

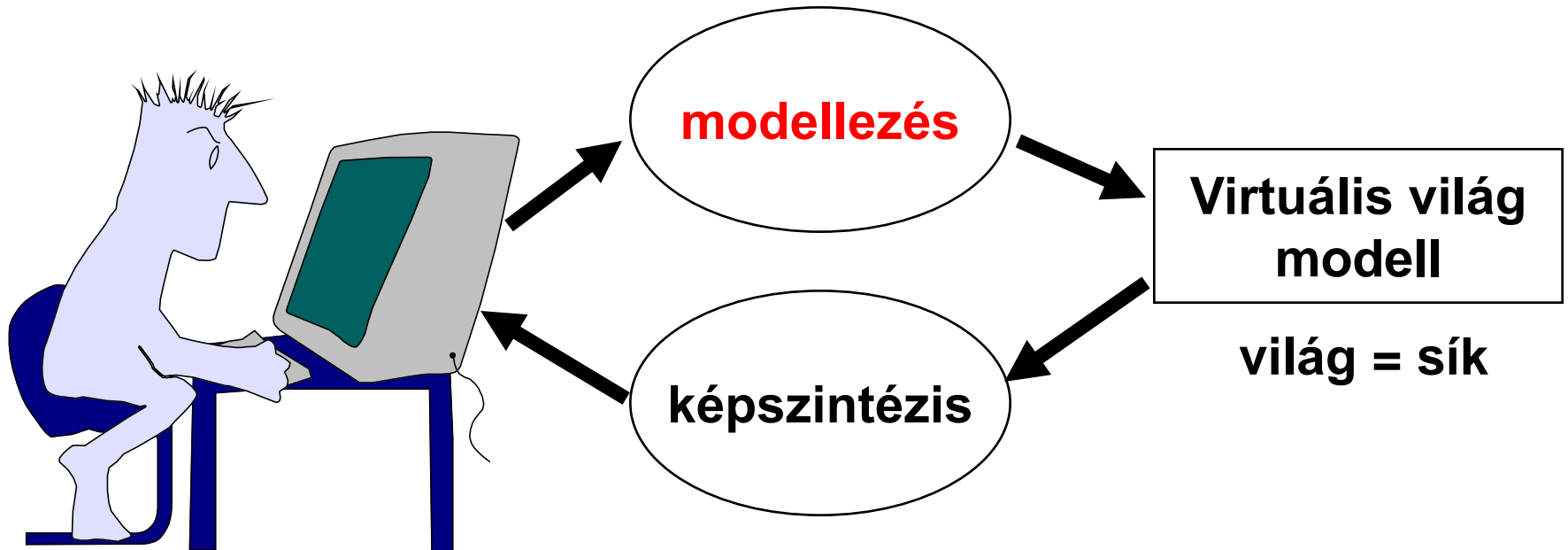
A számítógépes grafika alapjai

Textúrák a grafikában

Előadó: Benedek Csaba

Tananyag : Szirmay-Kalos László, Benedek Csaba

Textúrázás



Metafórák:

- **2D rajzolás**

Textúra leképezés

- Összetett mintázatú felületek (pl perzsaszőnyeg) BRDF-leírása nehézkes
 - bonyolult modellezés, hosszú képszintézis ☹️
- Megoldás: textúra
 - Bittérképes textúra: 2-D képet rendelünk a felülethez, a képpontok a felületelem színét tartalmazzák
 - Procedurális textúra: generáló műveletsorral írjuk le a mintázatot

Textúra leképezés:

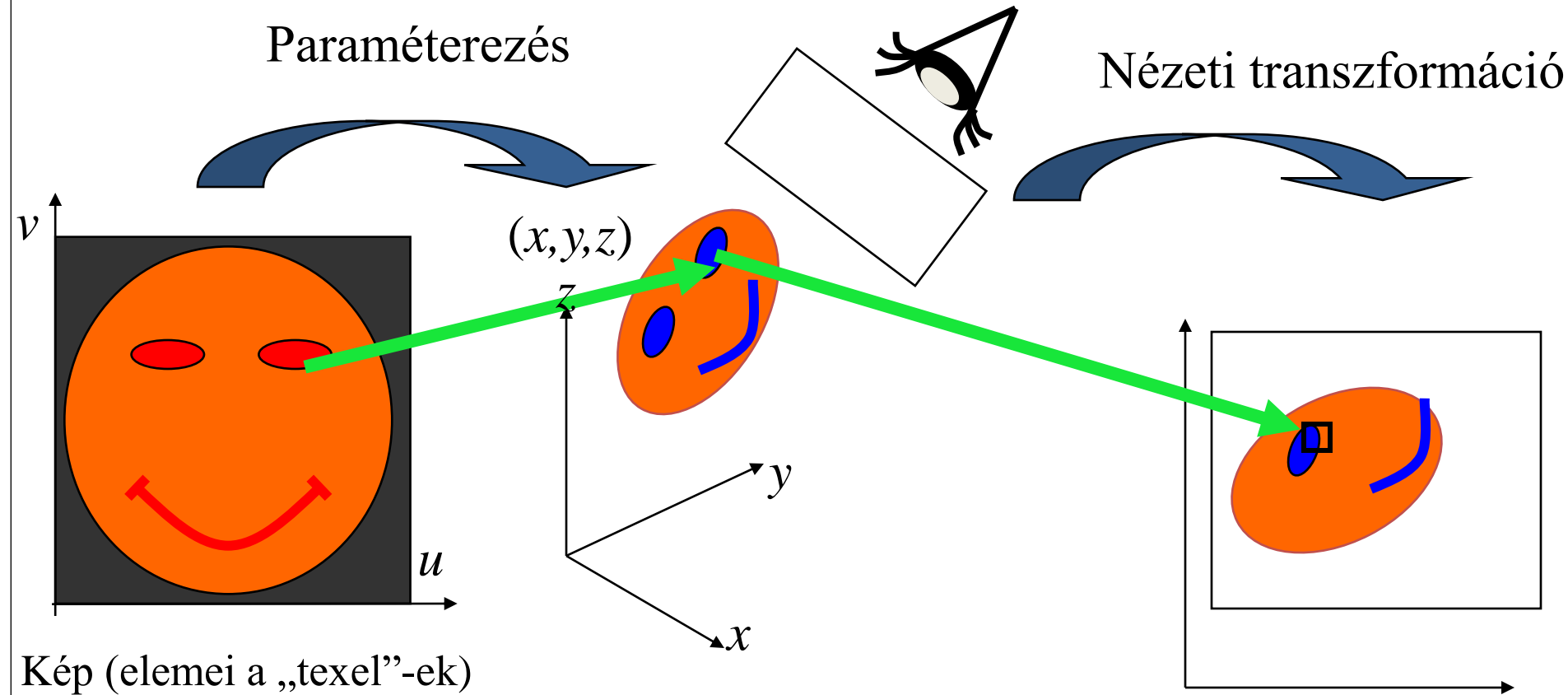
anyagjellemzők változnak a felületen



(2D) Textúra leképzés

Paraméterezés

Nézeti transzformáció



Kép (elemei a „texel”-ek)
vagy függvény

$$x = f_x(u, v)$$

$$y = f_y(u, v)$$

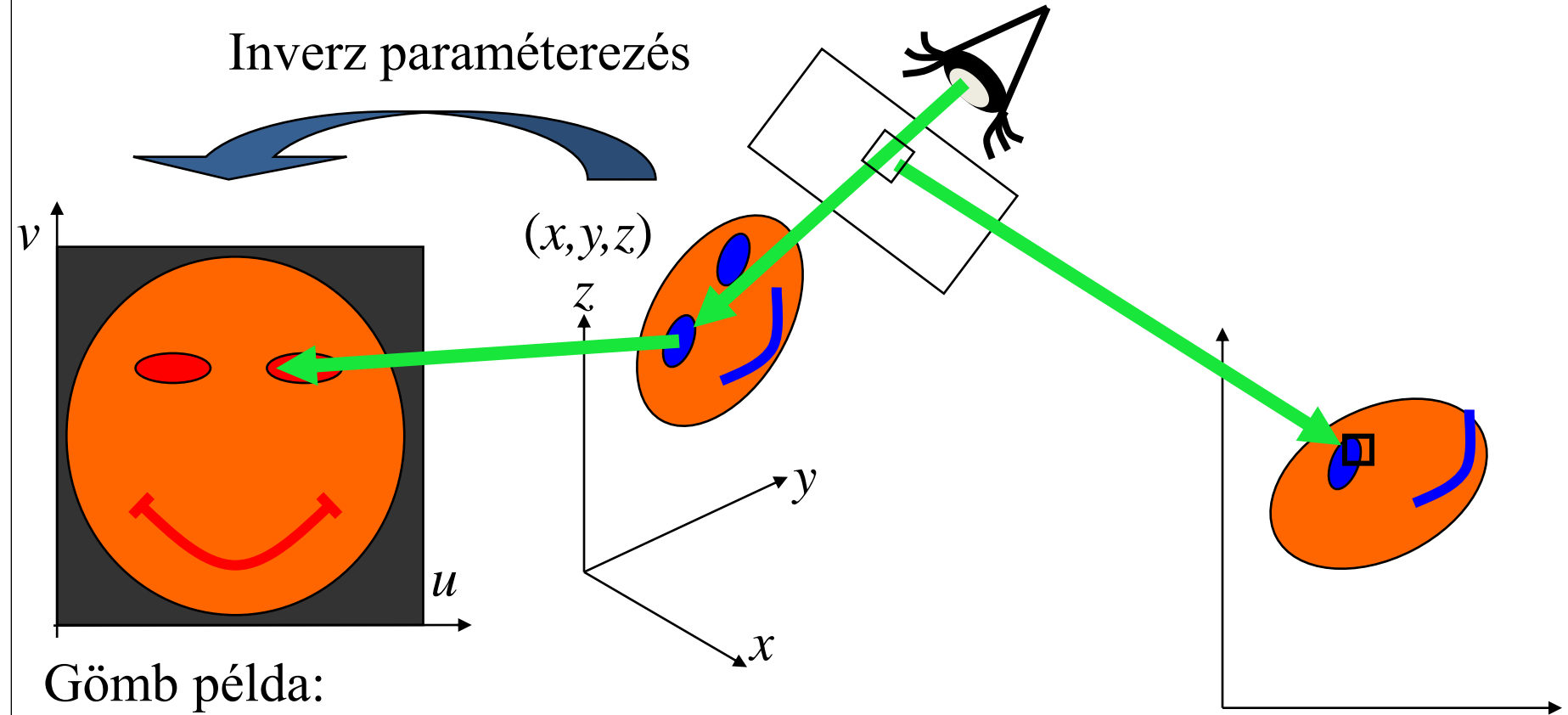
$$z = f_z(u, v) \quad u, v \in [0, 1]$$

Textúra-tér - képtér

- **Textúra alapú leképzés:** a textúra-térben levő ponthoz keresi meg a hozzá tartozó pixelt
 - Hatékony, de nem garantálja hogy a textúra-térben kijelölt pontok a képernyőn is egyenletesen helyezkednek el („lyukas lehet a kép”)
- **A képtér alapú leképzés:** a pixelhez keresi meg a hozzá tartozó textúra elemet
 - Vetítési transzformáció inverze!

Textúra leképzés sugárkövetésnél

Inverz paraméterezés



Gömb példa:

$$x = x_0 + r \cos 2\pi u \sin \pi v$$

$$y = y_0 + r \sin 2\pi u \sin \pi v$$

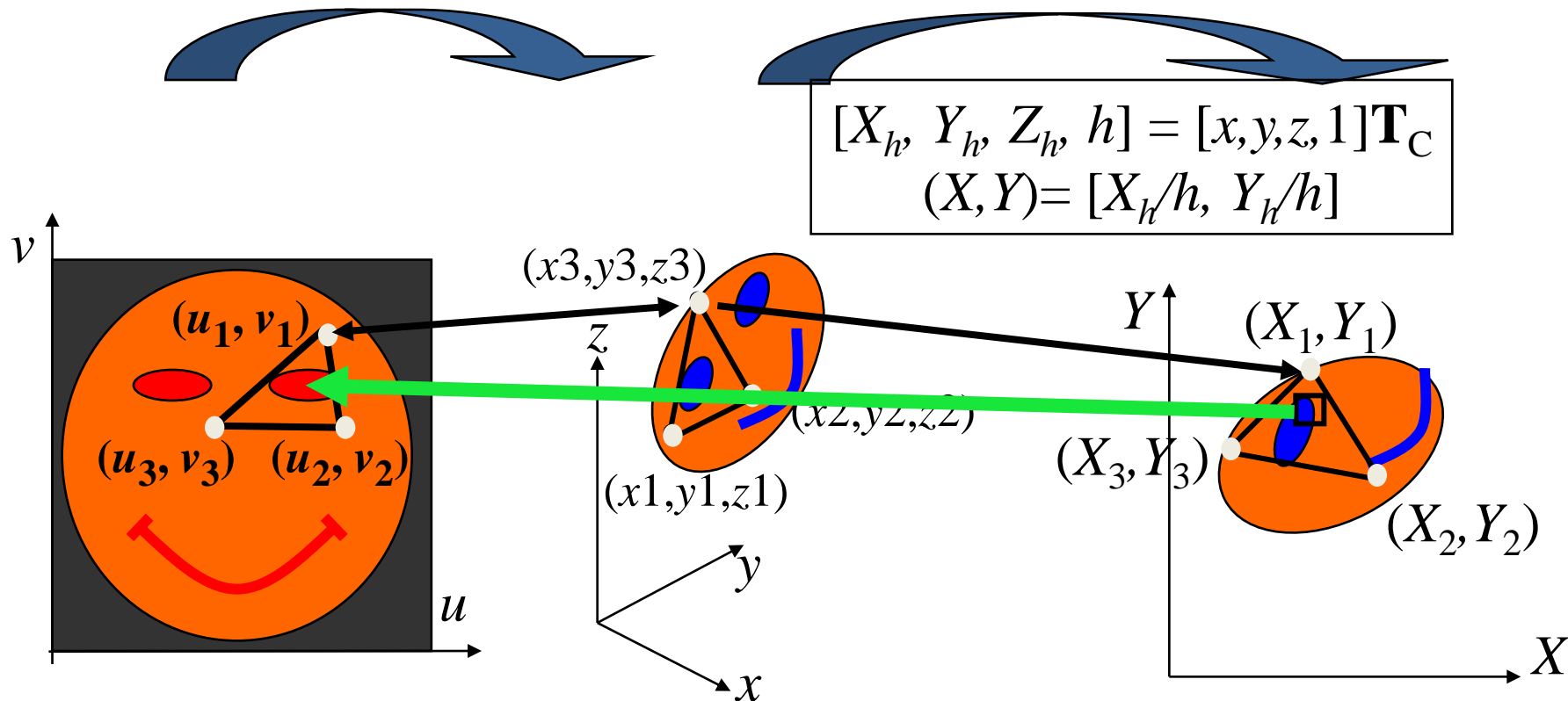
$$z = z_0 + r \cos \pi v \quad u, v \in [0, 1]$$

$$\left. \begin{array}{l} x = x_0 + r \cos 2\pi u \sin \pi v \\ y = y_0 + r \sin 2\pi u \sin \pi v \\ z = z_0 + r \cos \pi v \end{array} \right\} \begin{array}{l} u = 1/2\pi (\text{atan2}((y-y_0), (x-x_0)) + \pi) \\ v = 1/\pi \arccos((z-z_0)/r) \end{array}$$

Inkrementális képszintézis

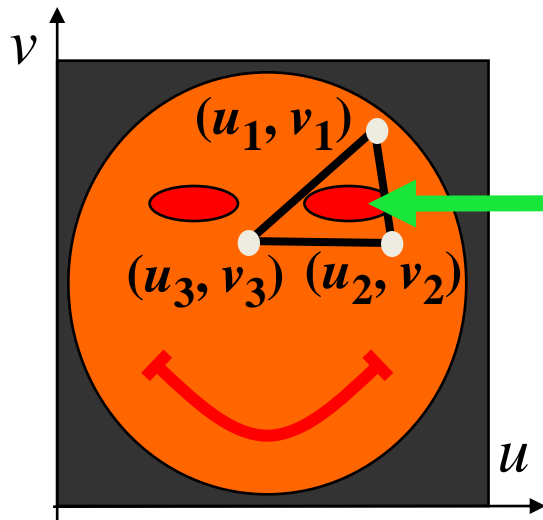
Paraméterezés

Modell-világ-képernyő transzf.



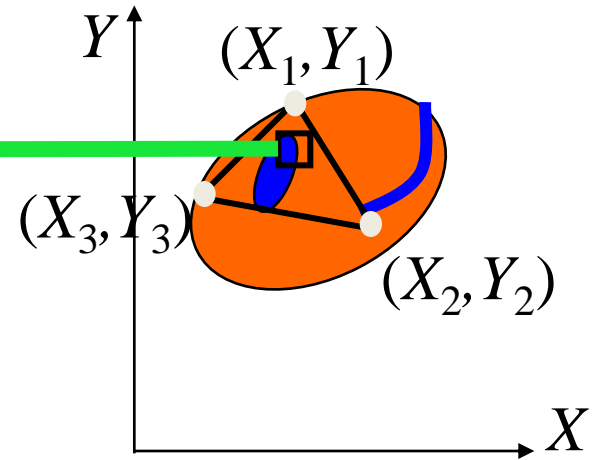
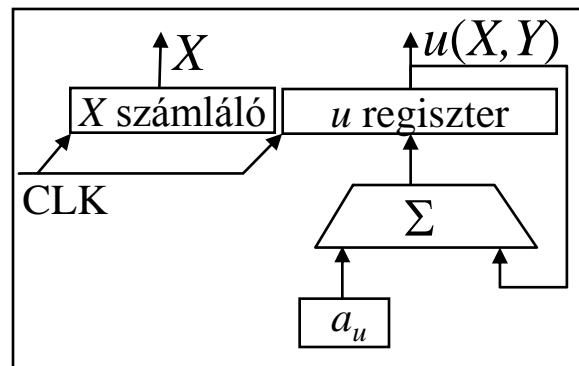
Lineáris interpoláció

Leképezés: 2x3 mátrix
6 egyenlet 6 ismeretlen



$$u = a_u X + b_u Y + c_u$$

$$v = a_v X + b_v Y + c_v$$



Interpolációs feltétel:

$$u_1 = a_u X_1 + b_u Y_1 + c_u$$

$$u_2 = a_u X_2 + b_u Y_2 + c_u$$

$$u_3 = a_u X_3 + b_u Y_3 + c_u$$

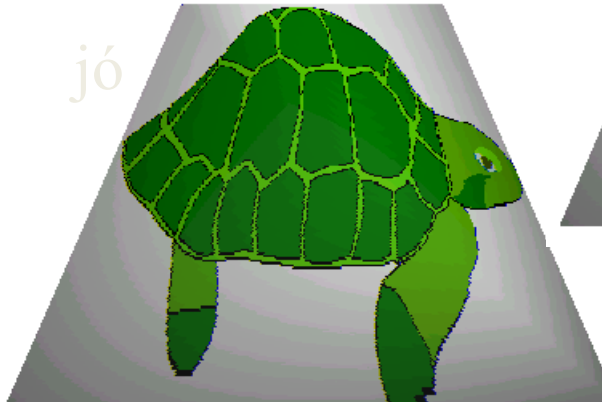
$$v_1 = a_v X_1 + b_v Y_1 + c_v$$

$$v_2 = a_v X_2 + b_v Y_2 + c_v$$

$$v_3 = a_v X_3 + b_v Y_3 + c_v$$

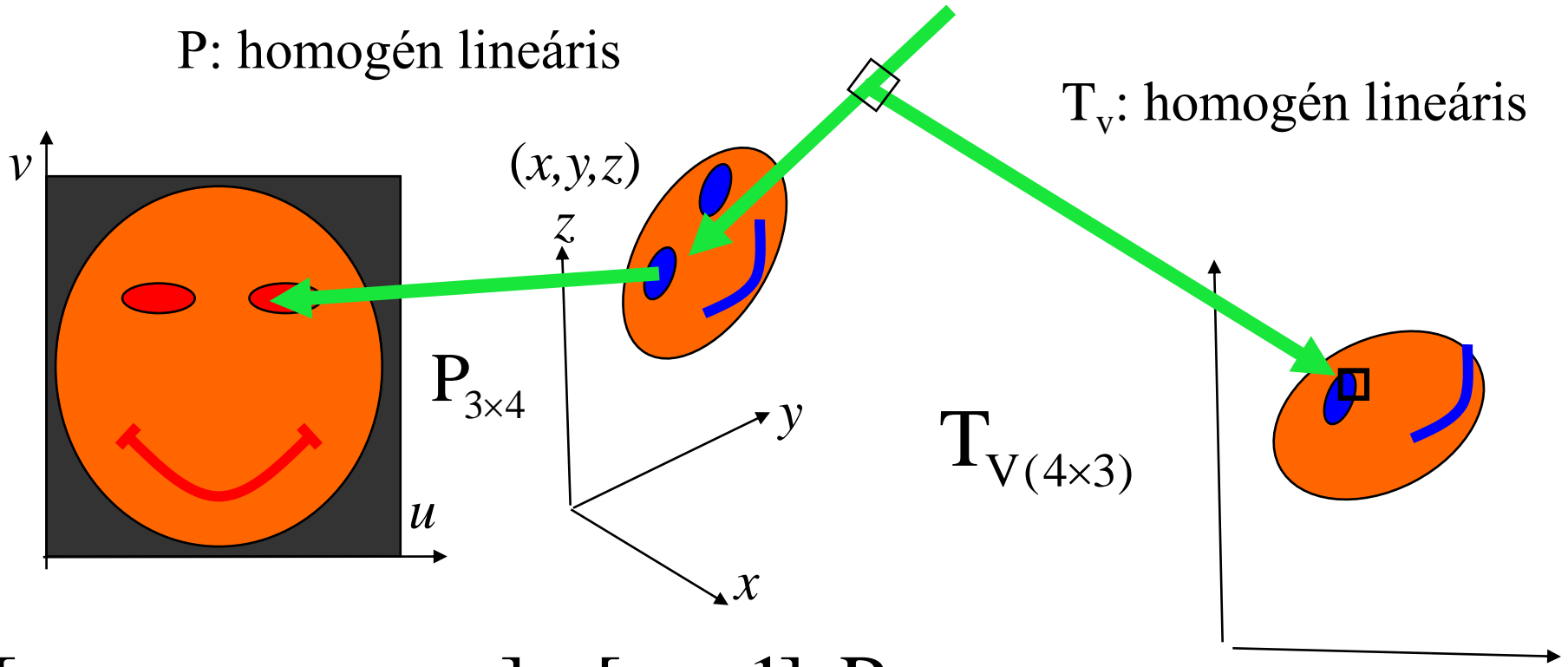
lineáris

jó



$[u, v]$ $[x, y, z]$ $[X, Y]$

P: homogén lineáris

 T_v : homogén lineáris

$$[x \cdot q, y \cdot q, z \cdot q, q] = [u, v, 1] \cdot P_{3 \times 4}$$

$$[X \cdot h, Y \cdot h, h] = [x \cdot q, y \cdot q, z \cdot q, q] \cdot T_{v(4 \times 3)}$$

$$[X \cdot h, Y \cdot h, h] = [u, v, 1] \cdot P_{3 \times 4} \cdot T_{v(4 \times 3)} = [u, v, 1] \cdot T_{c(3 \times 3)}$$

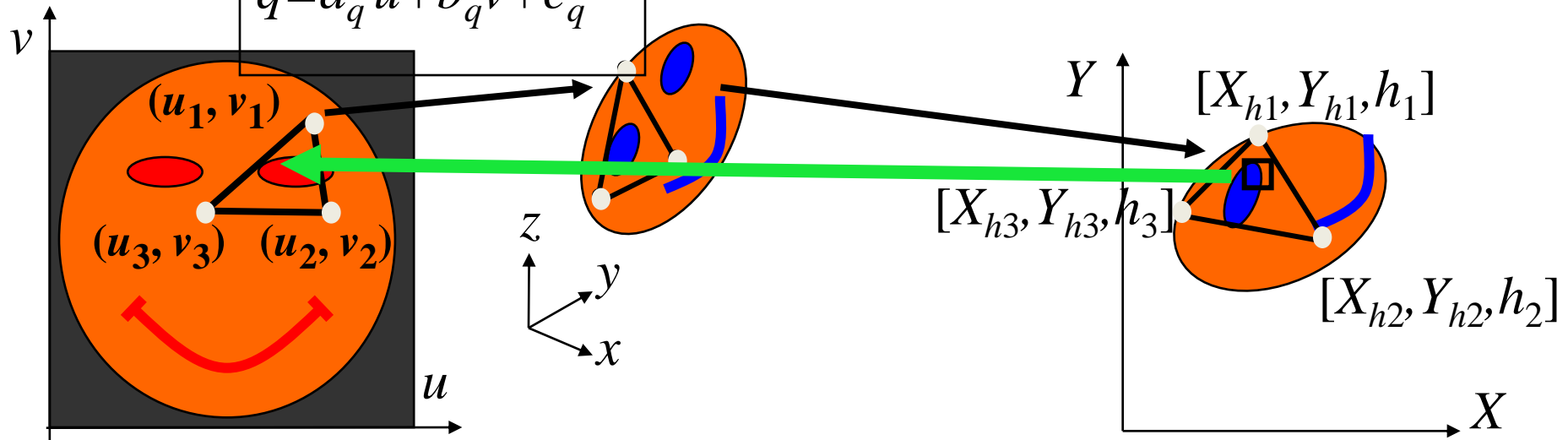
Perspektíva helyes textúrázás

Paraméterezés

Modell-világ-képernyő transzf.

$$\begin{aligned}x_q &= a_x u + b_x v + c_x \\y_q &= a_y u + b_y v + c_y \\z_q &= a_z u + b_z v + c_z \\q &= a_q u + b_q v + c_q\end{aligned}$$

$$\begin{aligned}[X_h, Y_h, h] &= [x_q, y_q, z_q, q] \mathbf{T}_{\mathbf{v}(4 \times 3)} \\(X, Y) &= [X_h/h, Y_h/h]\end{aligned}$$

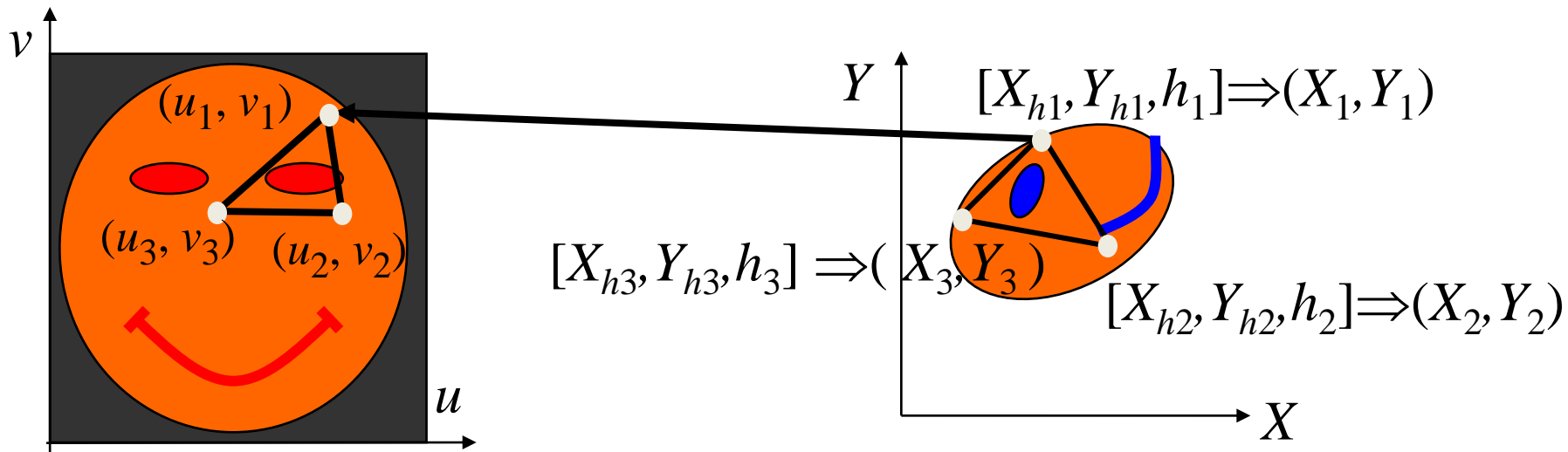


$$\begin{aligned}[u, v, 1] \mathbf{T}_{\mathbf{C}(3 \times 3)} &= [X_h, Y_h, h], \\[u/h, v/h, 1/h] &= [X, Y, 1] \mathbf{T}_{\mathbf{C}}^{-1}\end{aligned}$$

T_C^{-1} mátrixelemek

$$T_C^{-1} = \begin{pmatrix} a_u & a_v & a_h \\ b_u & b_v & b_h \\ c_u & c_v & c_h \end{pmatrix}$$

$$[u/h, v/h, 1/h] = [X, Y, 1] T_C^{-1}$$



$$\begin{aligned} u_1/h_1 &= a_u X_1 + b_u Y_1 + c_u \\ u_2/h_2 &= a_u X_2 + b_u Y_2 + c_u \\ u_3/h_3 &= a_u X_3 + b_u Y_3 + c_u \end{aligned}$$

$$a_u, b_u, c_u$$

$$\begin{aligned} v_1/h_1 &= a_v X_1 + b_v Y_1 + c_v \\ v_2/h_2 &= a_v X_2 + b_v Y_2 + c_v \\ v_3/h_3 &= a_v X_3 + b_v Y_3 + c_v \end{aligned}$$

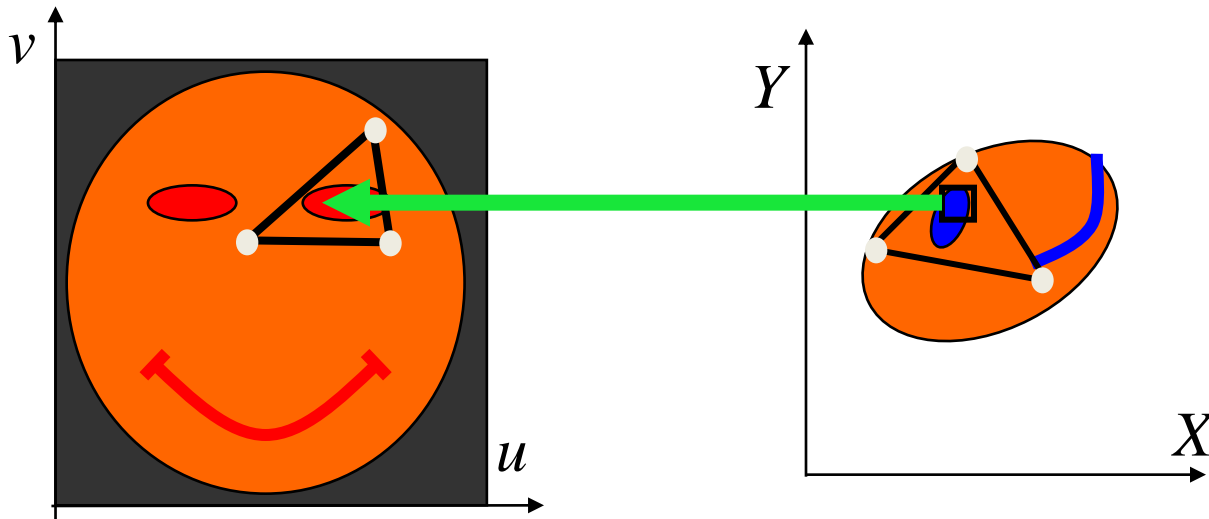
$$a_v, b_v, c_v$$

$$\begin{aligned} 1/h_1 &= a_h X_1 + b_h Y_1 + c_h \\ 1/h_2 &= a_h X_2 + b_h Y_2 + c_h \\ 1/h_3 &= a_h X_3 + b_h Y_3 + c_h \end{aligned}$$

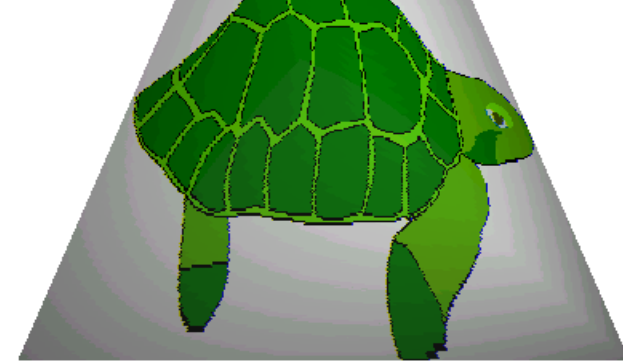
$$a_h, b_h, c_h$$

T_C^{-1} : leképzés: 3x3 mátrix 9 egyenlet 9 ismeretlen

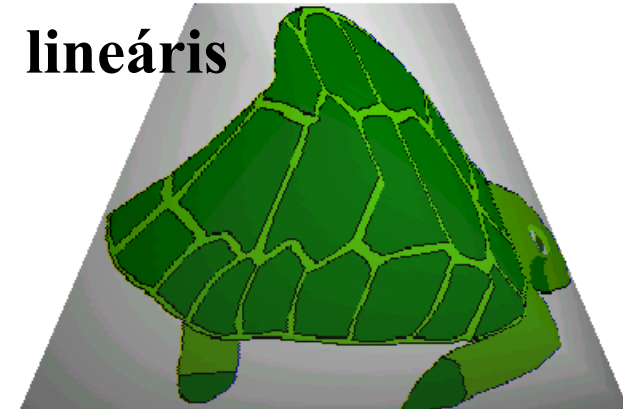
Perspektíva helyes textúrázás



Perspektív helyes



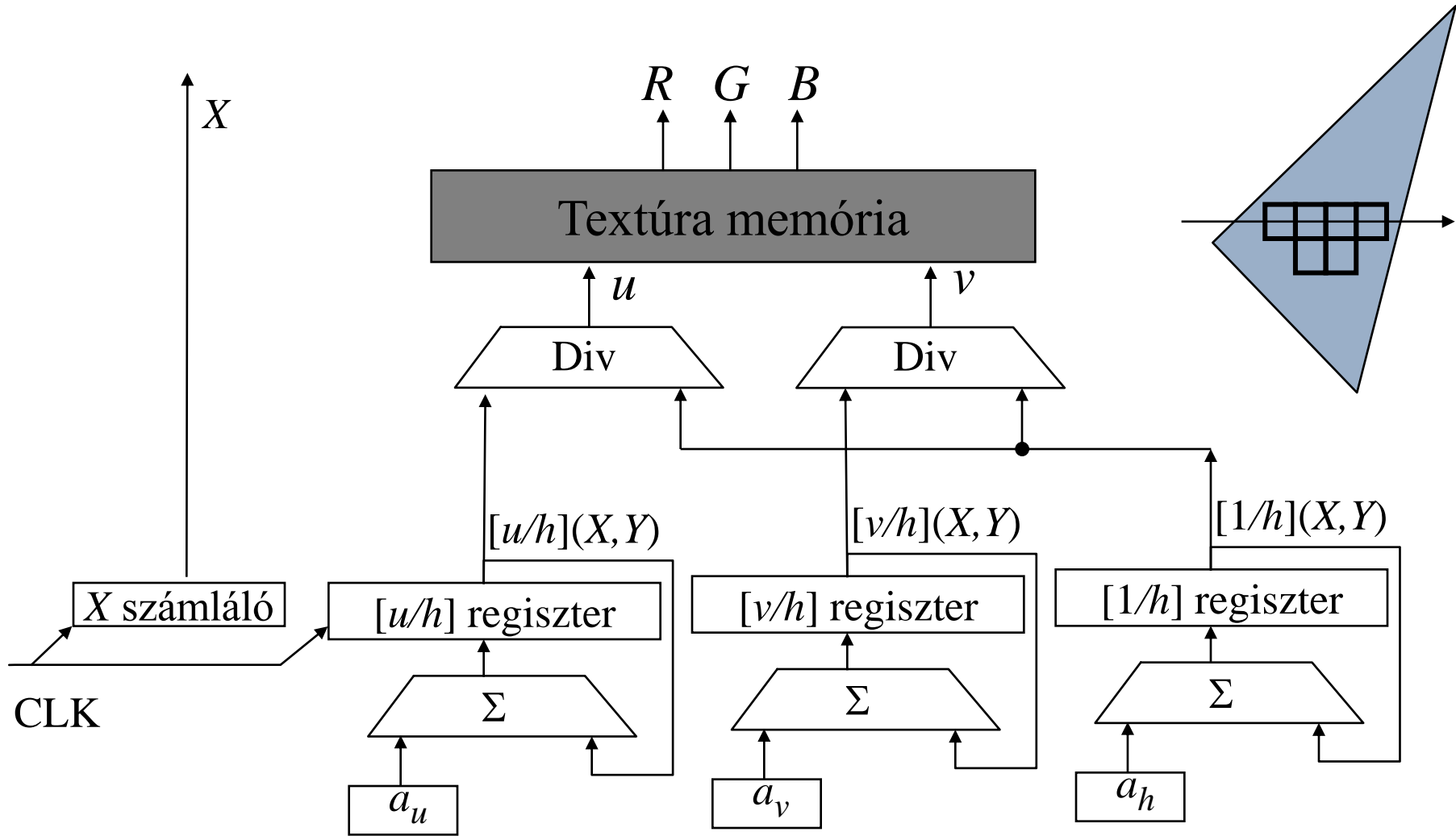
lineáris



$$\begin{aligned} u/h &= a_u X + b_u Y + c_u \\ v/h &= a_v X + b_v Y + c_v \\ 1/h &= a_h X + b_h Y + c_h \end{aligned}$$

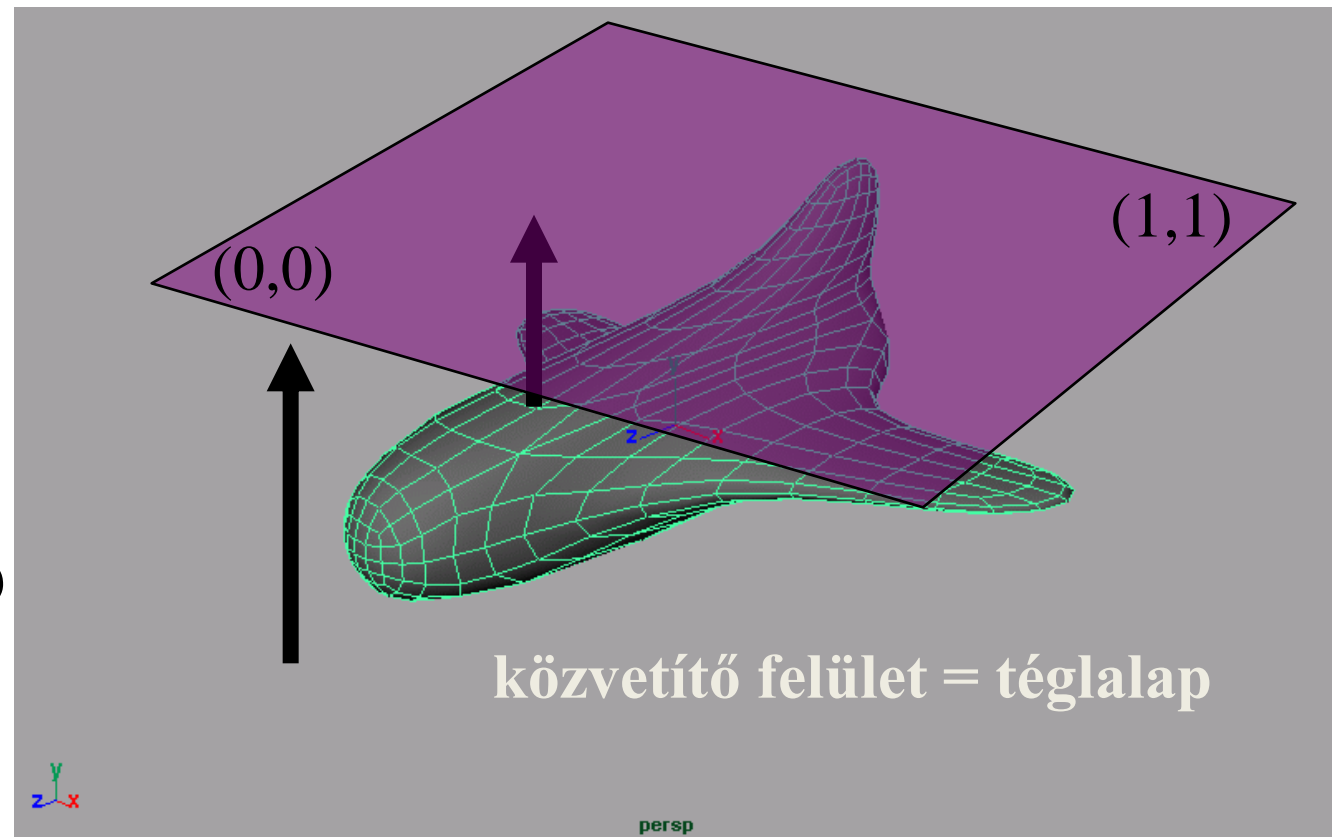
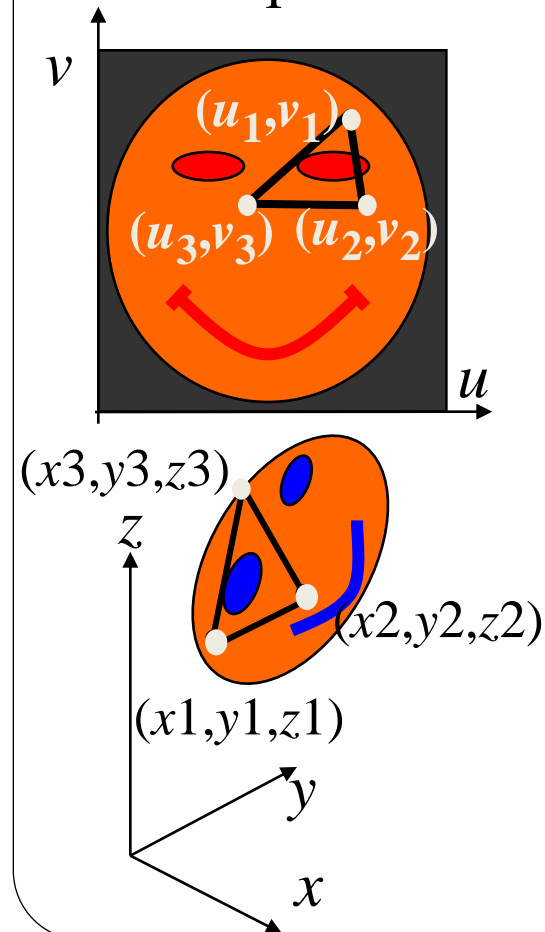
$$\begin{aligned} u &= \frac{a_u X + b_u Y + c_u}{a_h X + b_h Y + c_h} \\ v &= \frac{a_v X + b_v Y + c_v}{a_h X + b_h Y + c_h} \end{aligned}$$

Perspektíva helyes interpoláció hw.

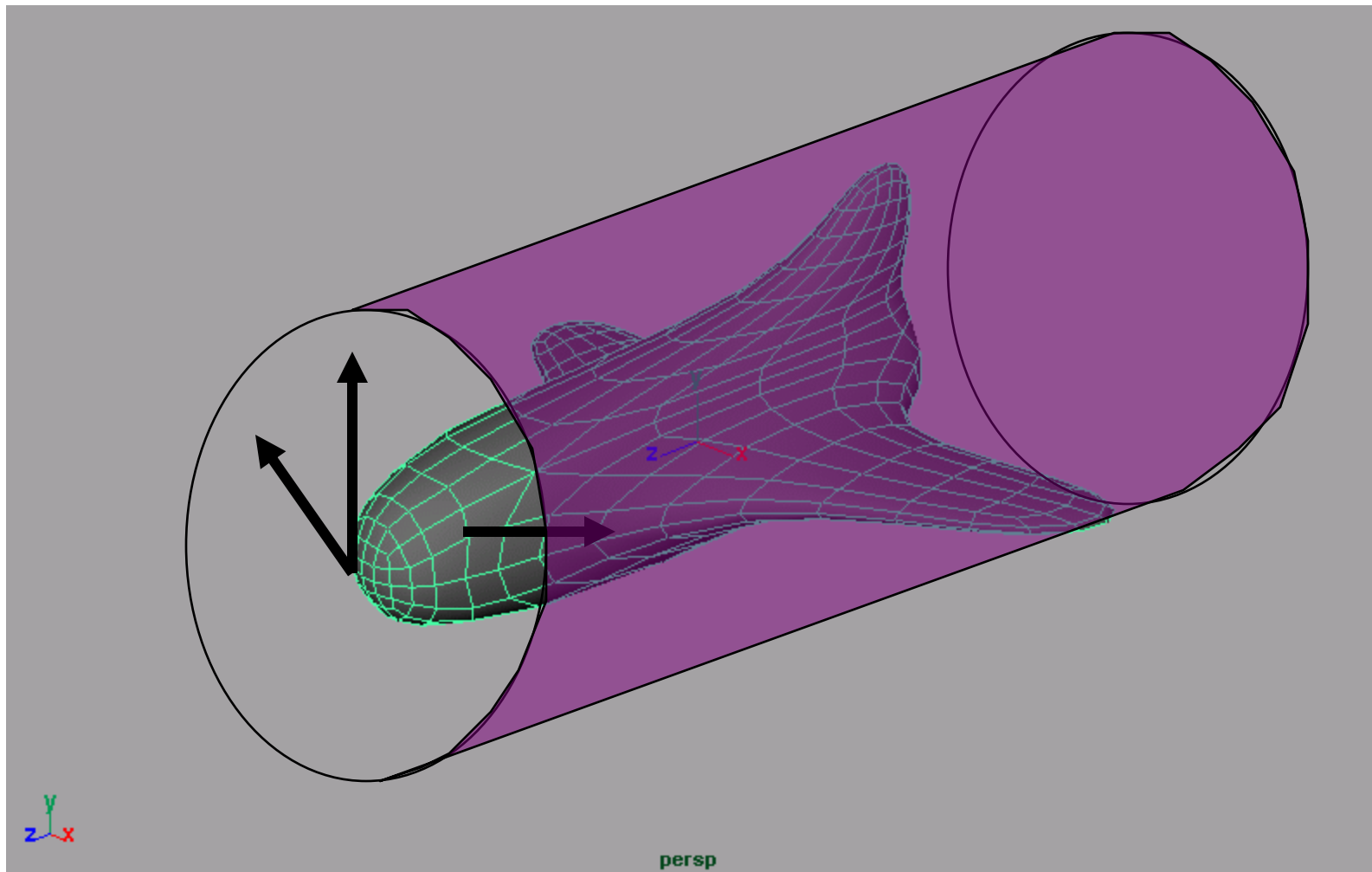


Textúra függvény definíciója

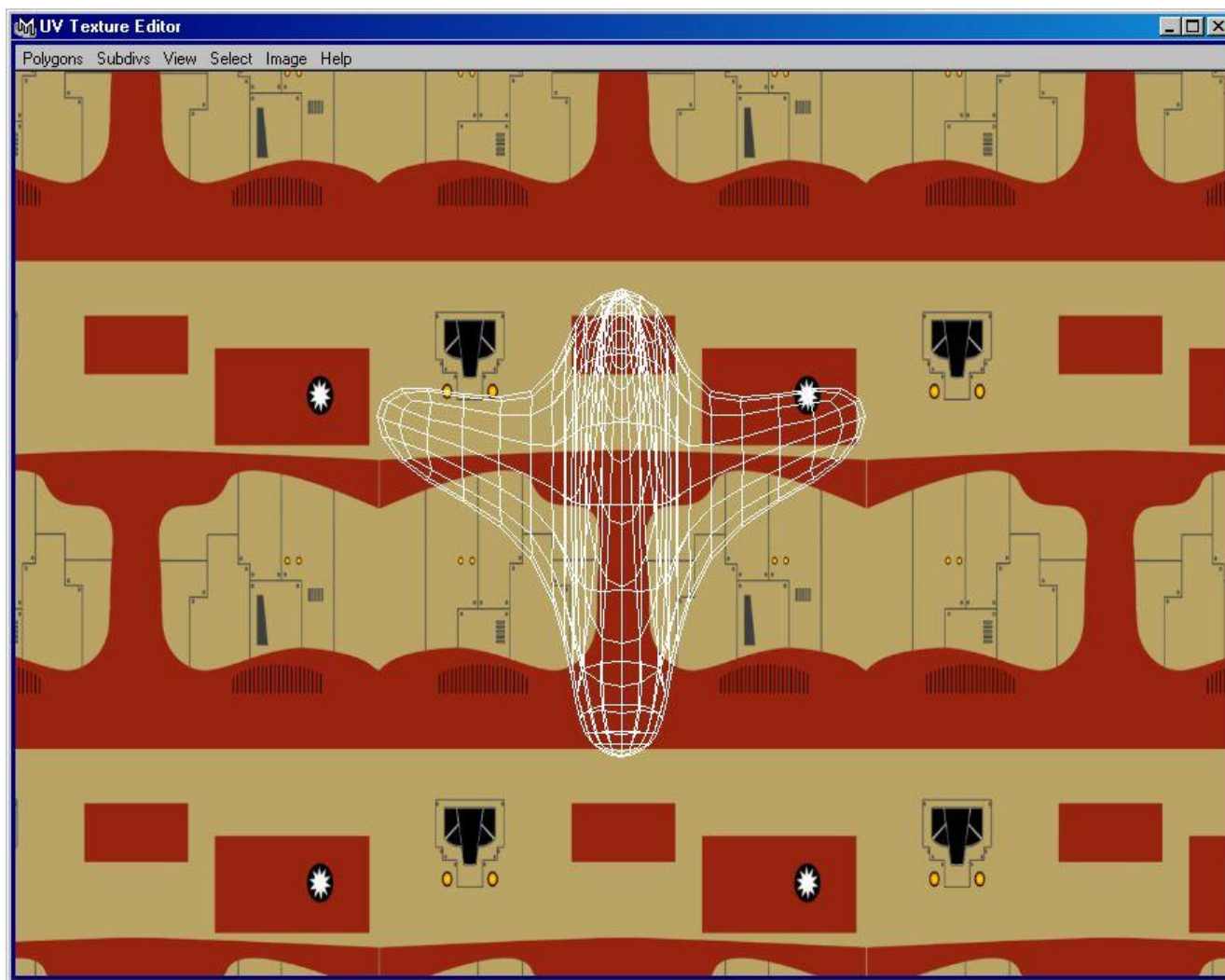
1. Parametrikus felületeknél triviális
2. Implicit felületeknél és poligonmodellnél: **közvetítő felületek**



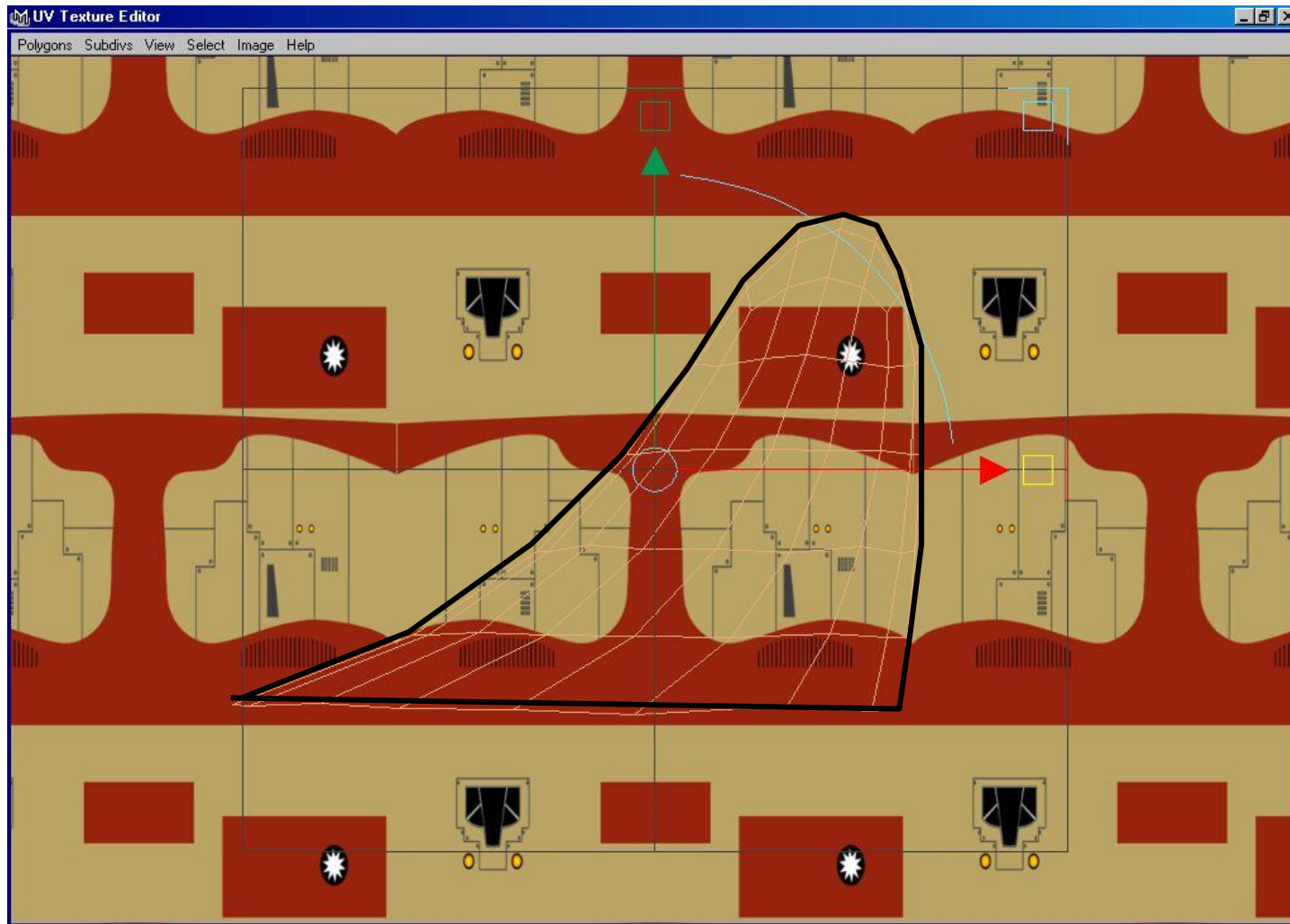
Henger (gömb) közvetítő felület



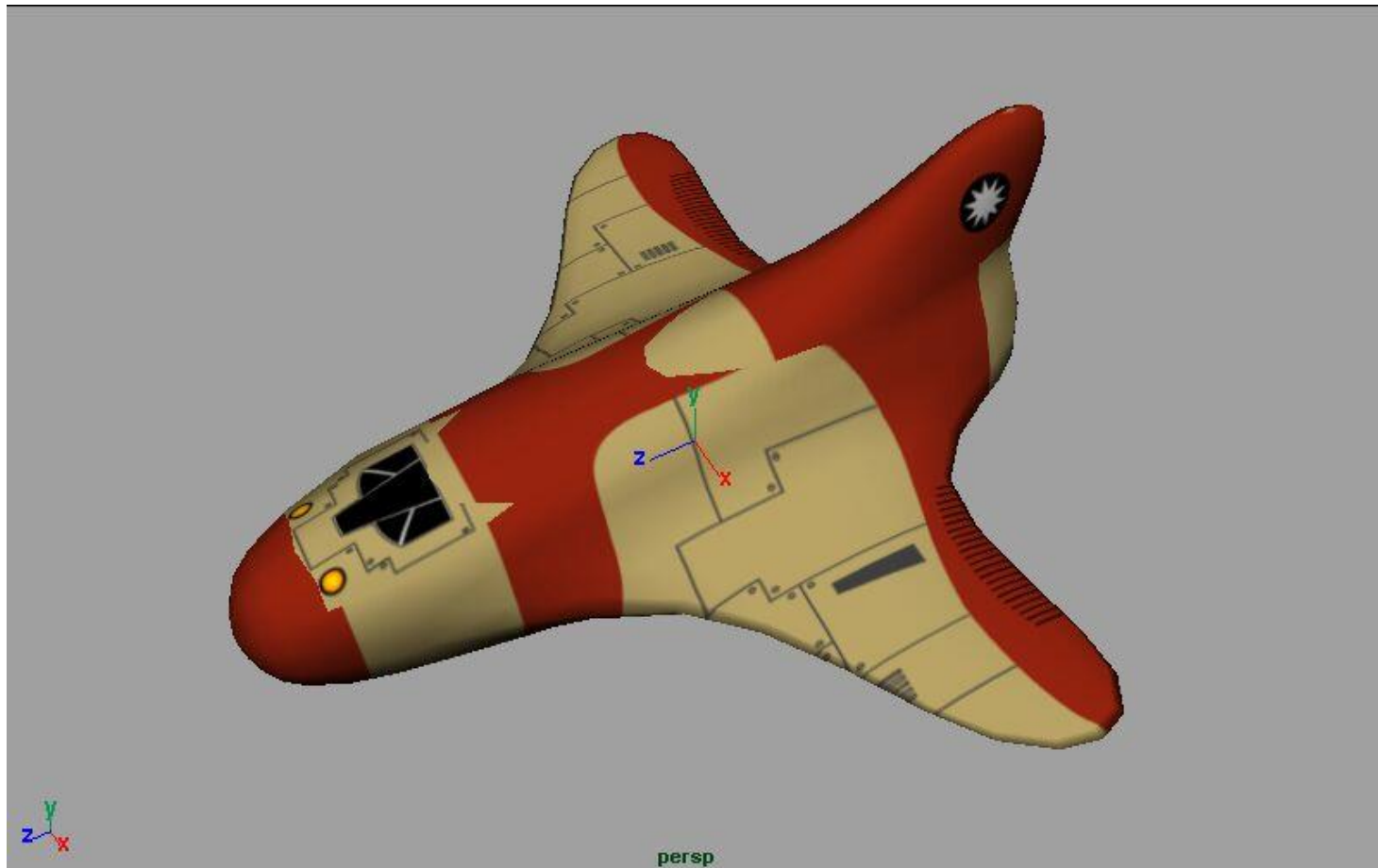
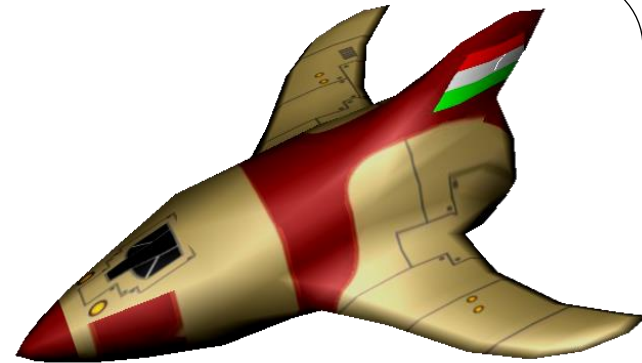
Téglalap közvetítő felület: textúratér nézet



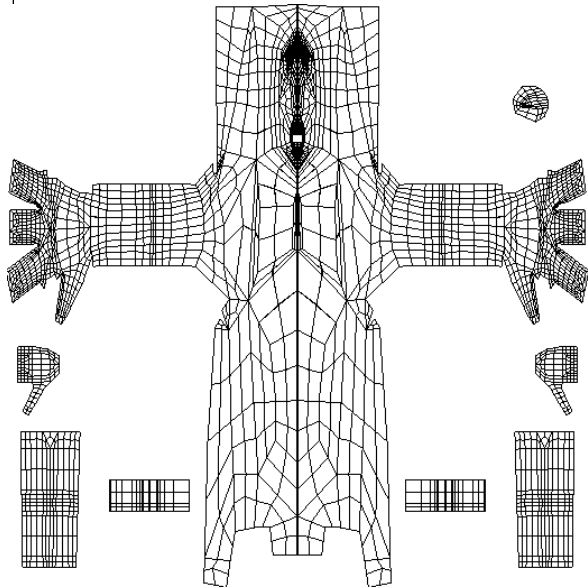
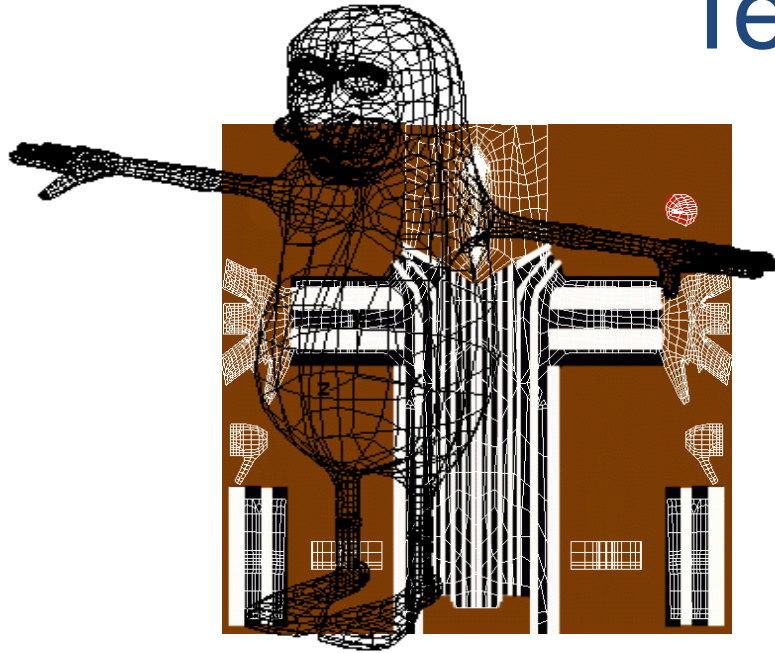
Téglalap közvetítő felület: textúratér nézet



Textúrázott űrhajó

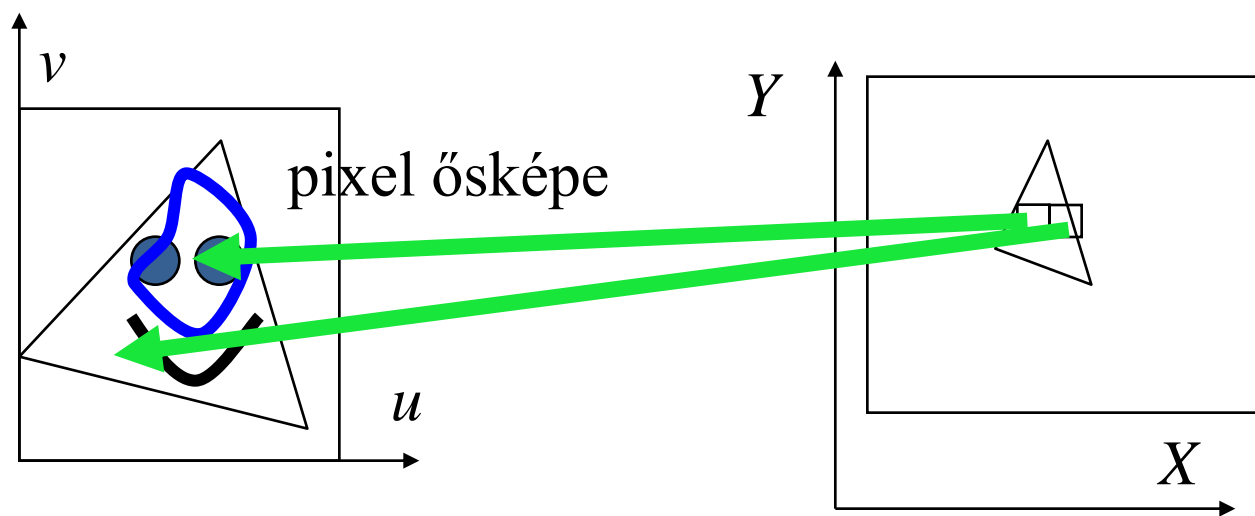


Textúrázás = kiterítés

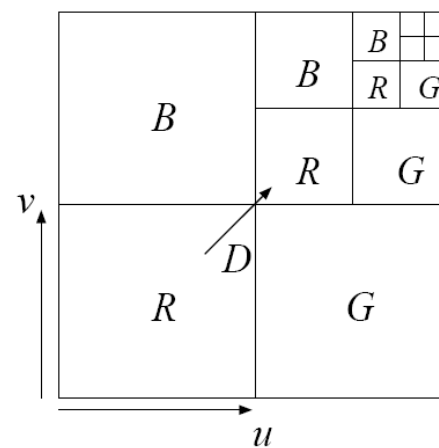
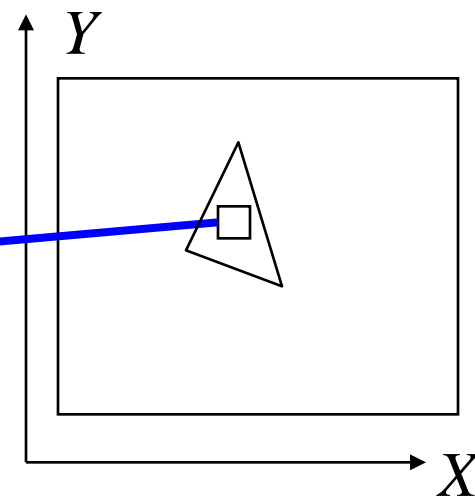
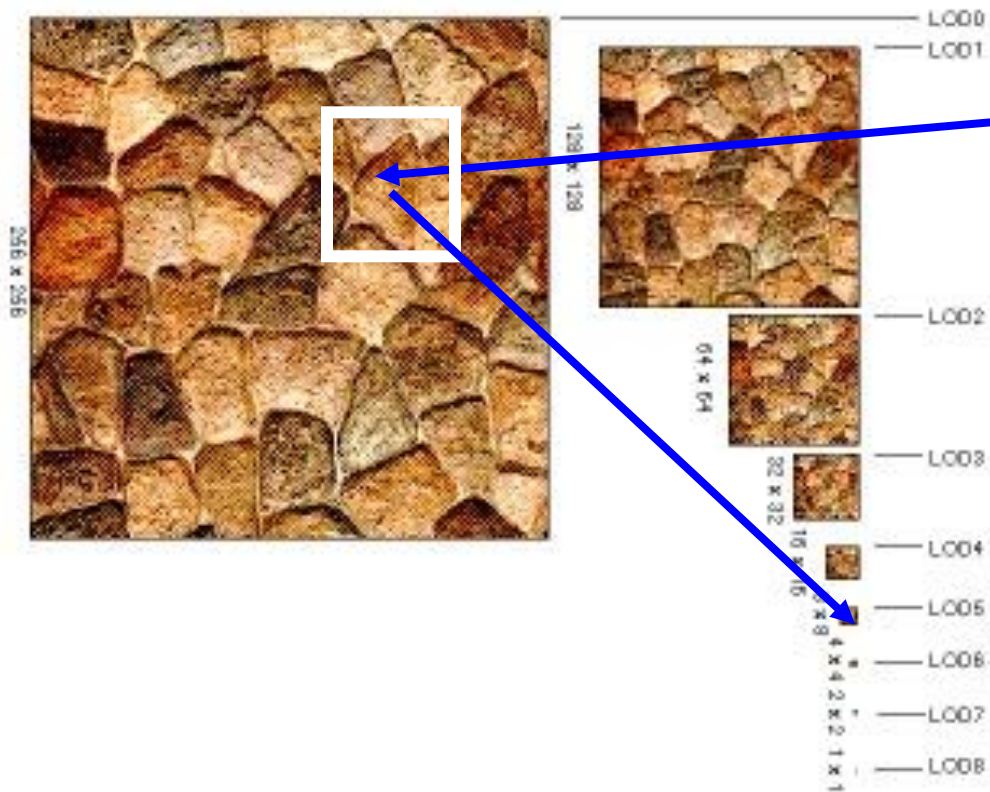


Torzításcsökkentés relaxációval

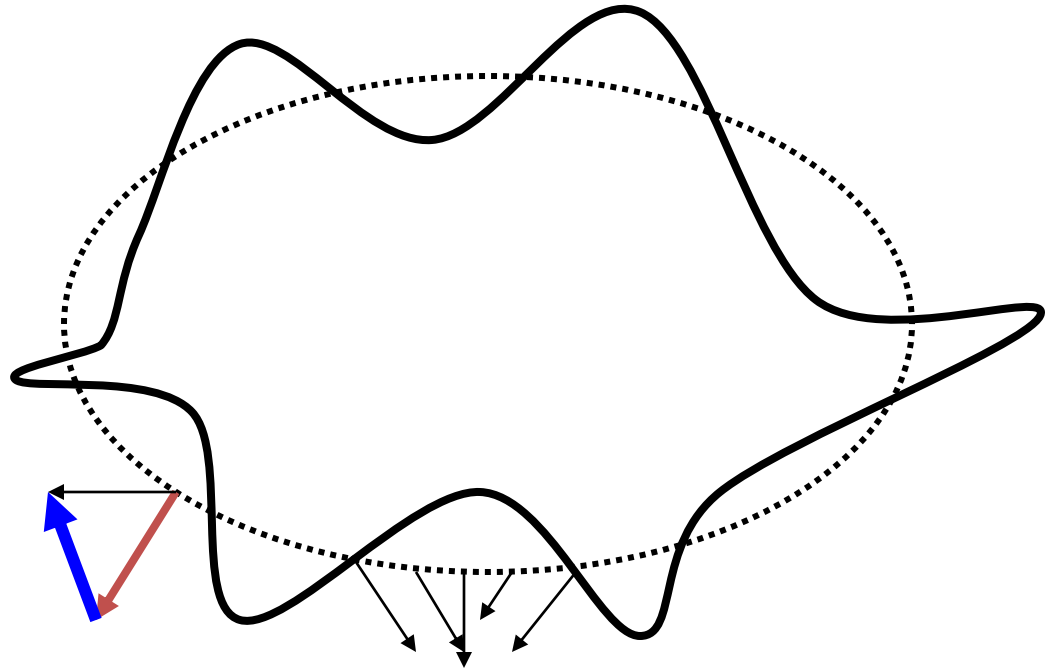
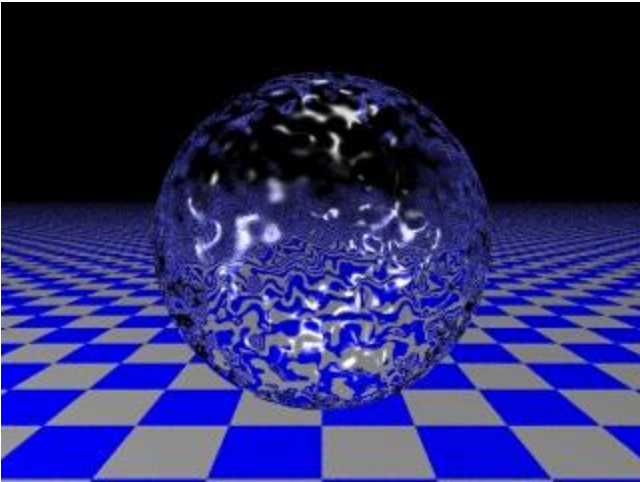
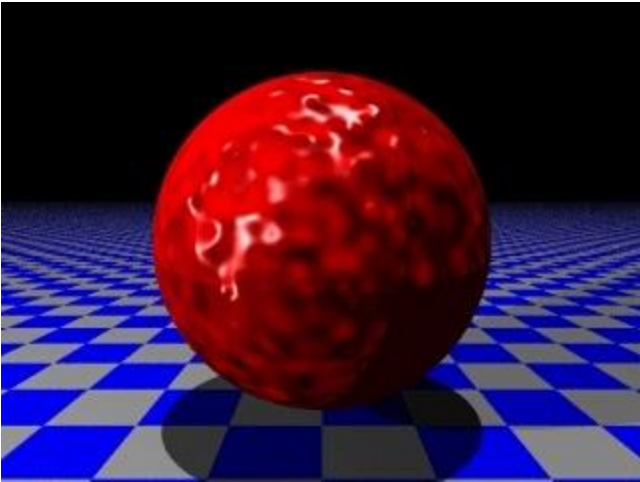
Textúrák szűrése



Mip-map adatstruktúra



Bucka leképzés (Bump mapping)



Tároljuk a normálvektort táblázatban
vagy a felületi elmozdulást

Textúrák az OpenGL-ben

A textúraobjektumok elnevezése

```
void glGenTextures (GLsizei  n,  
    GLuint *textureNames);
```

- n darab, jelenleg használaton kívüli textúraobjektum nevet a vissza a textureNames tömbbe. A visszaadott nevek nem feltétlenül egymást követő egészek

A textúraobjektumok létrehozása, használata

```
void glBindTexture (GLenum target,  
                    GLuint textureName);
```

Target: GL_TEXTURE_1D, GL_TEXTURE_2D, vagy GL_TEXTURE_3D,

- Ha először hívjuk a *textureName* azonosítóval: létrehoz egy textúraobjektumot és hozzákapcsolja a névhez
- Ha nem először hívjuk: *textureName* azonosítójú textúrát teszi kurrenssé

Textúra beállítása

```
void glTexParameter{if} (GLenum  
    target, GLenum pname, TYPE  
    param) ;
```

Target: GL_TEXTURE_1D, GL_TEXTURE_2D, vagy
GL_TEXTURE_3D,

Pname, param: számos paraméterezési lehetőség
(lásd OpenGL referencia)

Textúra beállítás - példák

- `glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);`
- `glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);`
 - Nagyításhoz ill kicsinyítéshez használt interpoláció (GL_NEAREST mellett lehet még GL_LINEAR)

Textúra beállítás - példák

- `glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);`
- `glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);`
- x és y (itt s és t) irányú ismétlődés (wrap=burkolat), lehet:
 - `GL_CLAMP` - rögzített
 - `GL_REPEAT` – ismétlődő parketta

Textúra beállítás - példák

void glTexImage2D(GLenum *target*, GLint *level*, GLint *internalFormat*, GLsizei *width*, GLsizei *height*, GLint *border*, GLenum *format*, GLenum *type*, const GLvoid **texels*);

- 2D struktúrát hoz létre.
- *Target*: GL_TEXTURE_2D (vagy más)
- *level* > 0 ha több felbontást használunk
- *internalFormat*: használt színt komponensek pl GL_RGBA
- GLsizei *width*, GLsizei *height*: a textúraobj mérete
- *Border*: határ szélessége (lehet 0)
- *Format*, *type*: a textúraadatok formátuma pl format=GL_RGBA, type= GL_UNSIGNED_BYTE
- *texels*: a textúra-minta adatait tartalmazó tömb

Minta textúra inicializálása

```
static GLubyte  textDescriptor[height][width][4];  
static GLuint texName[2];
```

```
...
```

```
glGenTextures(1, texName);  
glBindTexture(GL_TEXTURE_2D, texName[0]);  
glTexParameteri(...);
```

```
...
```

```
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA,  
width, height, 0, GL_RGBA,  
GL_UNSIGNED_BYTE, textDescriptor);
```

Textúra paraméterezése

- **void glTexCoord{1234}{sifd}(*TYPE coords*);**
 - Textúra koordinátái: s,t,r,q ($\sim x,y,z,w$), ahol q=1 ált.
 - A kurrens (s,t,r,q) textúrákoordinátákat állítja be.
Az ezt követően létrehozott csúcspontokhoz a rendszer ezt a textúrákoordinátát rendeli

Textúra paraméterezése, példa

```
glBegin(GL_QUADS);  
    glTexCoord2f(0, 0); glVertex3d(1,-1,0);  
    glTexCoord2f(0, 1); glVertex3d(-1,-1,0);  
    glTexCoord2f(1, 1); glVertex3d(-1,1,0);  
    glTexCoord2f(1, 0); glVertex3d(1,1,0);  
glEnd();
```

Textúra és saját szín kombinálása

```
glTexEnvf(GL_TEXTURE_ENV,  
GL_TEXTURE_ENV_MODE, GL_REPLACE);
```

- GL_TEXTURE_ENV_MODE lehet:
 - GL_REPLACE – a textúrával felülírjuk a saját v. megvilágításból adódó színt
 - GL_MODULATE – a megjelenített színt a textúrából és a saját színből közösen számítjuk