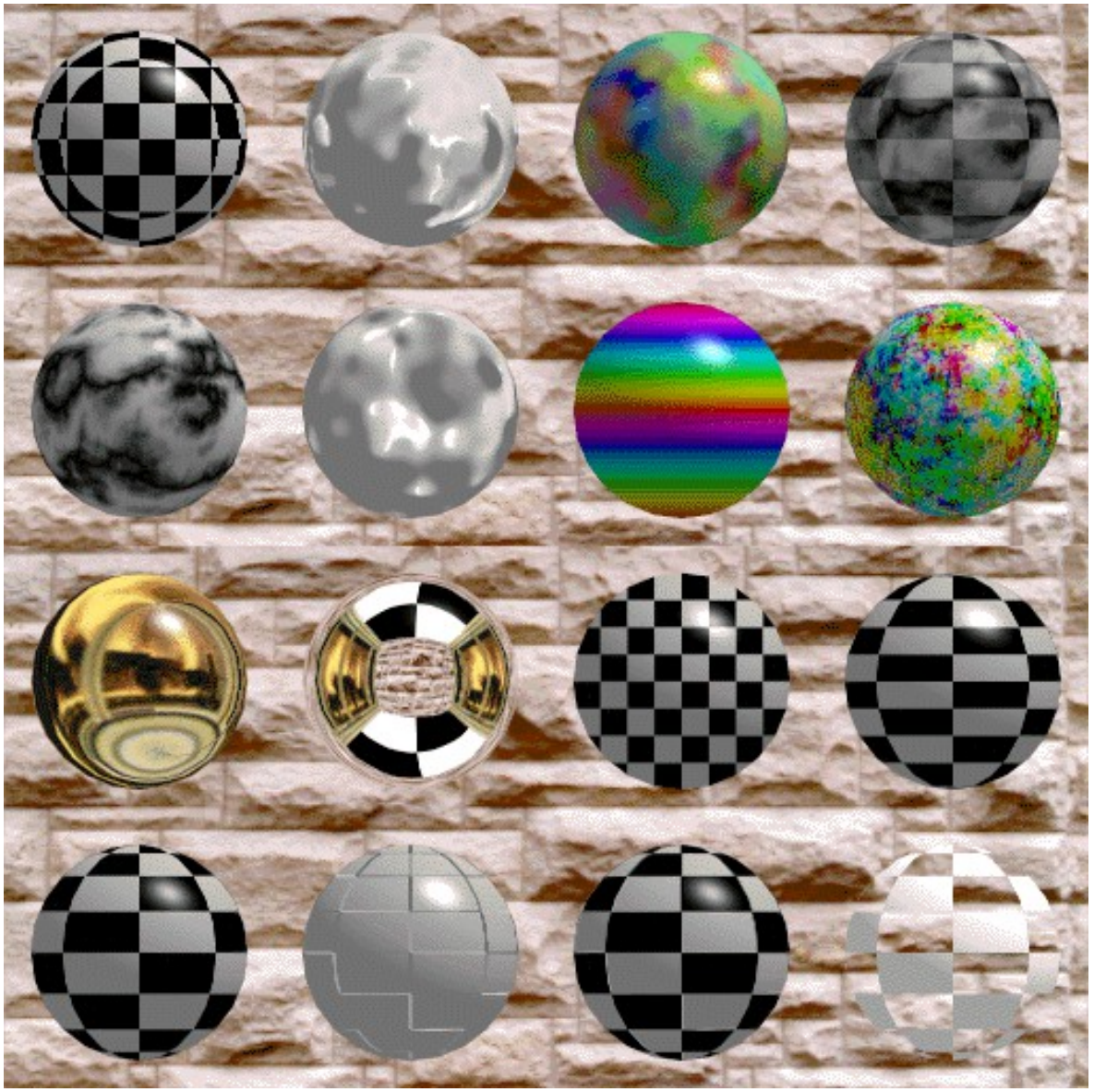
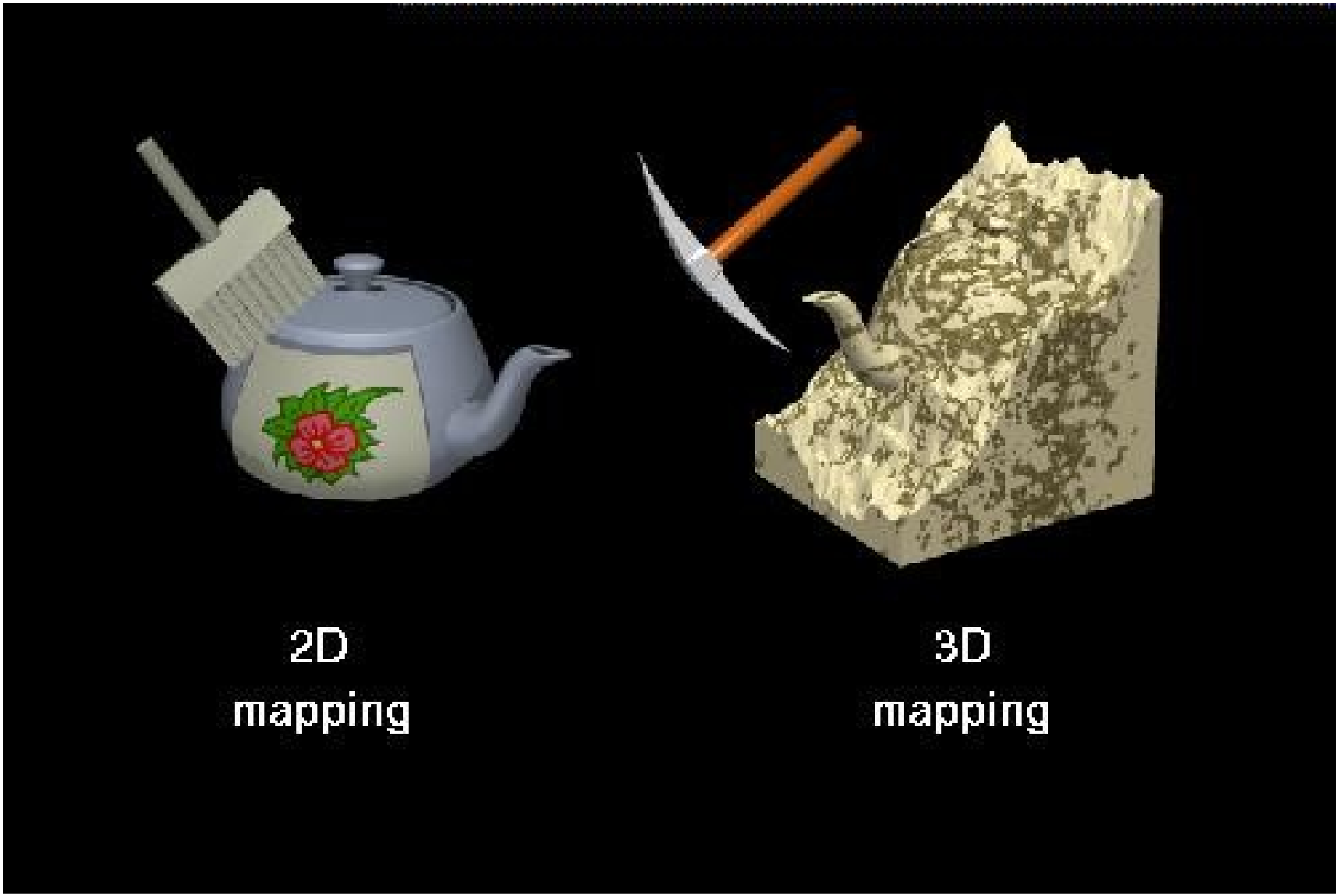
Teksture

Lastnosti snovi

O teksturah

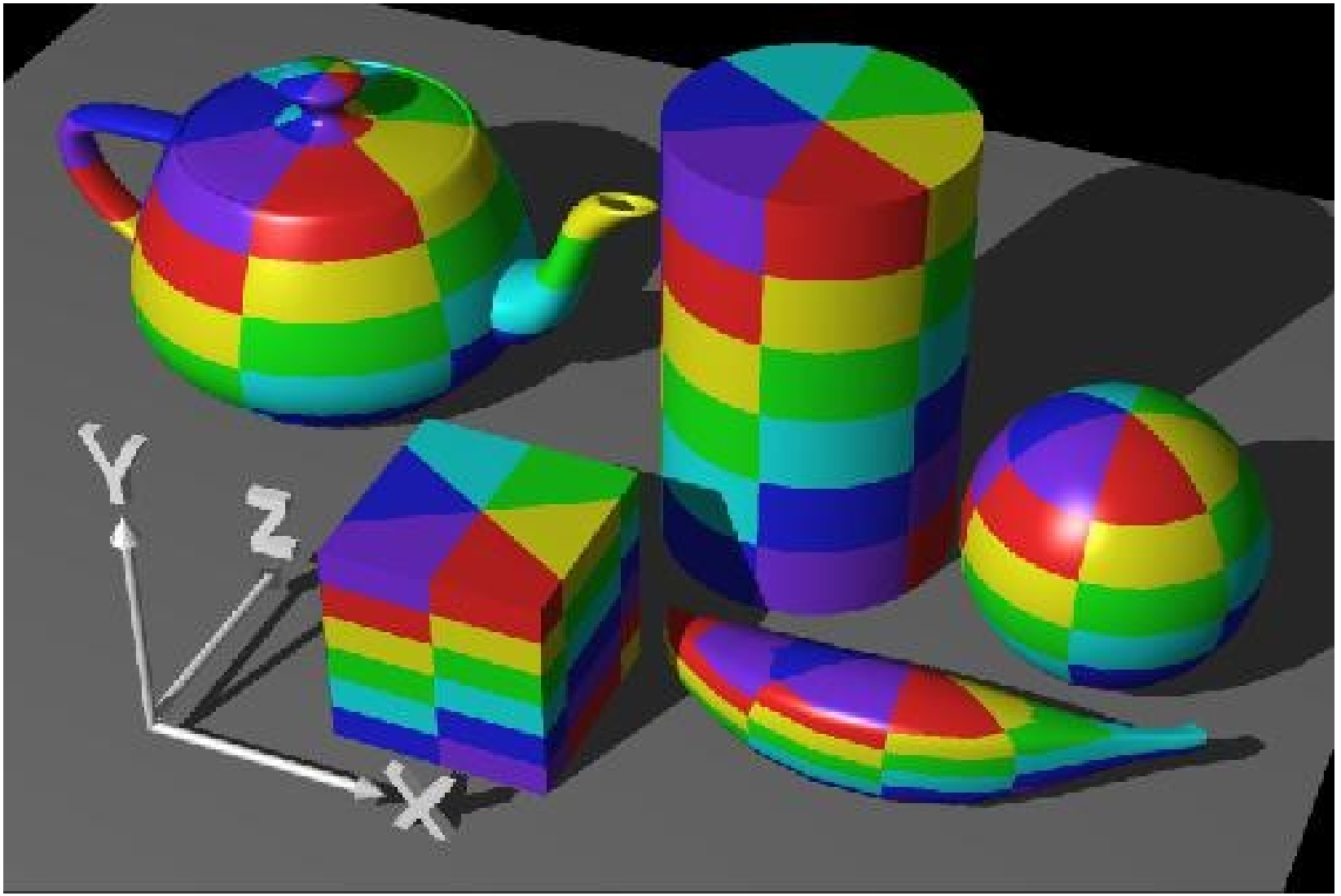


2D-3D lepljenje tekstur

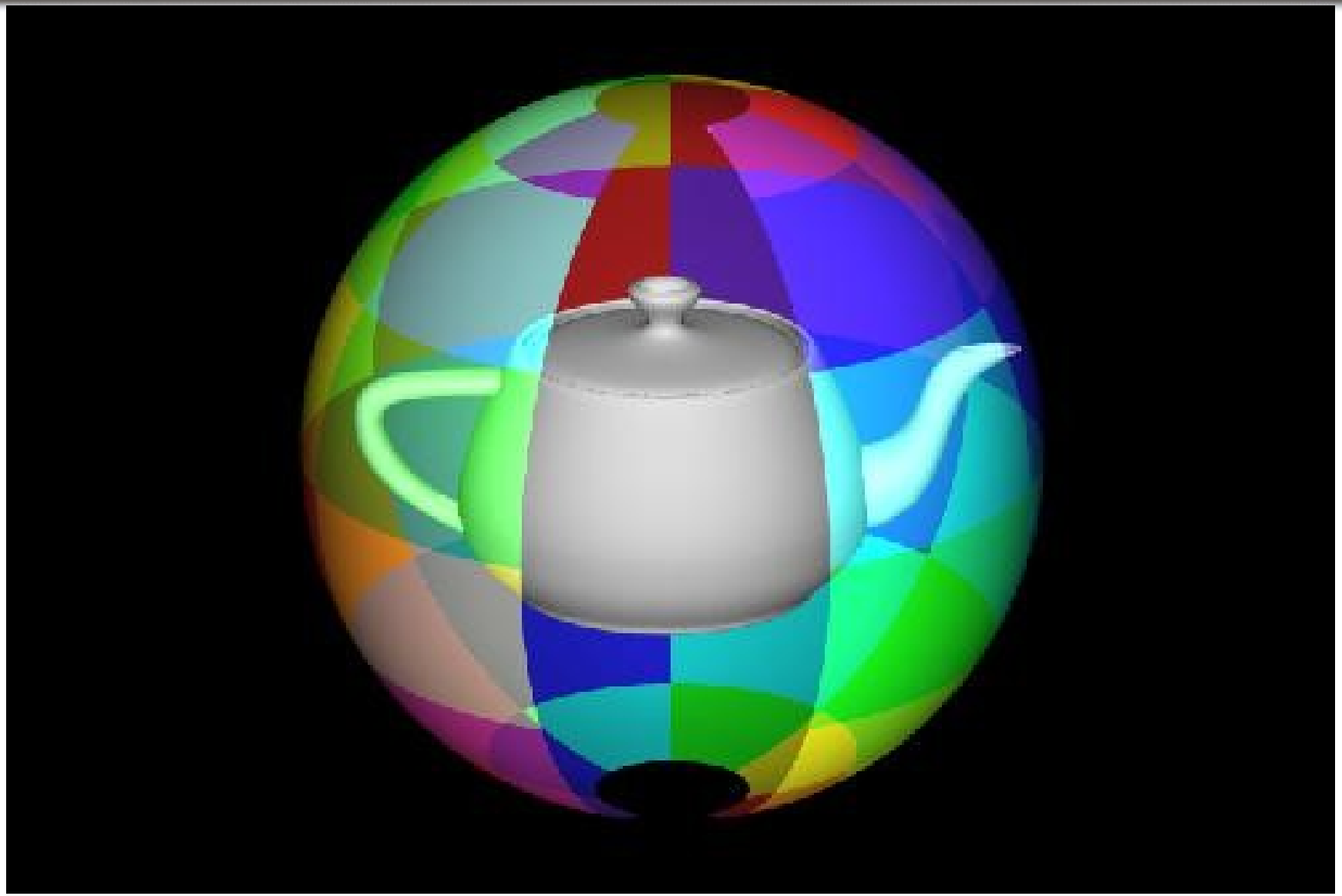


2D cilindricno lepljenje

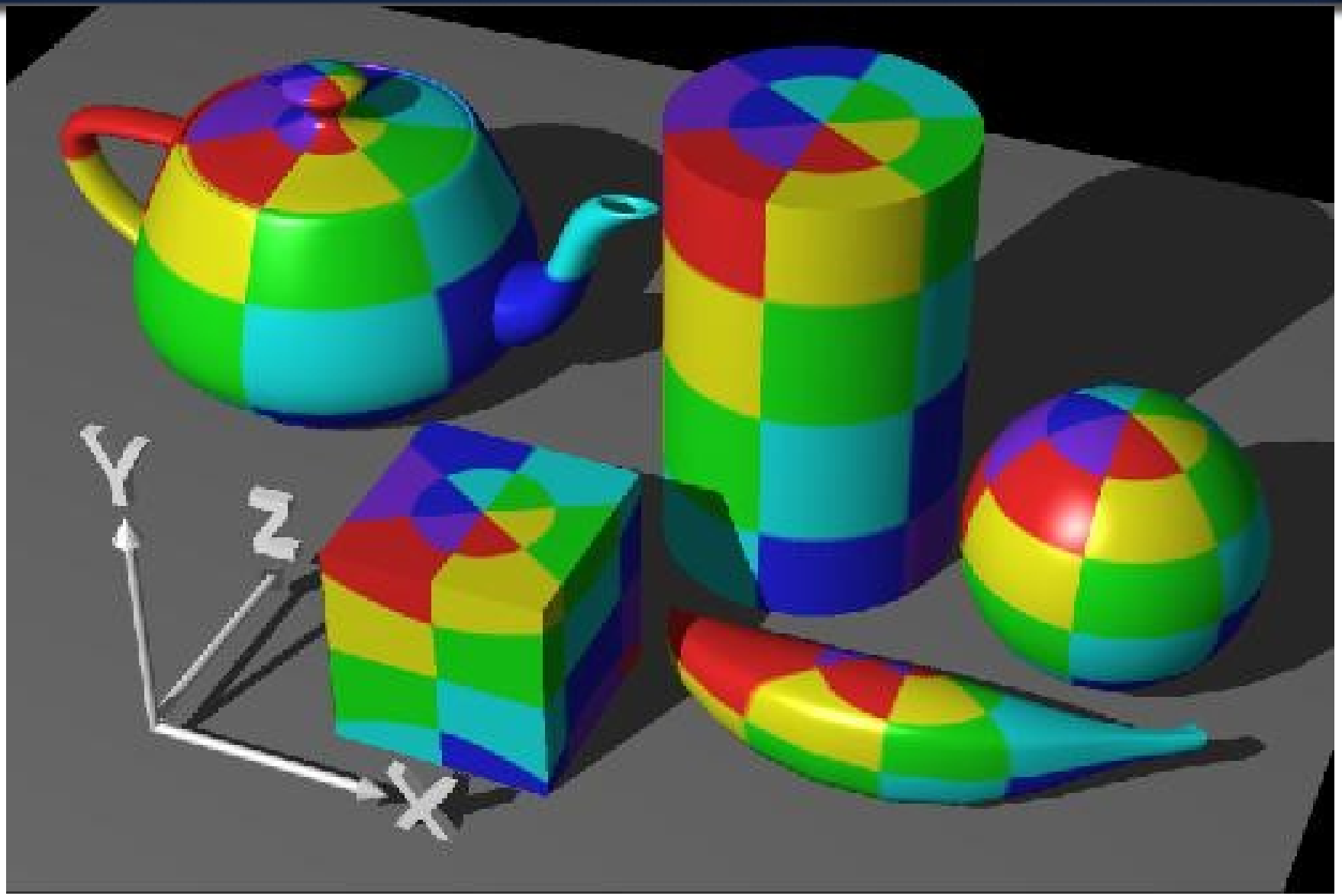
Primeri 2d cilindricnega lep.



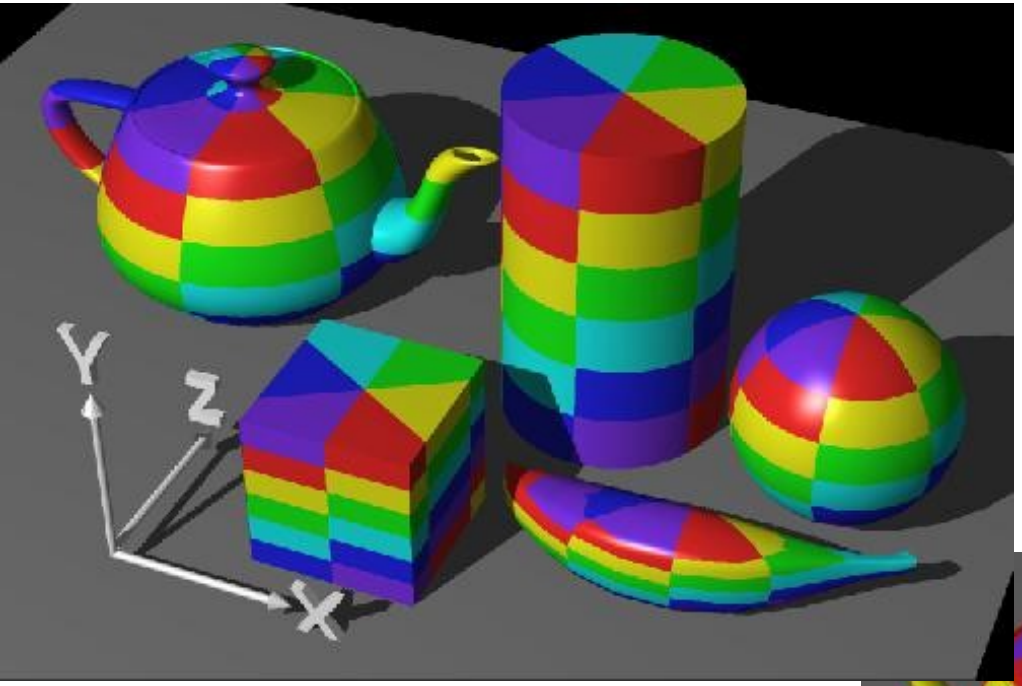
Sferično lepljenje

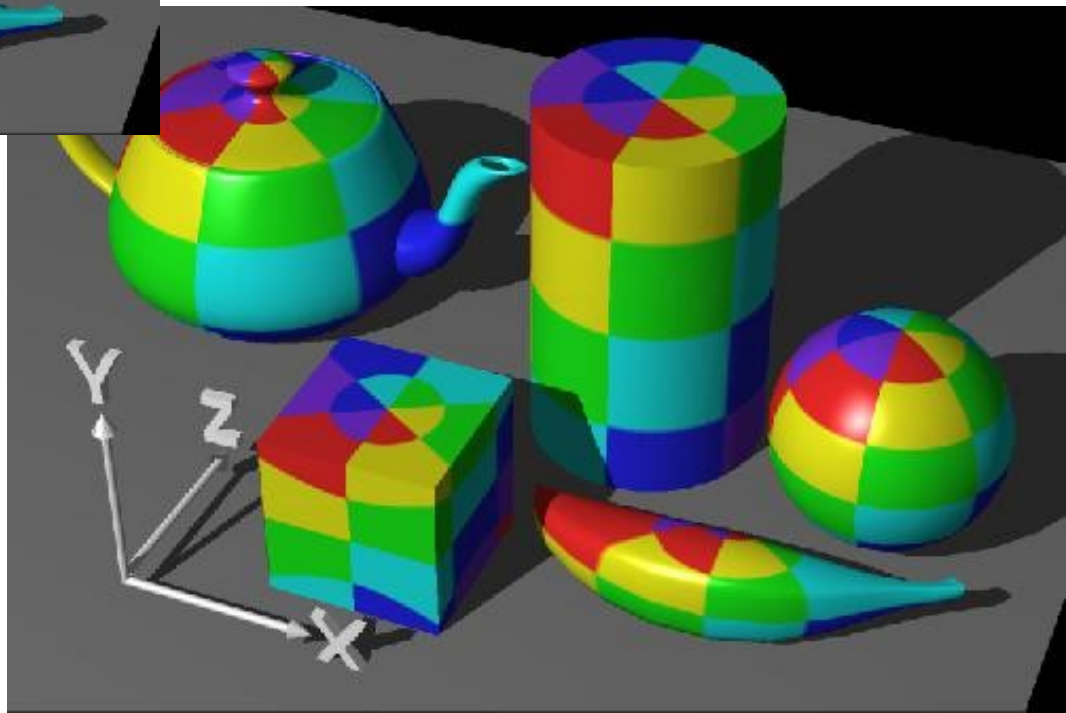


Primeri sferičnega lepljenja

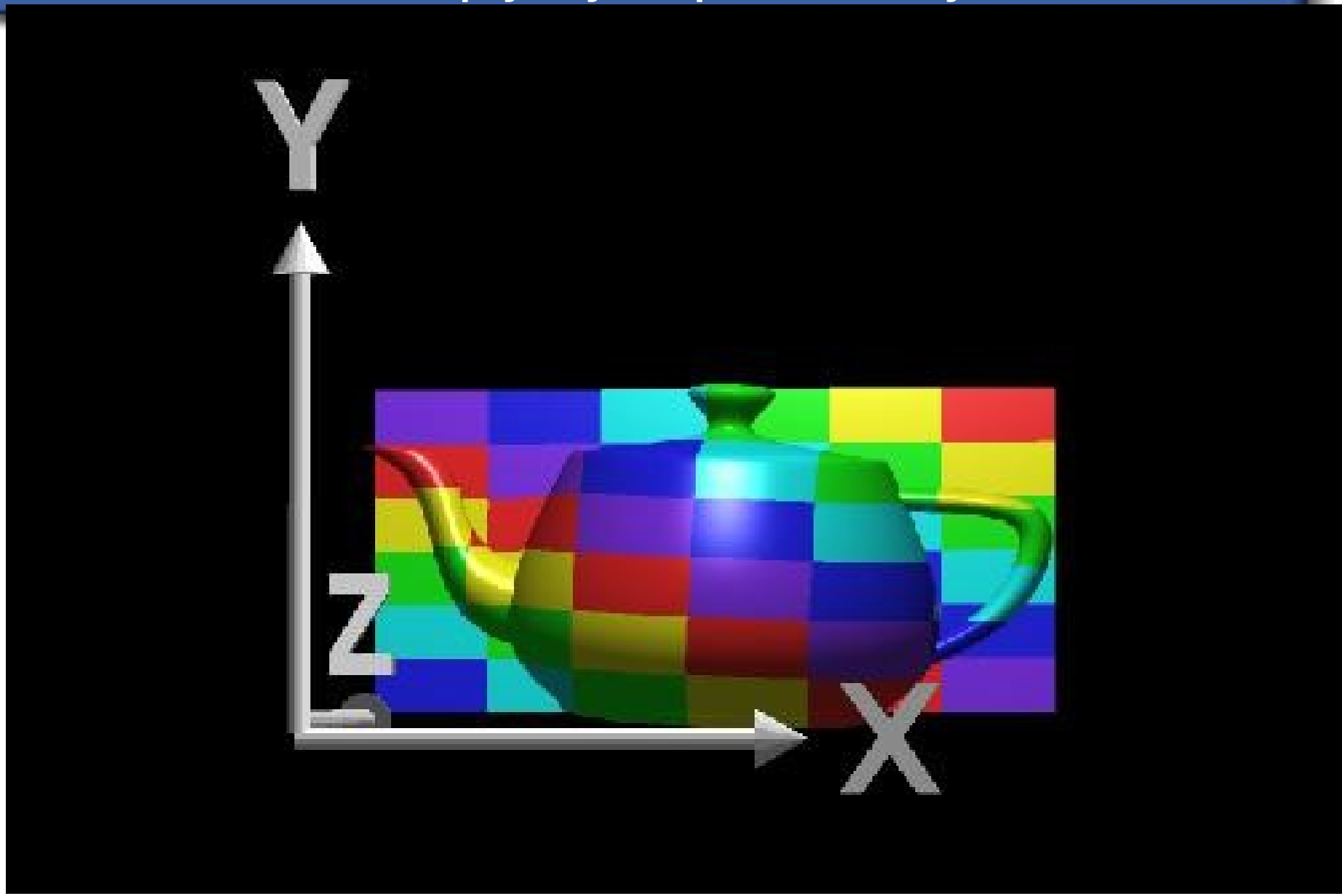


Primerjava cil-sfer

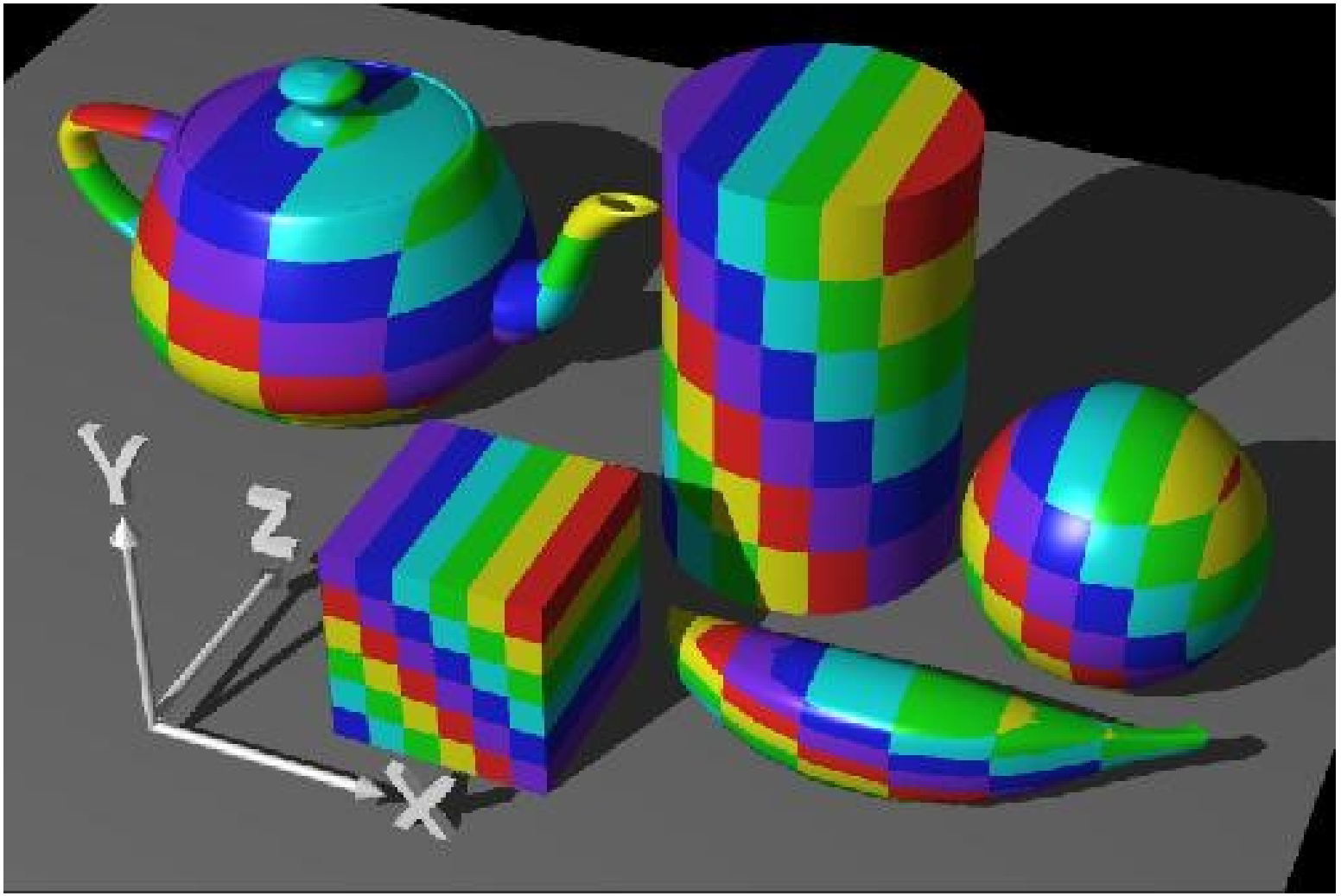




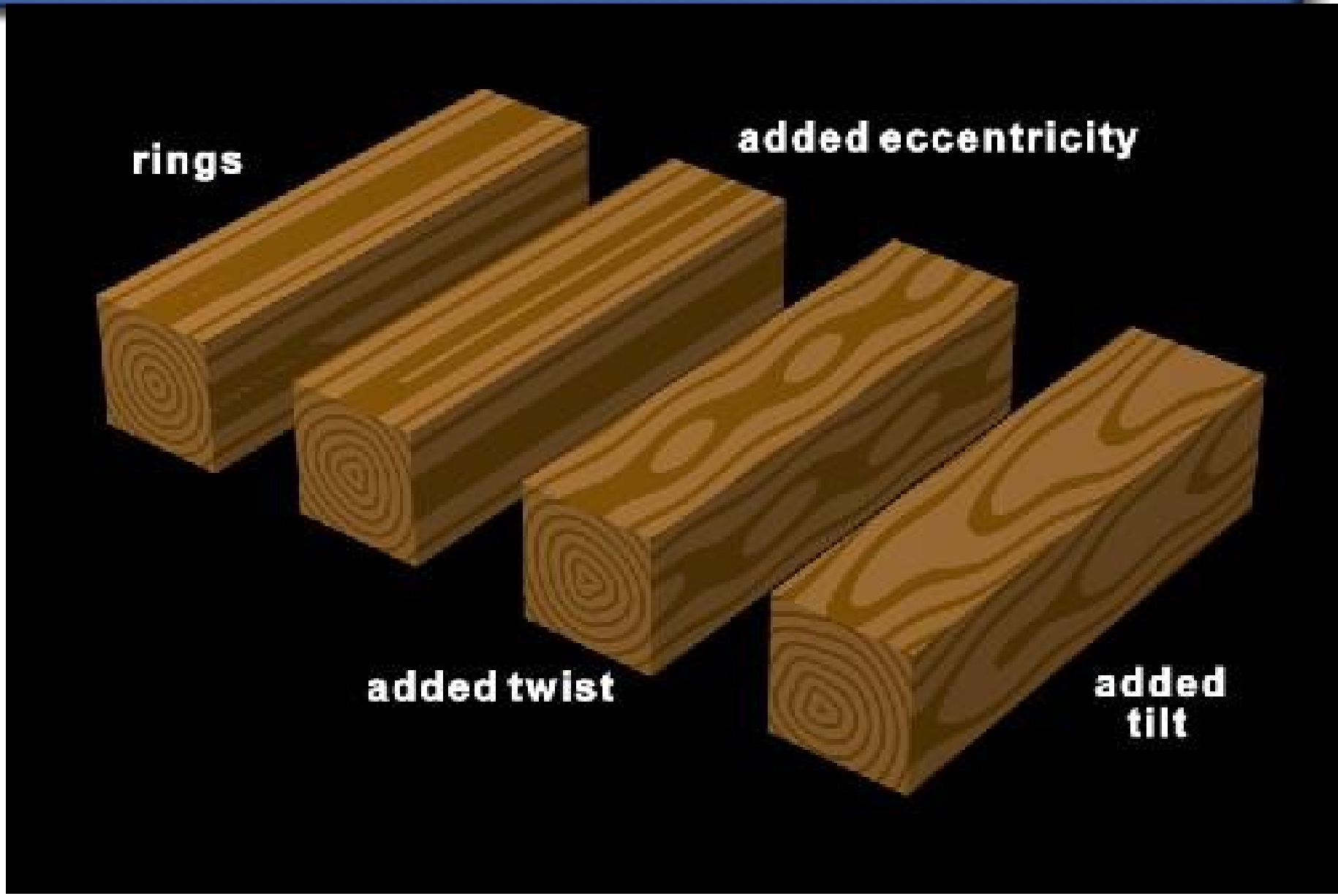
3D lepljenje - prosevanje



Primeri 3D lepljenja

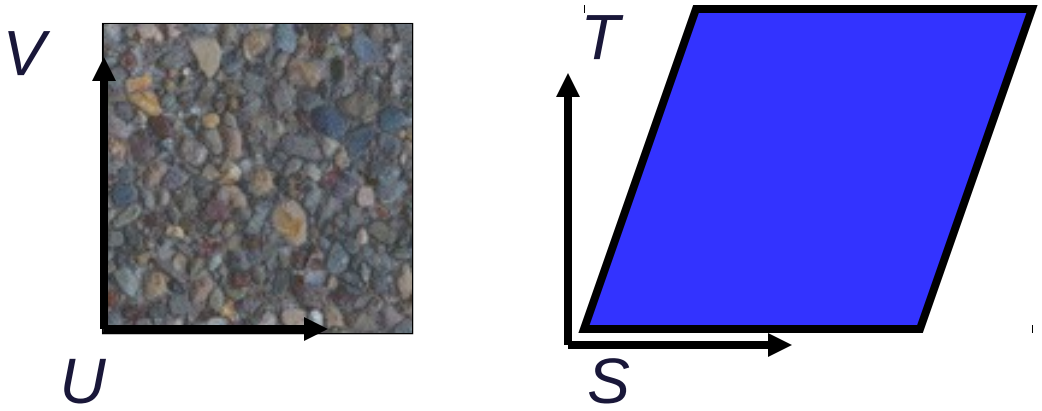


3D lepljenje, variacije, les



Lepljenje tekstur (Texture Mapping)

- Texture map is an image, two-dimensional array of color values (texels)
- Texels are specified by texture's (u,v) space
- At each screen pixel, texel can be used to substitute a polygon's surface property
- We must map (u,v) space to polygon's (s, t) space

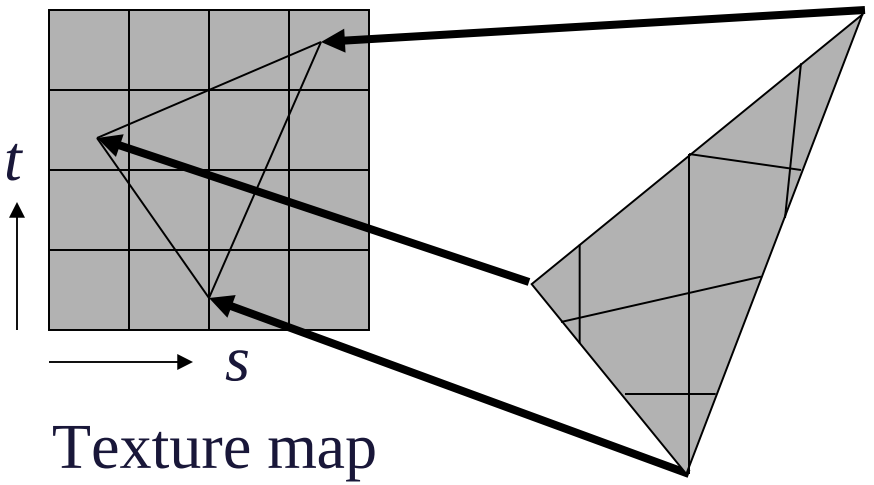


Lepljenje tekstur

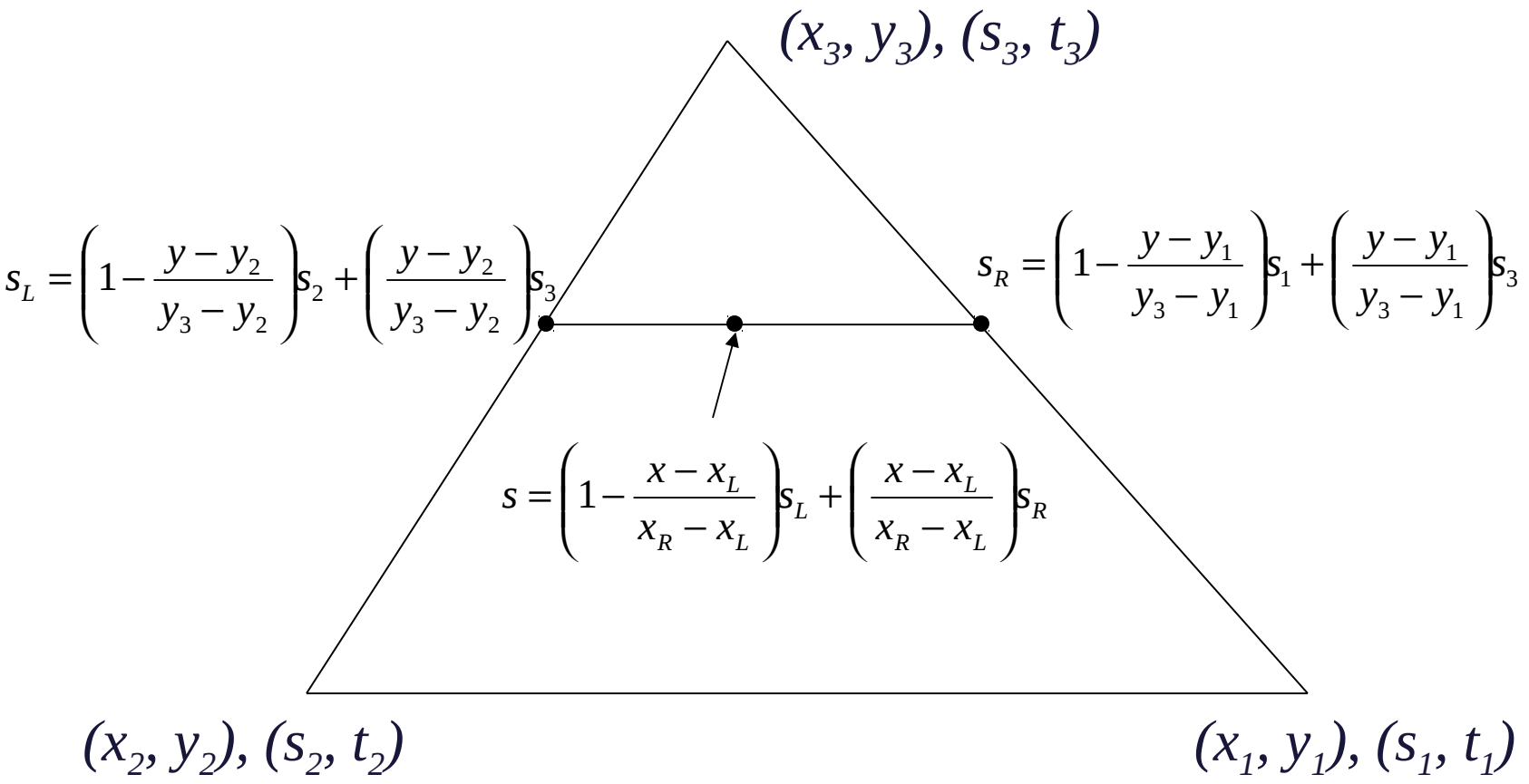
- (u,v) to (s,t) mapping can be explicitly set at vertices by storing texture coordinates with each vertex
- How do we compute (u,v) to (s,t) mapping for points in between
  - Watch for aliasing
  - Watch for many to one mappings
  - Watch for perspective foreshortening effects and linear interpolation

Interpolacija tekstur

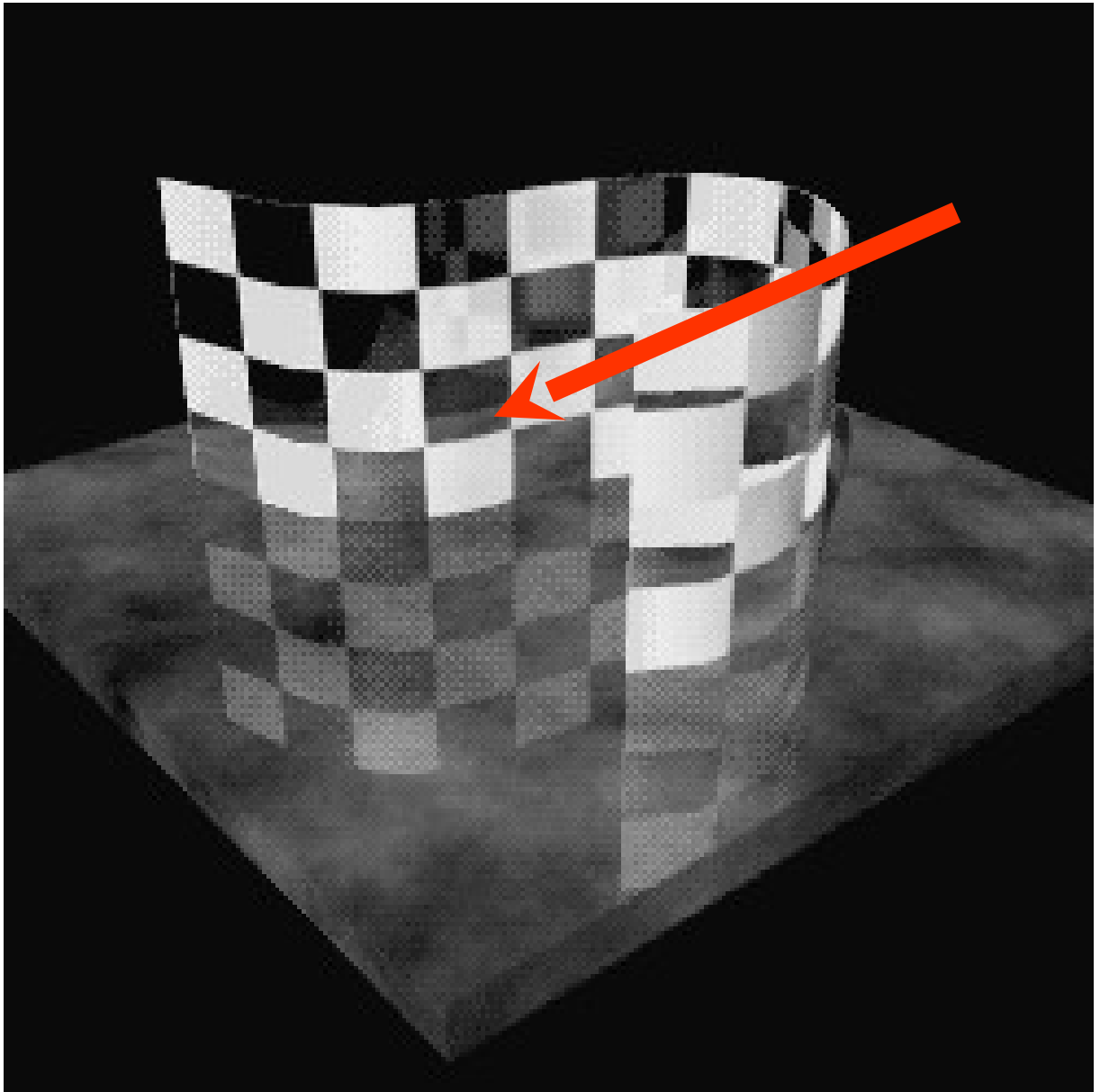
- Specify where the vertices in world space are mapped to in texture space
- Linearly interpolate the mapping for other points in world space
  - Straight lines in world space go to straight lines in texture space



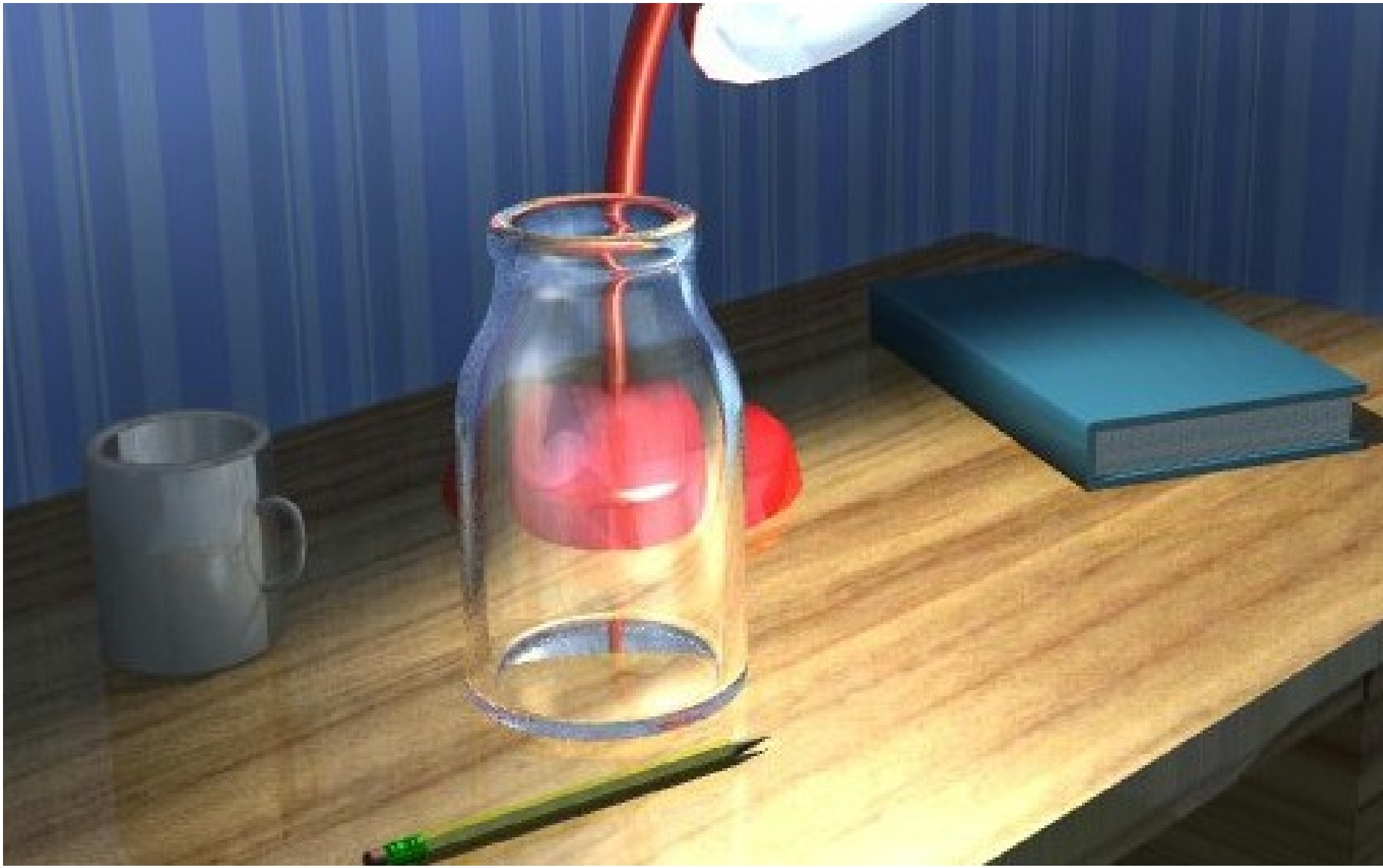
Interpolacija koordinat



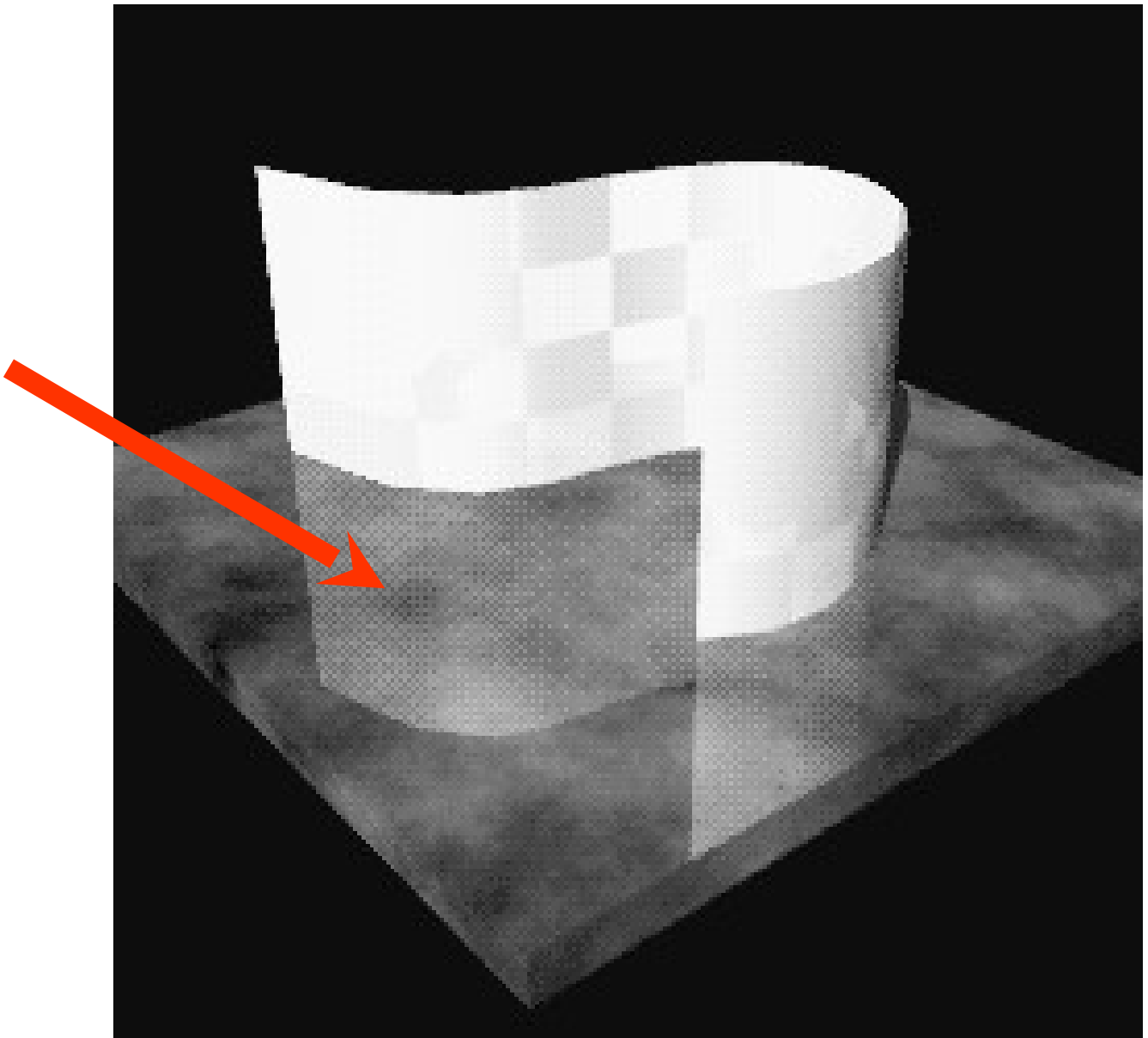
Tekstura prosojnosti



Prosojnost - refrakcija

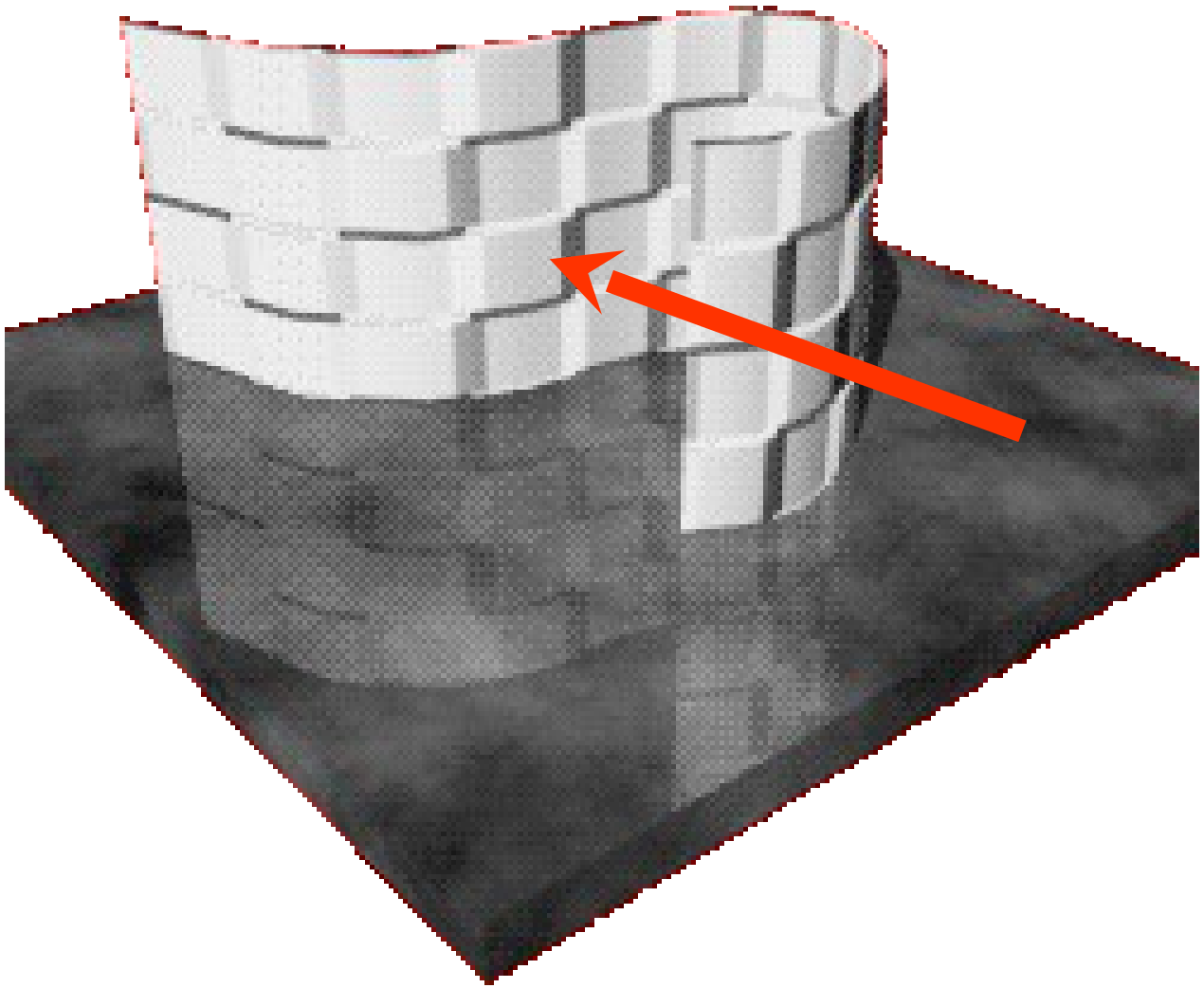


Tekstura odbojnosti

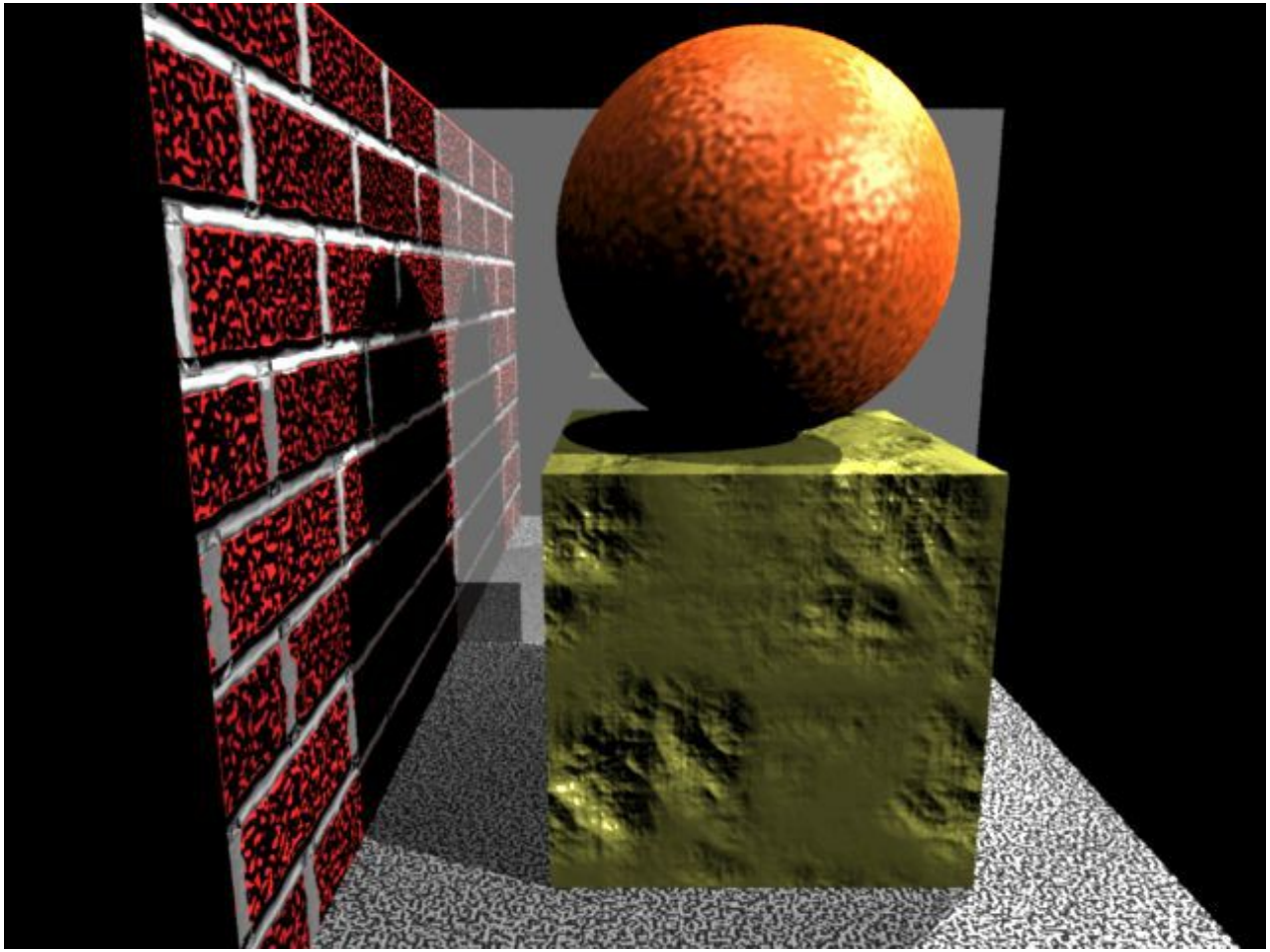


Reflektivnost

Tekstura izbočenja



Lepljenje izboklin (bump mapping)

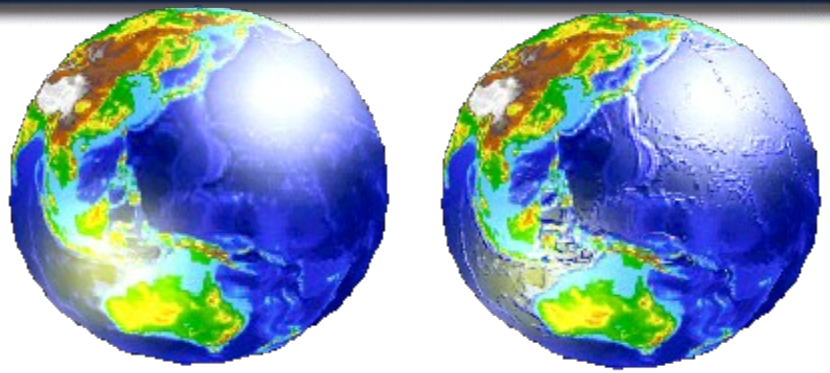


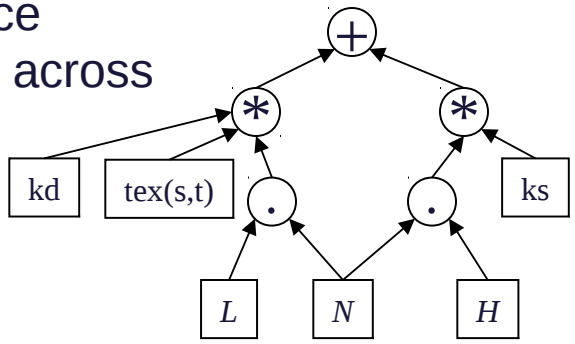
Lepljenje izboklin (bump mapping)

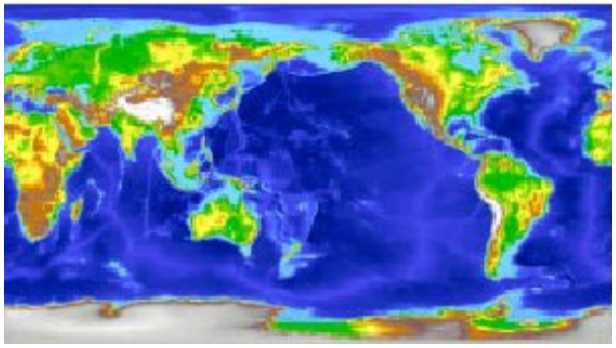
- Use textures to modify surface geometry
- Use texel values to modify surface normals of polygon
- Texel values correspond to height field
  - Height field models a rough surface
- Partial derivative of bump map specifies change to surface normal

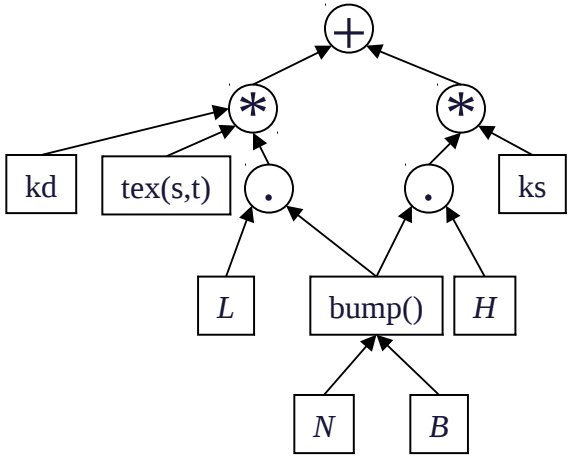
Primerjava tekstur in izboklin

- Texture mapping simulates detail with a color that varies across a surface
- Bump mapping simulates detail with a surface normal that varies across a surface









Primerjava lepljenja odmika in izboklin (Displacement v. Bump

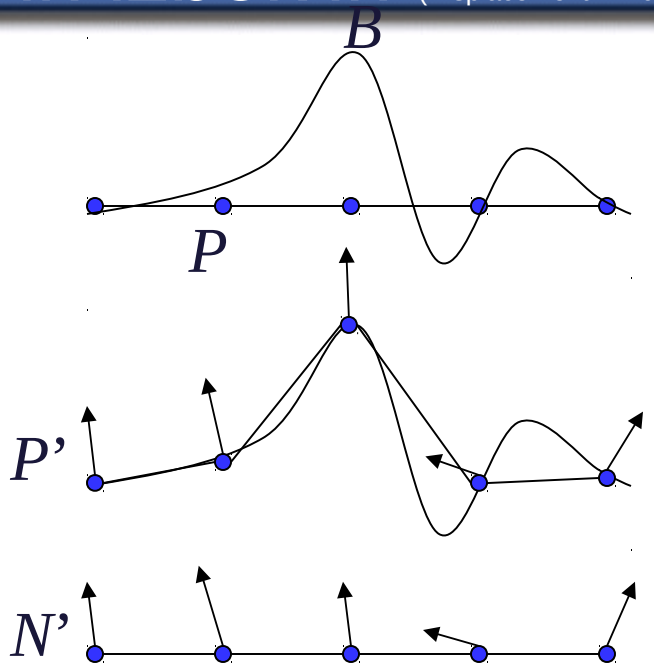
- Displacement mapping
  - Changes locations of vertices
  - Requires mesh to be subdivided to shader resolution

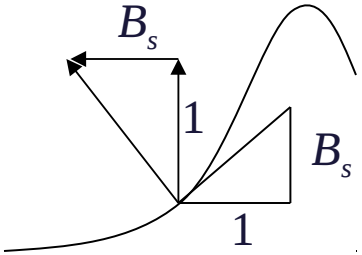
$$P' = P + B\,N$$

- Bump mapping
  - Changes only direction of surface normal
  - Give illusion of displacement map

$$N' = N + B_s(N \times P_t) - B_t(N \times P_s)$$

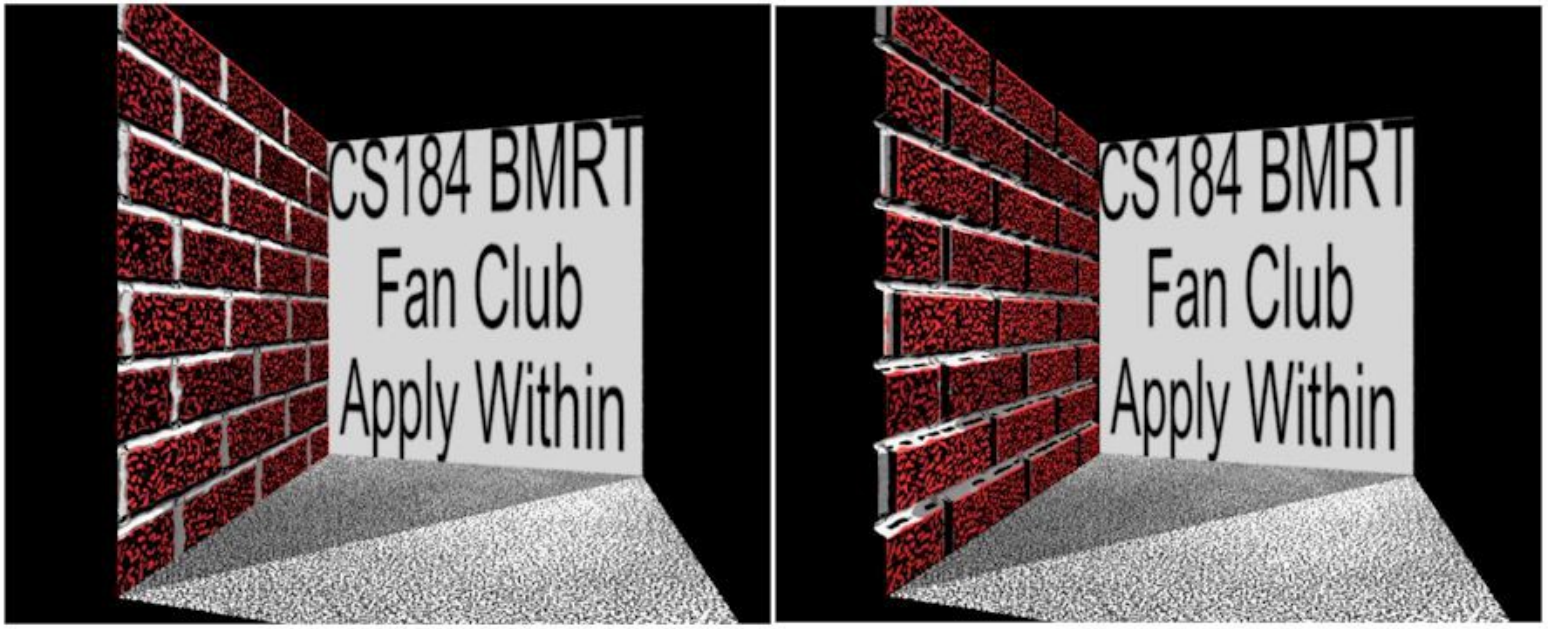
- Silhouettes are a dead giveaway
  - Displacements alter silhouettes
  - Bump maps leave silhouette smooth





Lepljenje odmikov (Displacement Mapping)

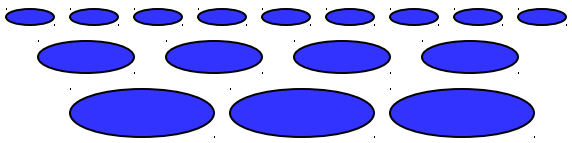
- Bump mapped normals are inconsistent with actual geometry. Problems arise (shadows).
- Displacement mapping actually affects the surface geometry



Celularne teksture

Teksture

- Snov bo izgledala bolj zanimivo
  - Barva
  - prosojnost (sir, železo)
  - Umazanija, rja
- Nudi dodatno globino človeškemu pogledu



Preslikava tekstur

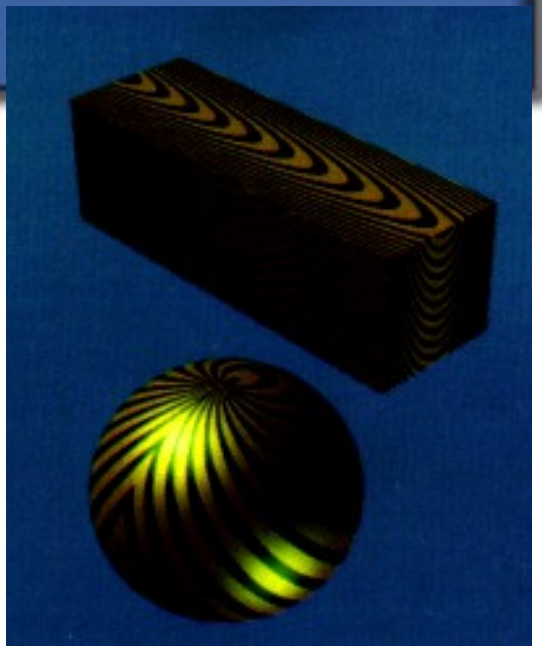
- Maps image onto surface
- Depends on a surface parameterization (*s*,*t*)
  - Difficult for surfaces with many features
  - May include distortion
  - Not necessarily 1:1

3D Teksture

- Uses 3-D texture coordinates (*s*,*t*,*r*)
- Can let *s* = *x*, *t* = *y* and *r* = *z*
- No need to parameterize surface
- No worries about distortion
- Objects appear sculpted out of solid substance

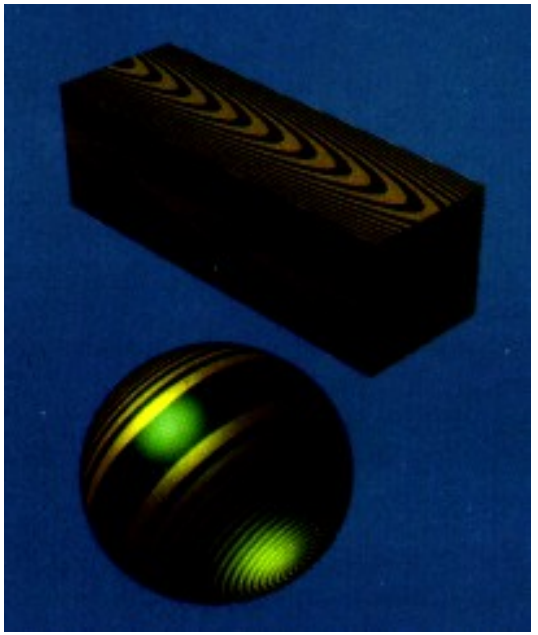
Surface Texture

features don't line up



Solid Texture

features do line up

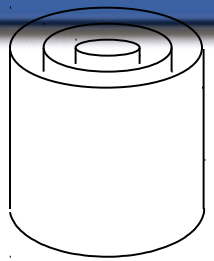


Darwyn Peachey, 1985

Problemi s 3D teksturami

- How can we deform an object without making it swim through texture?
- How can we efficiently store a procedural texture?

Proceduralne teksture



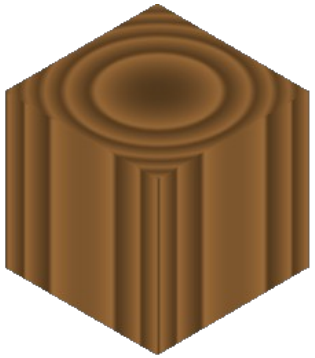
Woodmap(f)

- Texture map is a function
- Write a procedure to perform the function
  - input: texture coordinates - s,t,r
  - output: color, opacity, shading
- Example: Wood
  - Classification of texture space into cylindrical shells

$$f(s,t,r) = s^2 + t^2$$

  - Outer rings closer together, which simulates the growth rate of real trees
  - Wood colored color table
    - Woodmap(0) = brown “earlywood”
    - Woodmap(1) = tan “latewood”

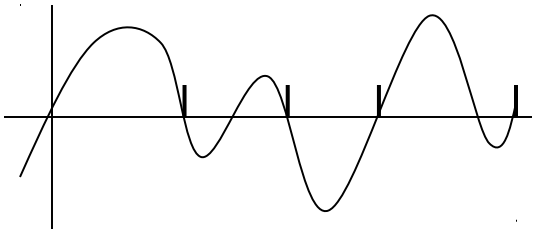
$f(s,t,r) = s^2 + t^2$



Wood(s,t,r)

Wood(s,t,r) = Woodmap(f(s,t,r) mod 1)

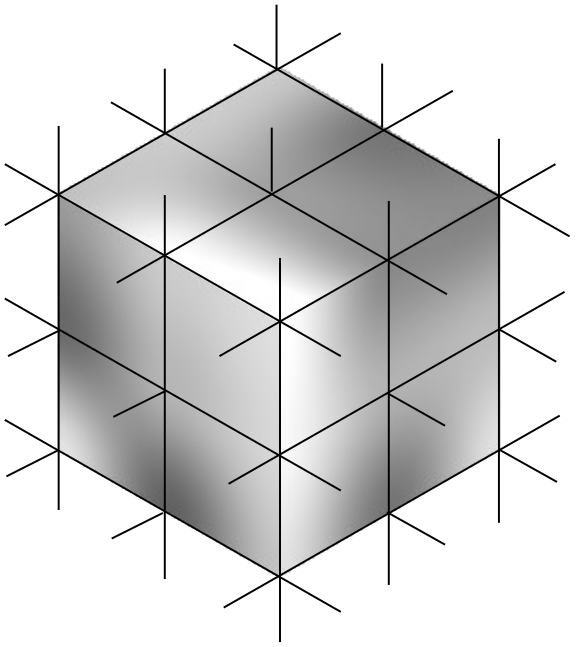
Šumne funkcije



- Add “noise” to make textures interesting
- Perlin noise function $N(x,y,z)$
  - Smooth
  - Correlated
  - Bandlimited
- $N(x,y,z)$ returns a single random number in [-1,1]
- Gradient noise
  - Like a random sine wave

    $N(x,y,z)=0$ for int x,y,z
- Value noise
  - Also like a random sine wave

    $N(x,y,z)$=random for int x,y,z



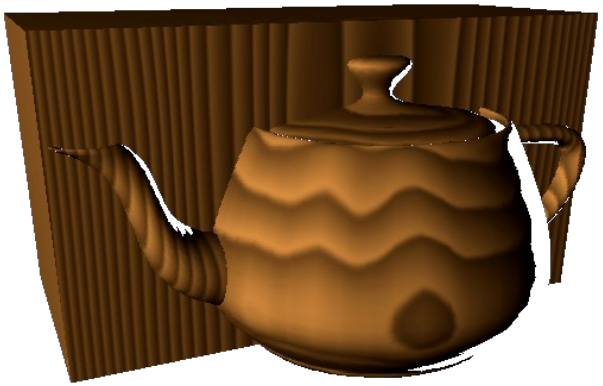
Uporaba šuma

- Add noise to cylinders to warp wood
  - Wood($s^2 + t^2 + N(s,t,r)$)
- Controls
  - Amplitude: power of noise effect

    $a\,N(s, t, r)$
  - Frequency: coarse v. fine detail

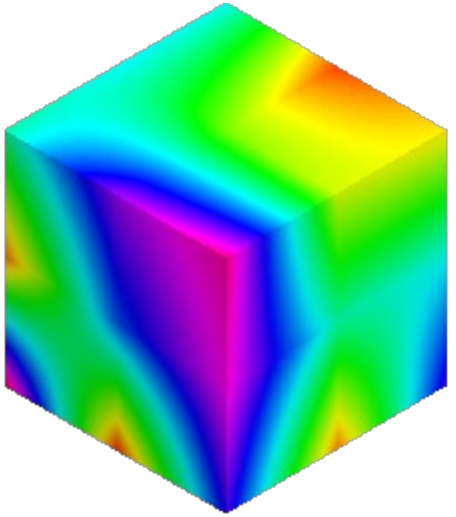
    $N(f_s\,s, f_t\,t, f_r\,r)$
  - Phase: location of noise peaks

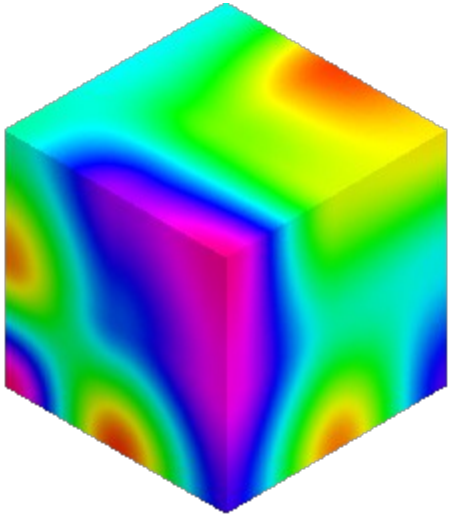
    $N(s + \phi_s, t + \phi_t, r + \phi_r)$



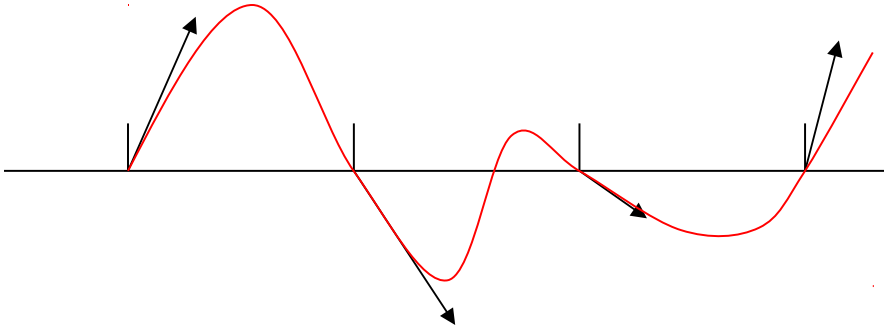
Kako naredimo šum

- Good:
  - Create 3-D array of random values
  - Trilinearly interpolate
- Better
  - Create 3-D array of random 3-vectors
  - Hermite interpolate

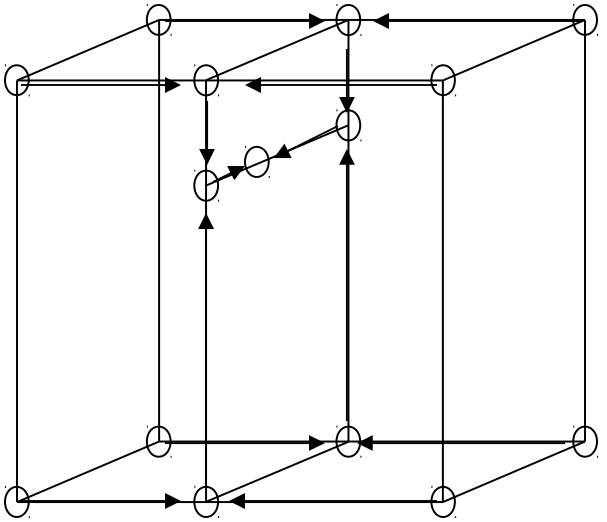




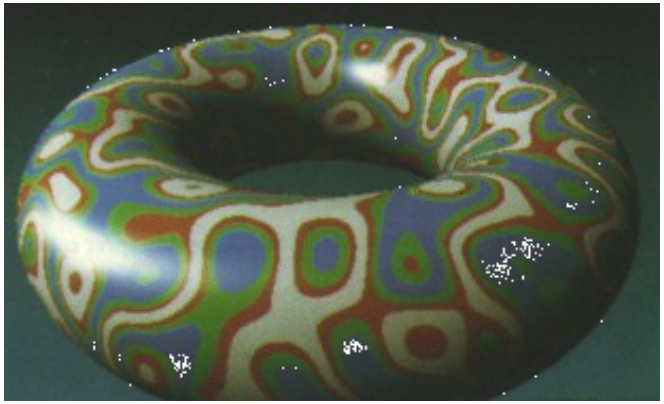
Hermitova interpolacija



- Some cubic $h(t) = at^3 + bt^2 + ct + d$ s.t.
  - $h(0) = 0 \quad (d = 0)$
  - $h(1) = 0 \quad (a + b + c = 0)$
  - $h'(0) = r_0 \quad (c = r_0)$
  - $h'(1) = r_1 \quad (3a + 2b + r_0 = r_1)$
- Answer:
  - $h(t) = (r_0 + r_1)\, t^3 - (2r_0 + r_1)\, t^2 + r_0 t$
- Tricubic interpolation
  - Interpolate corners along edges
  - Interpolate edges into faces
  - Interpolate faces into interior

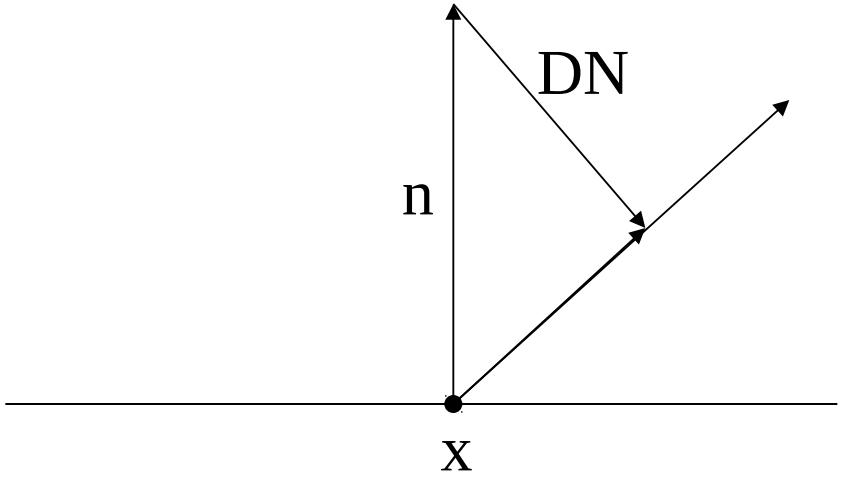


Pobarvani obroči



- Spotted donut
  - Gray(N(40*x,40*y,40*z))
  - Gray() - ramp colormap
  - Single 40Hz frequency
- Bozo donut
  - Bozo(N(4*x,4*y,4*z))
  - Bozo() - banded colormap
  - Cubic interpolation means contours are smooth

Obroči s preslikavo izboklin



n += DNoise(x,y,z); normalize(n);

- DNoise(*s*,*t*,*r*) = ∇Noise(*s*,*t*,*r*)
- Bumpy donut
  - Same procedural texture as spotted donut
  - Noise replaced with DNoise

Sestavljene teksture na obročih

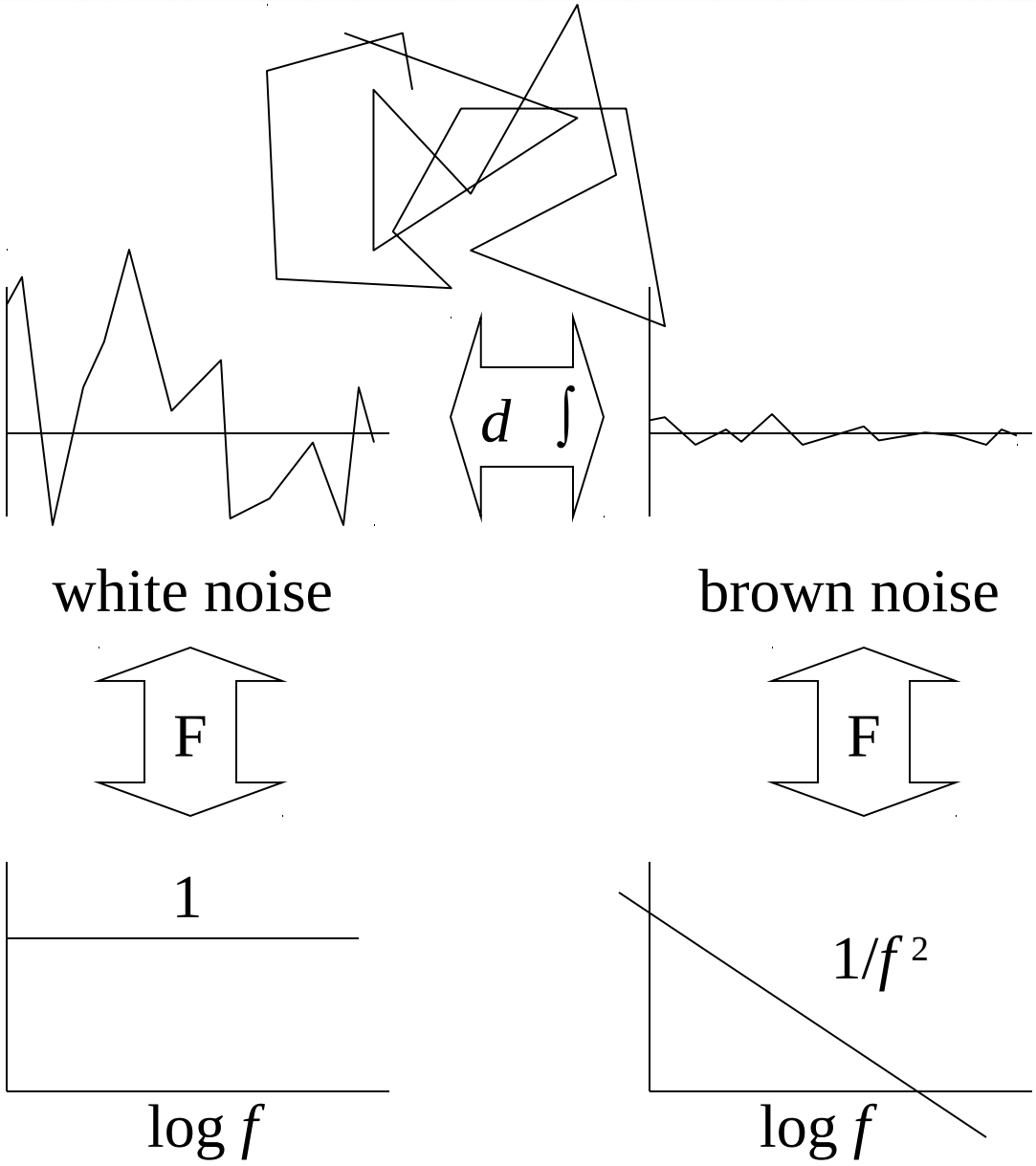
- Stucco donut
  - Noise(x,y,z)*DNoise(x,y,z)
  - Noisy direction
  - noisy amplitude

- Fleshy donut
  - Same texture
  - Different colormap

Tekstura z naključnim lomom svetlobe

Brownovo gibanje

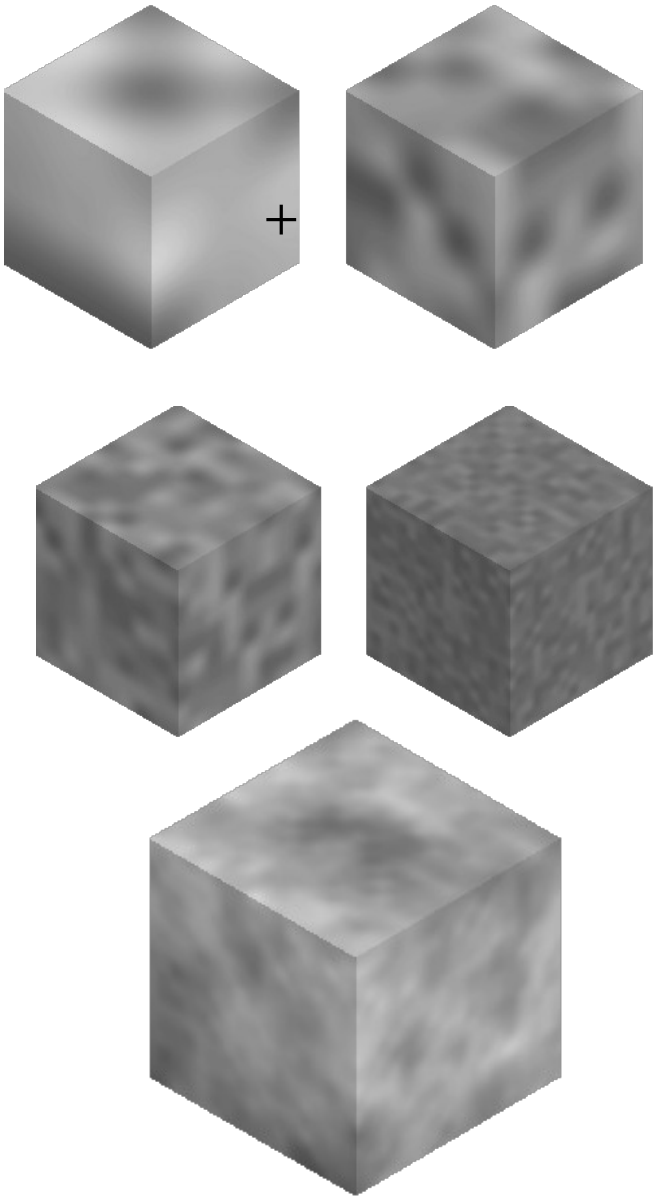
- random paths
- Integral of white noise
- $1/f^2$ distribution



Fraktalsko Brownovo gibanje

- $1/f^{\beta}$ distribution
- Roughness parameter β
  - Ranges from 1 to 3
  - $\beta = 3$ - smooth, not flat, still random
  - $\beta = 1$ - rough, not space filling, but thick
- Construct using spectral synthesis
  - Add several octaves of noise function
  - Scale amplitude appropriately

$$f(\mathbf{s}) = \sum_{i=1}^{k} 2^{-i\beta} n(2^i \mathbf{s})$$



Obroč s fraktalsko nagubano površino

```
fbm(beta) {
      val = 0; vec = (0,0,0);
      for (i = 0; i < octaves; i++) {
            val += Noise(2^i*x, 2^i *y, 2^i *z)/pow(2,i*beta);
            vec += DNoise(2^i*x, 2^i *y, 2^i *z)/pow(2,i*beta);
      }
      return vec or val;
}
```