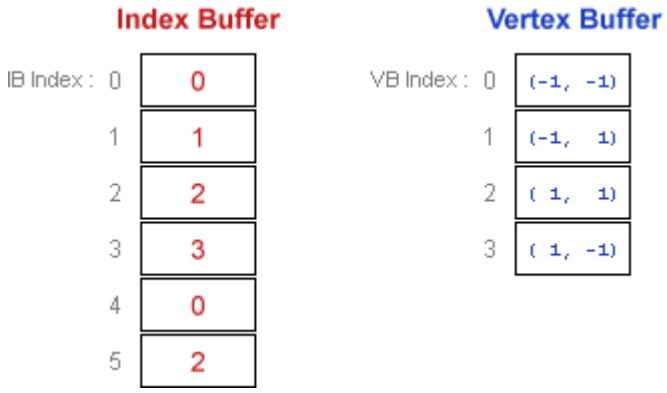


Рис. 2.3 Indexed Triangle List

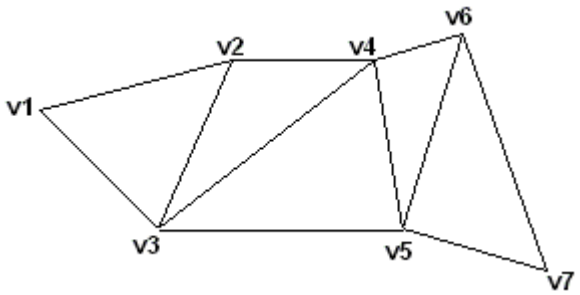


Рис. 2.4 Strip

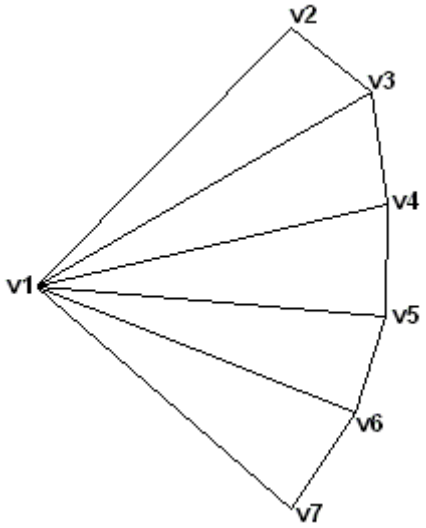


Рис. 2.5 Fan

b. Vertex format

Пример:
Vertex:
- Position
- Normal
- Color (diffuse)
- Color (specular)
- Texture coordinate (n)

c. Materials

3. Transforms

d. Matrix Transforms

Однородная (homogeneous) система координат

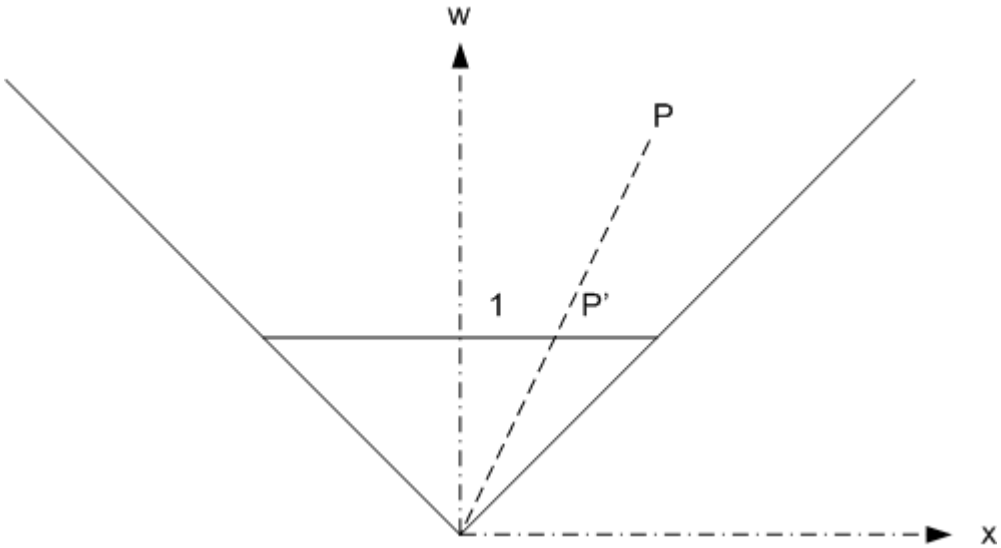


Рис. 3.1

p=[x, y, z, w]
p'=[x/w, y/w, z/w, 1]

Translate (T)

$$[x'y'z'1]=[x\ y\ z\ 1]\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ T_x & T_y & T_z & 1\end{bmatrix}$$

Scale (S)

$$[x'y'z'1]=[x\ y\ z\ 1]\begin{bmatrix}s_x & 0 & 0 & 0\\ 0 & s_y & 0 & 0\\ 0 & 0 & s_z & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Rotate (R_x)**:**

$$[x'y'z'1]=[x\ y\ z\ 1]\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & \cos\theta & \sin\theta & 0\\ 0 & -\sin\theta & \cos\theta & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$R=R_x*R_y*R_z$

W=R*S*T

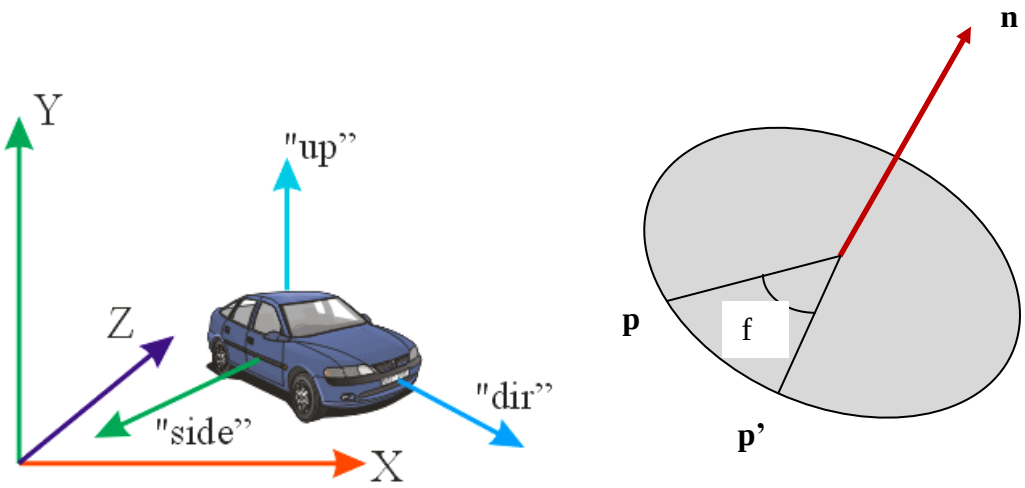


Рис. 3.2

(f, **n**)<=>R
p'=**p**R

Кватернион

q=(x, y, z, w)=(**n***sin(f/2), cos(f/2))

$$\begin{bmatrix} 1-2\cdot(y^2+z^2) & 2\cdot(x\cdot y-w\cdot z) & 2\cdot(x\cdot z+w\cdot y) \\ 2\cdot(x\cdot y-w\cdot z) & 1-2\cdot(x^2+z^2) & 2\cdot(y\cdot z-w\cdot x) \\ 2\cdot(x\cdot z-w\cdot y) & 2\cdot(y\cdot z+w\cdot x) & 1-2\cdot(x^2+y^2) \end{bmatrix}$$

$$\mathrm{slerp}\left(q_1, q_2, u\right) = q_1 \frac{\sin\left(1-u\right)\Omega}{\sin\Omega} + q_2 \frac{\sin\Omega u}{\sin\Omega}$$

$$q_1 \cdot q_2 = \cos\Omega$$

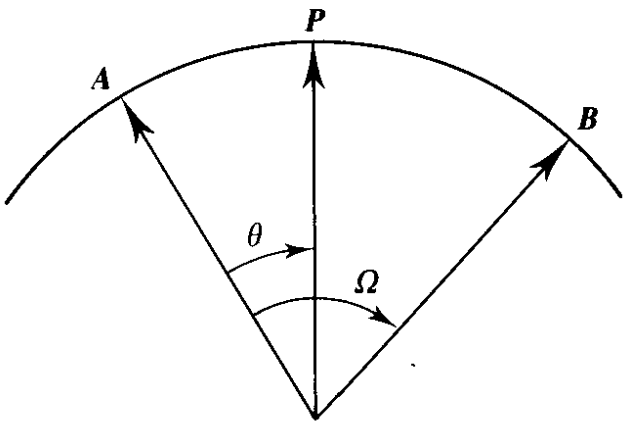


Рис. 3.3

q1*(1-u) + q2*u

e. World Transform

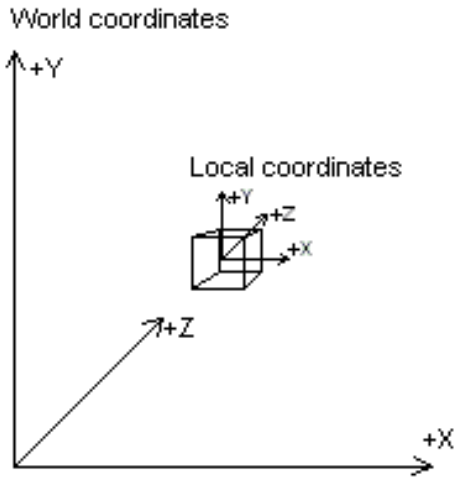


Рис. 3.4

P_w=P_l*W

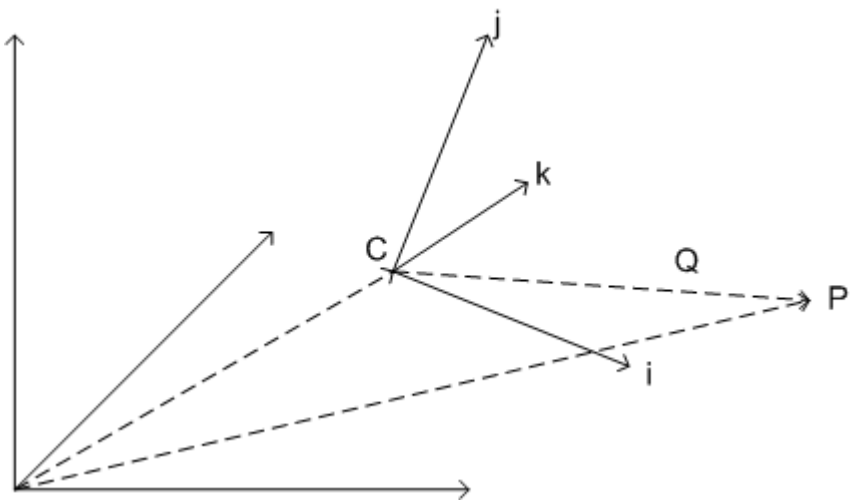


Рис. 3.5

P=C + Q
P=C + Q_xi + Q_yj + Q_zk

$$W = \begin{bmatrix} i_x, i_y, i_z, 0 \\ j_x, j_y, j_z, 0 \\ k_x, k_y, k_z, 0 \\ c_x, c_y, c_z, 1 \end{bmatrix}$$

P=Q*W

f. View Transform

V= W^{-1}

$$W = \begin{bmatrix} R, 0 \\ C, 1 \end{bmatrix}$$

$$W^{-1} = \begin{bmatrix} R^{-1}, 0 \\ -CR^{-1}, 1 \end{bmatrix}$$

R^{-1}= R'

g. Projection Transform.

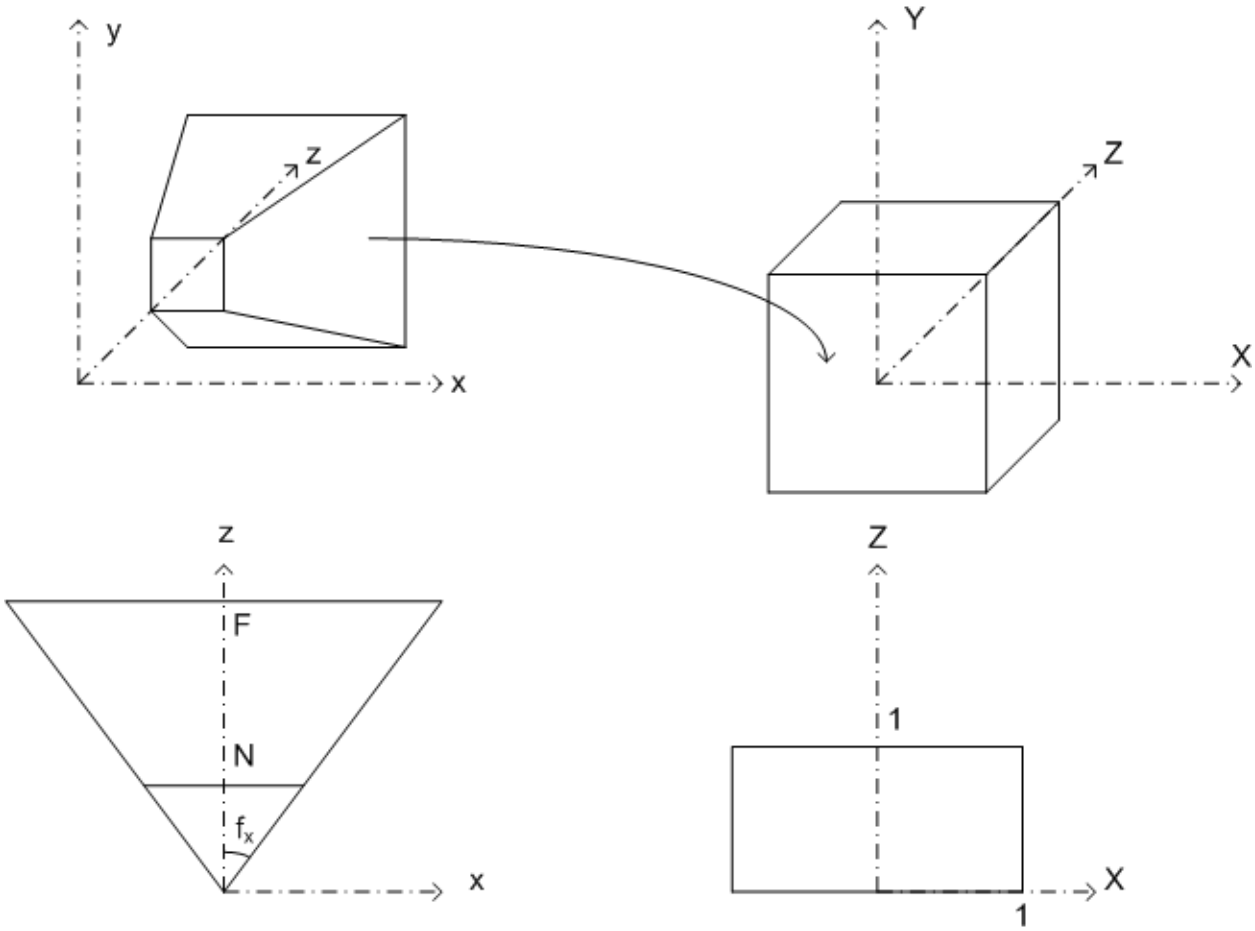


Рис. 3.6

$$P = \begin{bmatrix} W,0,0,0 \\ 0,H,0,0 \\ 0,0,a,1 \\ 0,0,b,0 \end{bmatrix}$$

(x, y, z, w) =(x, y, z, 1)*P
x=Wx; y=Hy; z=az+b; w=z;

X=W*x/z; Y=H*y/z; Z=a + b/z

1=W*tan(f_x); 1=H*tan(f_y)

0=a + b/N
1=a + b/F

W=1/tan(f_x); H= 1/tan(f_y)

a=-b/N;
1=-b(1/N-1/F) => b=-N*F/(F-N)
a=F/(F-N)

h. Viewport Transform

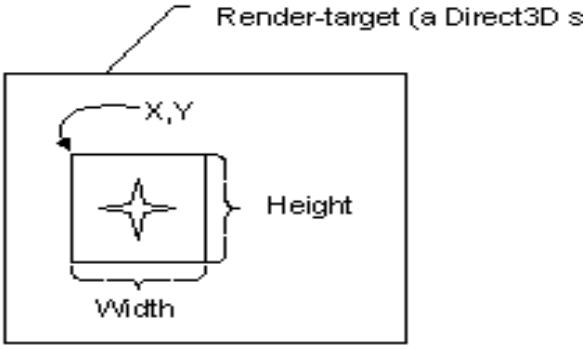


Рис. 3.7

$$VP = \begin{bmatrix} W/2 & 0 & 0 & 0 \\ 0 & -H/2 & 0 & 0 \\ 0 & 0 & Z\max & 0 \\ X+W/2 & Y+H/2 & 0 & 1 \end{bmatrix}$$

(x, y, z, w) =(x, y, z, 1)*W*V*P*VP
X=x/w; Y=y/w; Z=z/w; - экранные координаты

4. Lighting

i. Light Types

Directional Light
Point Light. Attenuation:

$Att = 1/(att0_i + att1_i * d + att2_i * d^2)$

SpotLight

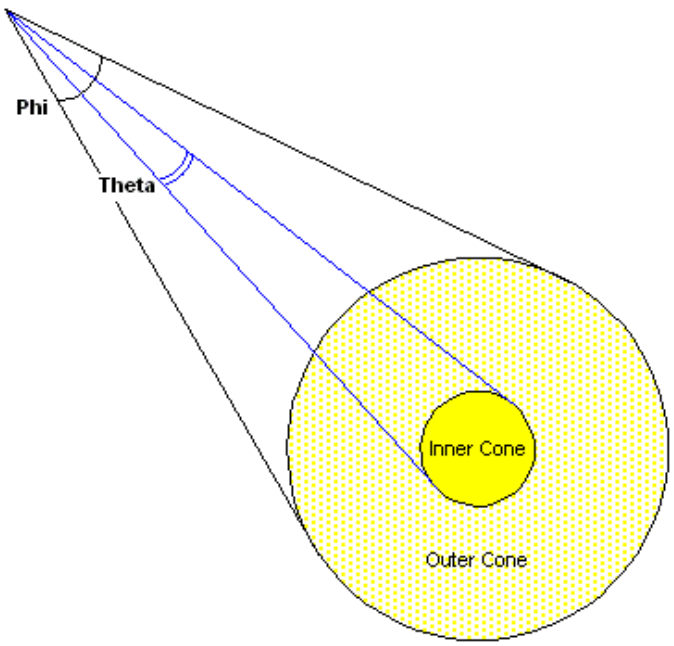
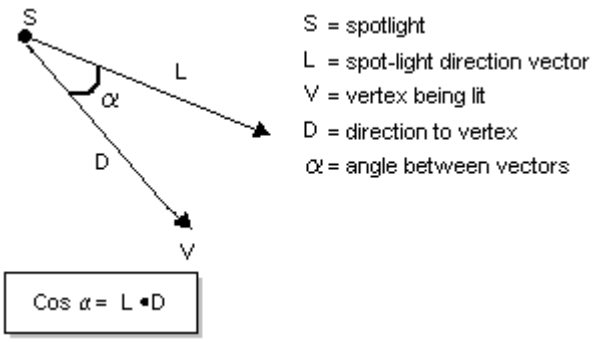


Рис. 4.1



Spot = f(cos(α))

j. Компоненты освещения

Global Illumination = Ambient Light + Diffuse Light + Specular Light + Emissive Light

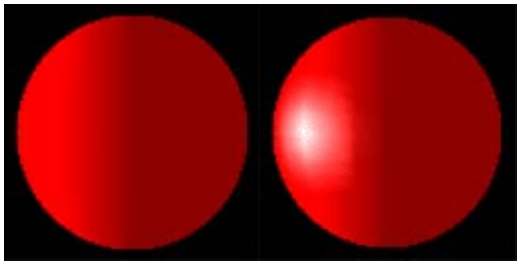


Рис. 4.2

Ambient Lighting = C_a*[G_a + sum(Att_i*$Spot_i$*L_{ai})]

C_a Material ambient color
G_a Global ambient color
Att_i Light attenuation of the ith light..
$Spot_i$ Spotlight factor of the ith light..
L_{ai} Light ambient color of the *i*th light

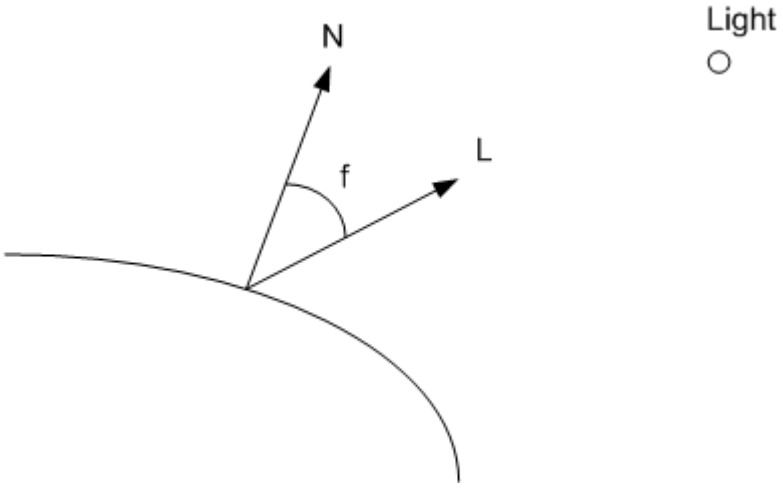


Рис. 4.3

Diffuse Lighting = C_d*sum[L_d*(N·L)*Att*Spot]

C_d Diffuse color.
L_d Light diffuse color.
N Vertex normal
L Direction vector from object vertex to the light.
Att Light attenuation.
Spot Spotlight factor..

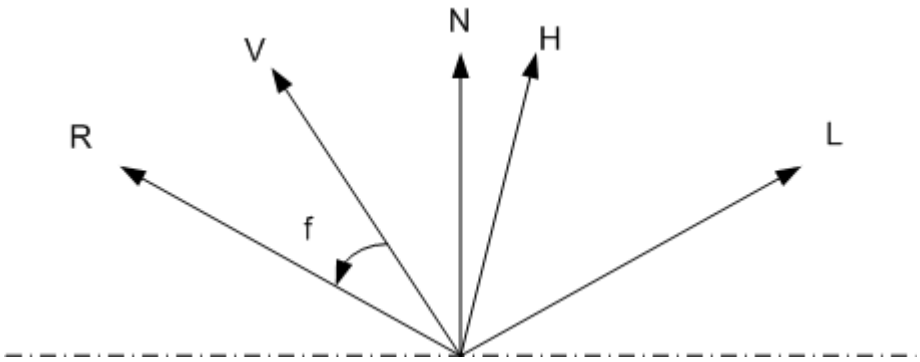


Рис. 4.4

$Cos^p(f) = (V,R)^p$

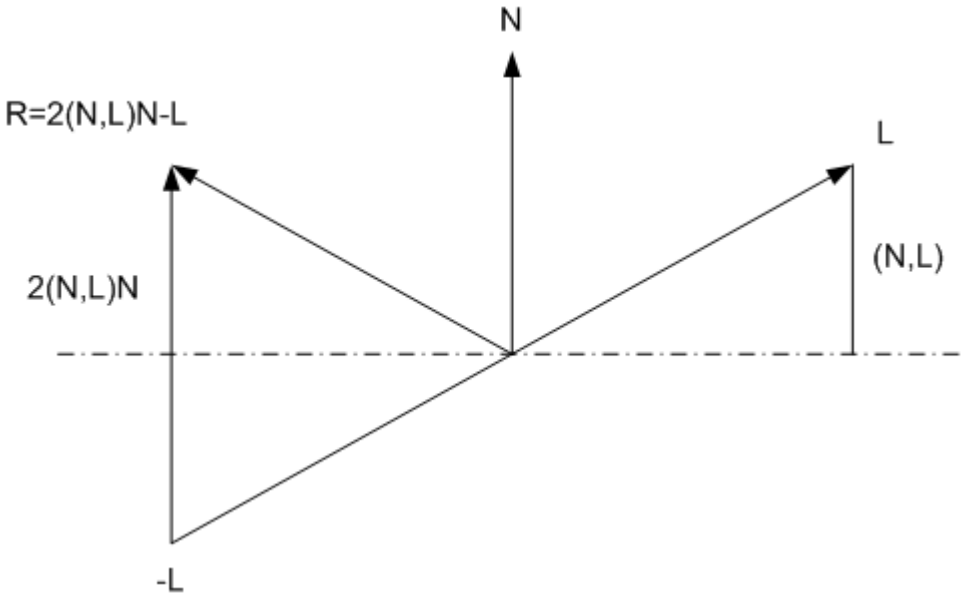


Рис. 4.5

H=L+N

$(R,V)^p \approx (N,H)^{4p}$

Specular Lighting = C_s * sum[L_s * $(N \bullet H)^P$ * *Att* * *Spot*]

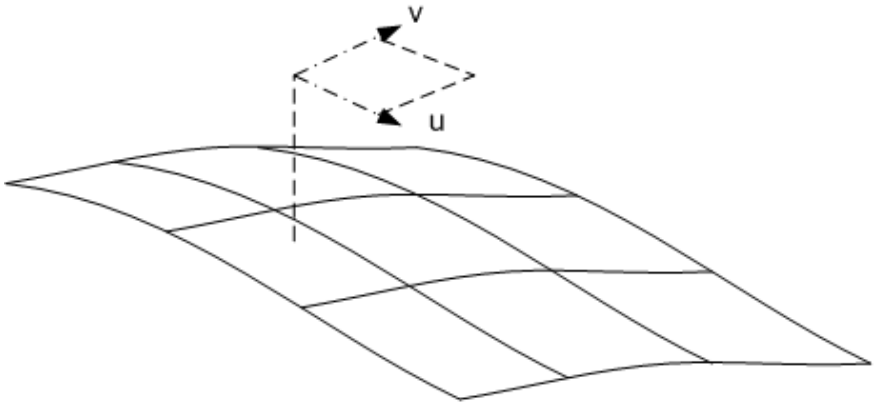
C_s Specular color.
N Vertex normal.
H Half way vector. See the section on the halfway vector.
P Specular reflection power. Range is 0 to +infinity
L_s Light specular color.
Att Light attenuation value.
$Spot$ Spotlight factor.

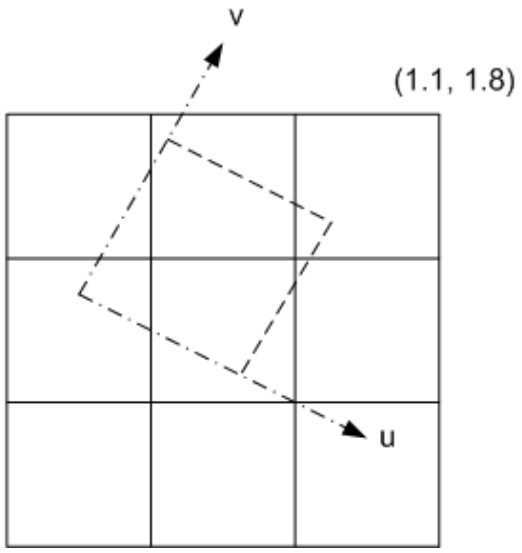
k. Camera Space Transformations

v=**v***WV; WV= W*V
N=N* WV
N=N* $(WV^{-1})'$
L=L*V - Directional Light
L=L_p*V-**v**- PointLight, SpotLight
X=norm(X);

5. Textures

l. Texture mapping





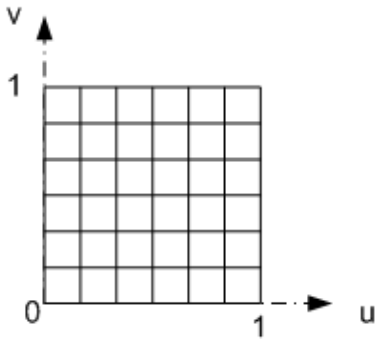


Рис. 5.1

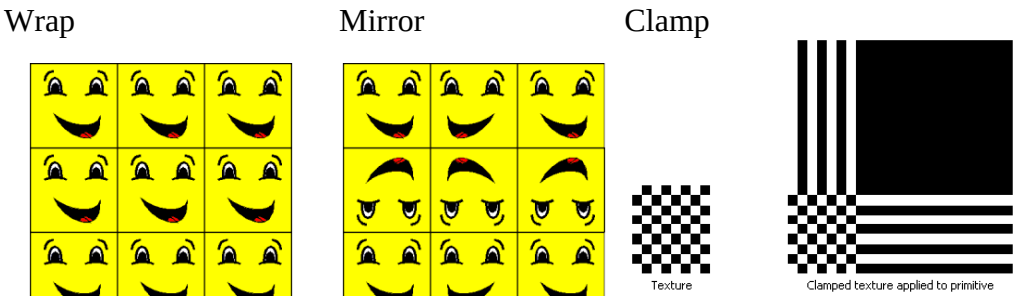


Рис. 5.2

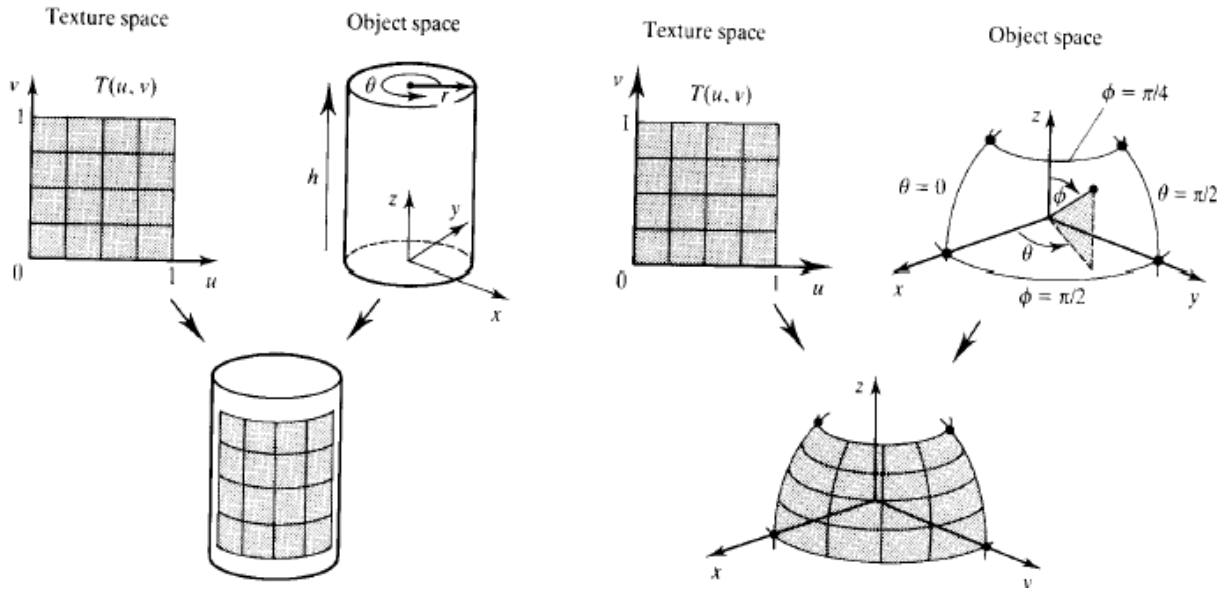


Рис. 5.3

(u,v)=(t/(2*Pi*n), z/(h*m))
x=r*cos(t)
y=r*sin(t)
t=arctg(x/y)

m. Texture Filtering

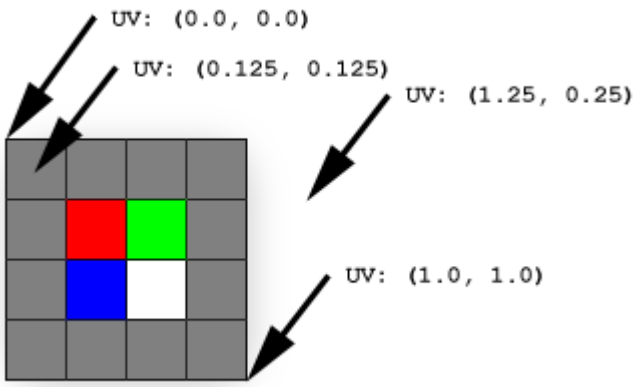


Рис. 5.4

Point Sampling, Linear (Bilinear) Filtering

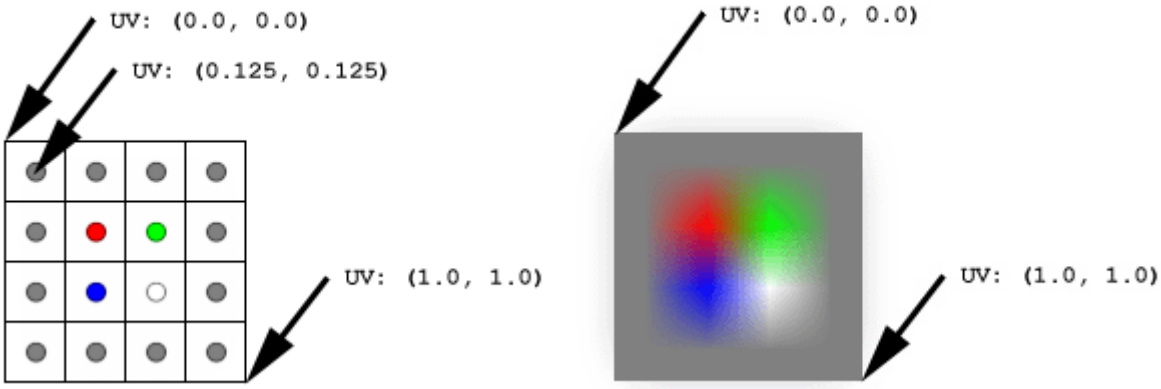


Рис. 5.5

Texture Filtering with Mipmaps

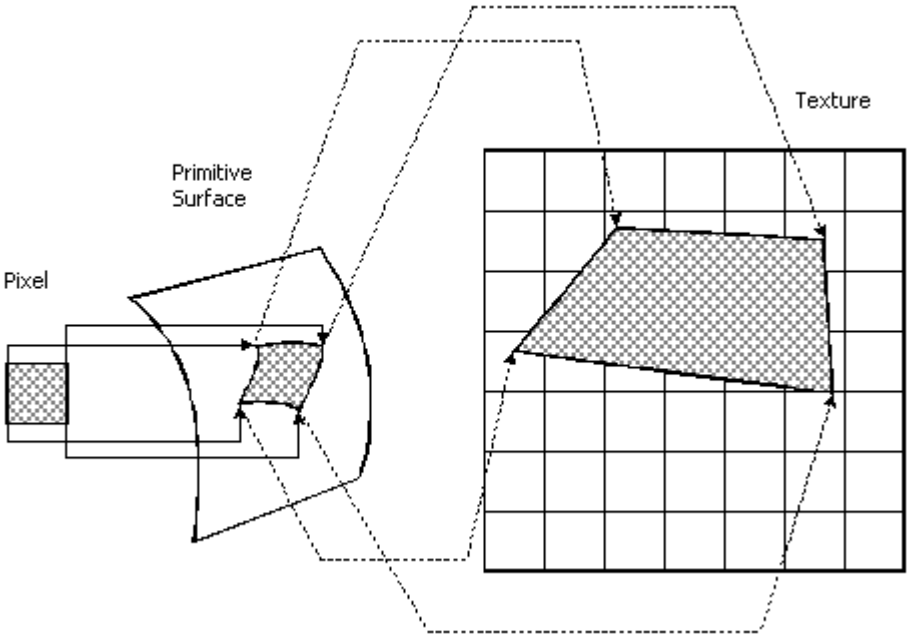


Рис. 5.6

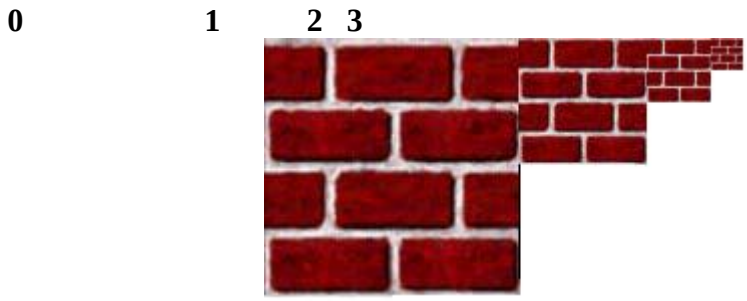


Рис. 5.7

Color = t*Texture$_i$[u,v] + (t-1)*Texture$_{i-1}$[u,v]; - Trilinear Filtering

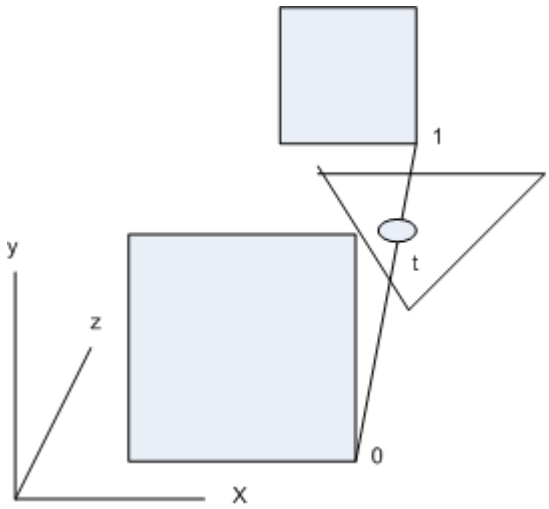


Рис. 5.8

6. Blending

Texture blending

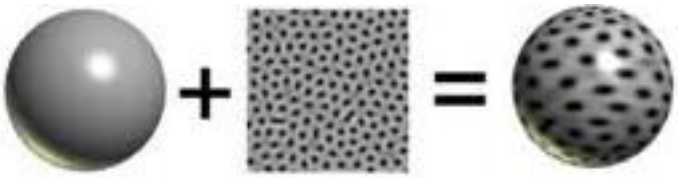


Рис. 6.1 Color * Texture[u,v]

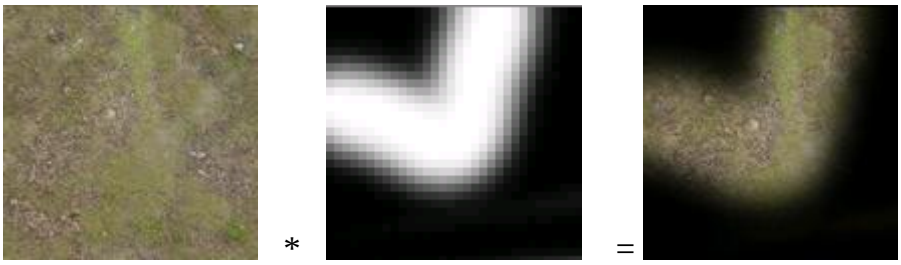


Рис. 6.2 Texture1[u,v] * Texture2[u,v]

DiffuseColor + SpecularColor * T

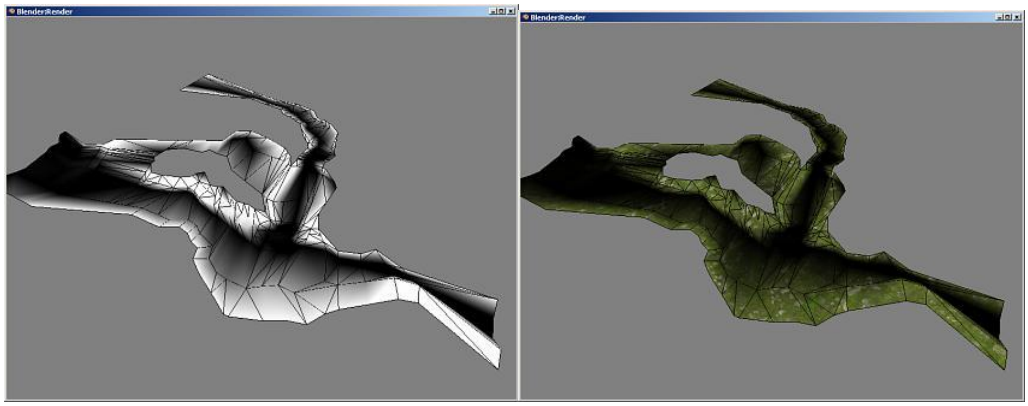


Рис. 6.3 Texture1[u,v] * Texture2[u,v]

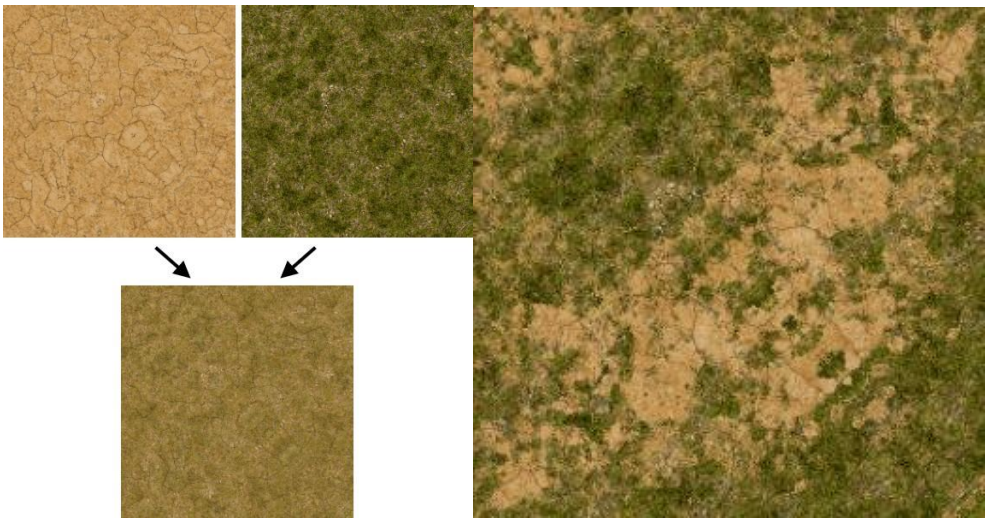


Рис. 6.4. a * Texture1[u,v] + (1-a) * Texture2[u,v], a= Texture3[u,v]

Рис. 6.5.

Alpha Blending (с буфером экрана)

C=(R,G,B,A)
c=(R,G,B)
c=A*c1+(1-A)*c2

Рис. 6.6

- Лодка под водой (A в вершинах воды, сортировка)
- Елки (A в текстуре)
- Буквы (A в текстуре + antialiasing)

Z-test проблемы
Alfa - test

Alpha Texture Blending

a ***** Texture1[u,v] + (1-a) * Texture2[u,v], a= Texture1[u,v].a

7. Fog and antialiasing

a. Fog

- Формулы
  F=Кусочно-линейная
  F=Exp(-d*density)
  F=Exp(-d*density)^2

C=F*C + (1-F)*FogColor

- Vertex fog
  Range-based (Direct3D uses the actual distance from the viewpoint to a vertex)
- Per pixel fog
  Вычисления в драйвере

b. Antialiasing

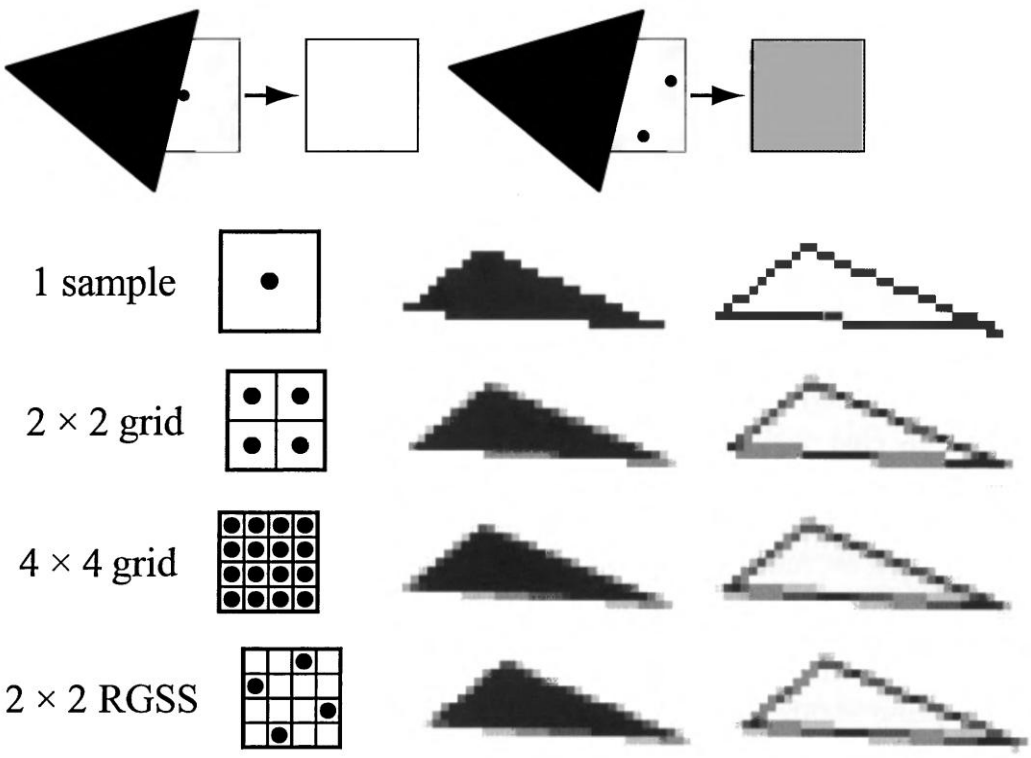


Рис. 7.1

8. Clipping

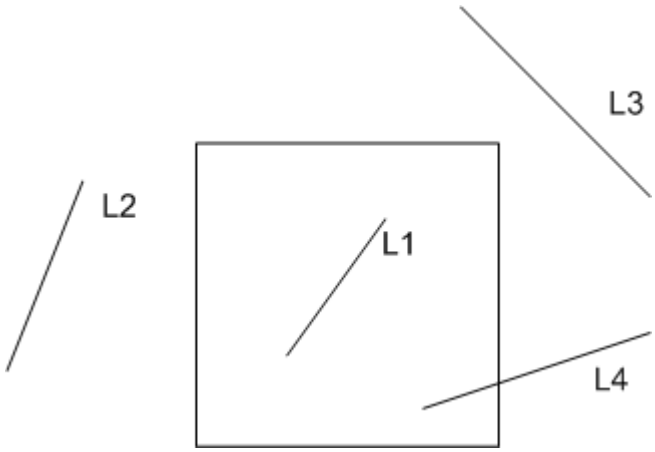


Рис. 8.1

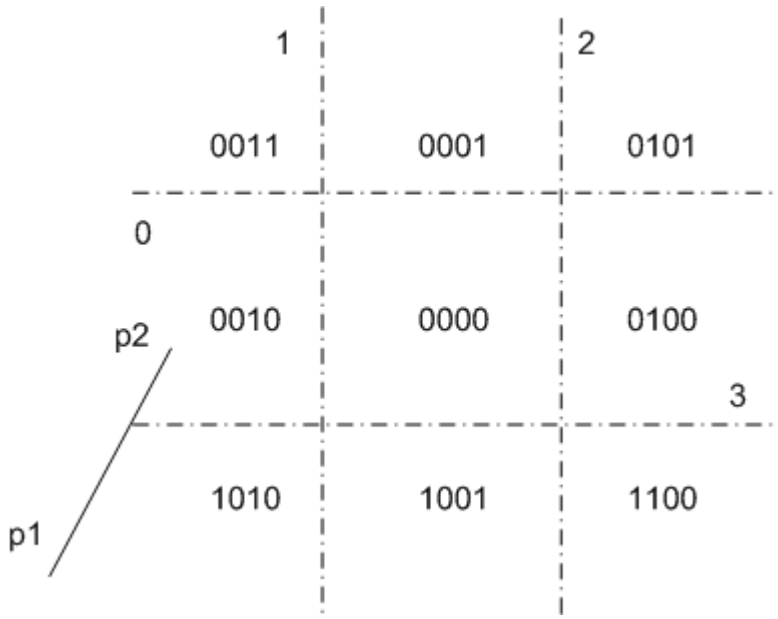


Рис. 8.2

```
If (p1.ClipCode & p2.ClipCode) culling
Else
{
        If(p1.ClipCode | p2.ClipCode == 0) PutLine
        Else
                For (i=0; i<4;i++) cut
                        If (p1.ClipCode[i]!= p2.ClipCode[i]) cut
 }
```

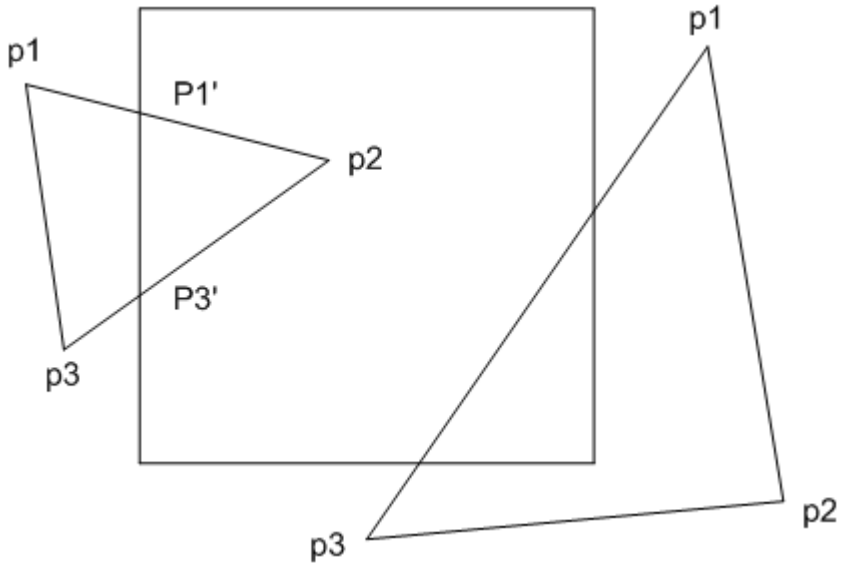


Рис. 8.3

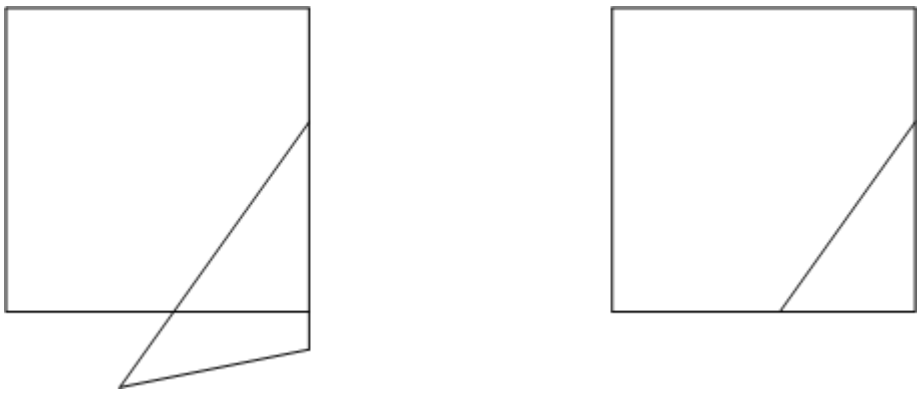


Рис. 8.4

(x, y, z, w) =(x, y, z, 1)*P ---------- (2.5)

-w<x<w
-w<y<w
0<z<w

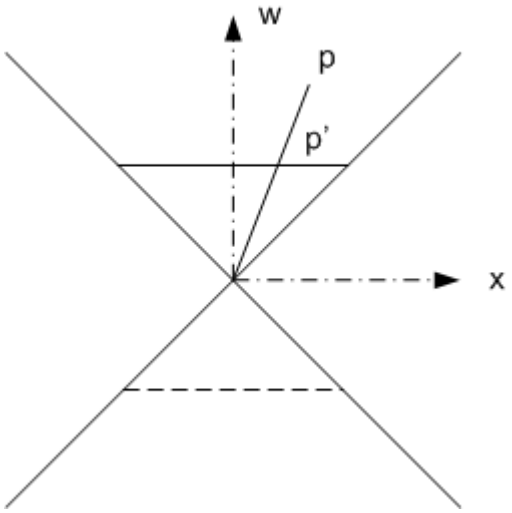


Рис. 8.5

w>0

x=x1+t(x2-x1)
y=y1+t(y2-y1)
z=z1+t(z2-z1)
w=w1+t(w2-w1)

w>0: 0=w1+t(w2-w1) =>t=>x,y,z
x=w: x1+t(x2-x1)= w1+t(w2-w1) =>t=>x,y,z