

# *Visualisation, Rendering and Animation*

*2 VO / 1 KU (2001-2004)*

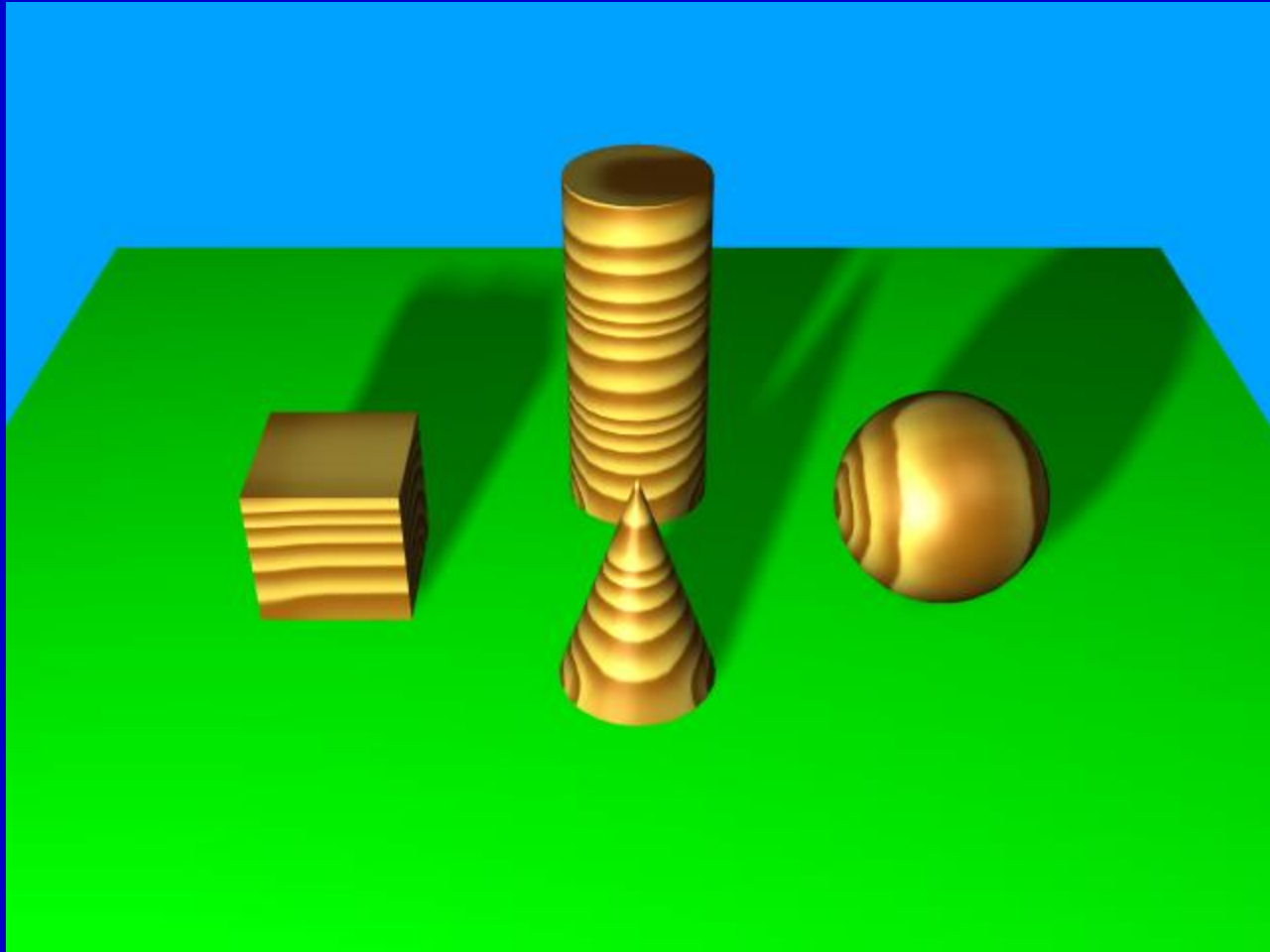
---

*Heinz Mayer, Franz Leberl & Andrej Ferko*

[ferko@icg.tu-graz.ac.at](mailto:ferko@icg.tu-graz.ac.at)

Short podcast version 2020





# *Textures*

---

***Standard Photorealism Method***

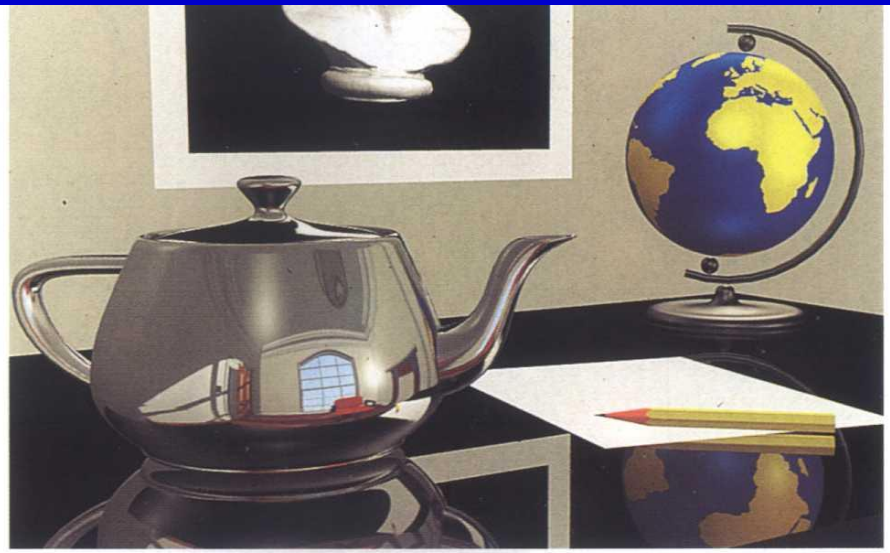
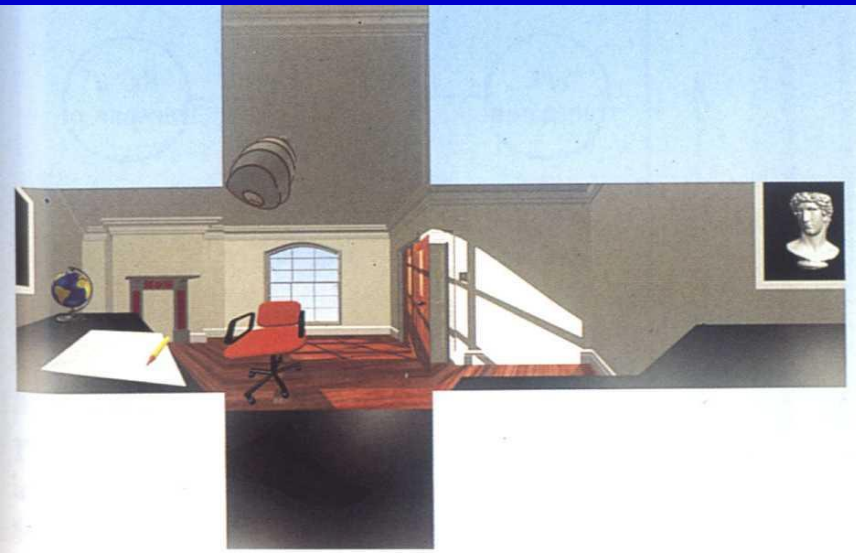


# Surface Detail

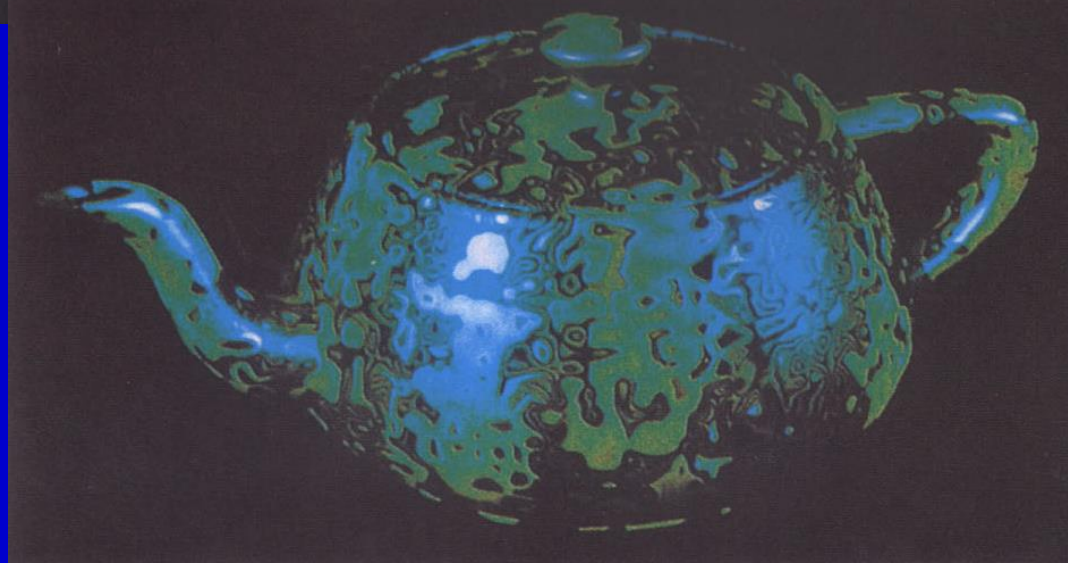
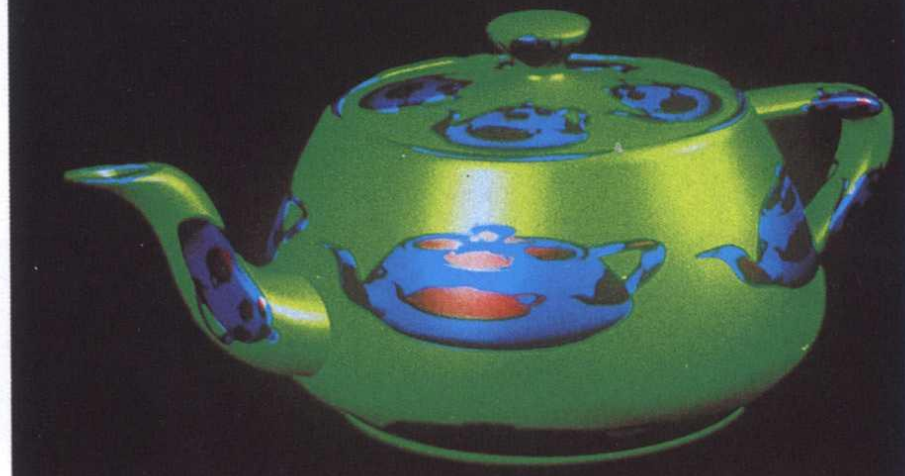
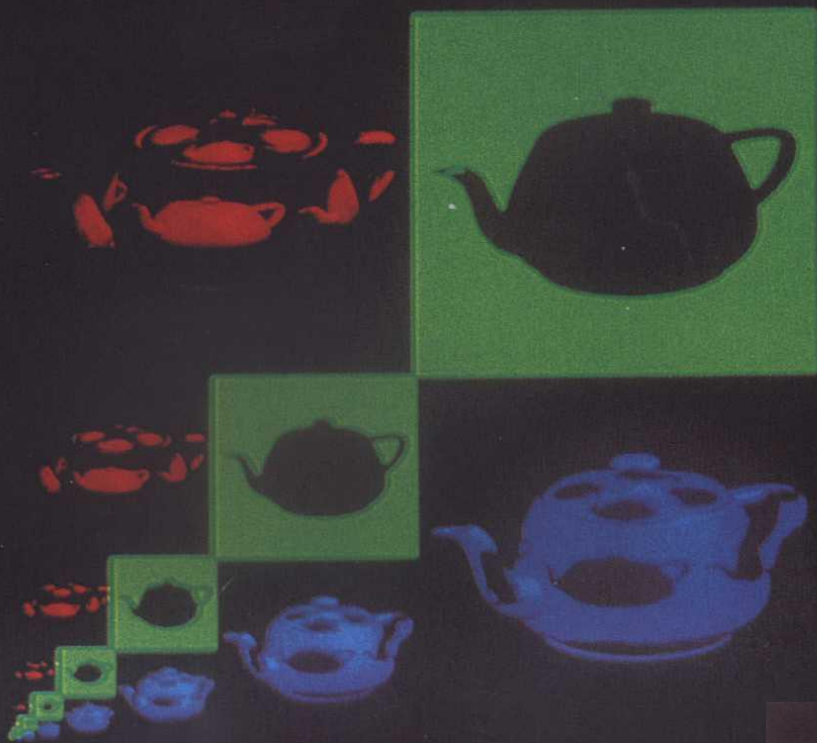
- **Idea: Improving the realism of Phong-shaded objects**
- **Modeling versus Texturing**
- **Modulated parameter:**
  - **Surface color/Transparency**
  - **Reflection (specular and diffuse)**
  - **Geometric properties**
- **Spatial and procedural textures**
- **In Maya: a texture is a collection of attributes that creates surface detail**

# *Textures at the SIGGRAPH Page*

- *[http://www.siggraph.org/education/materials/HyperGraph/mapping/r\\_wolfe/r\\_wolfe\\_mapping\\_1.htm](http://www.siggraph.org/education/materials/HyperGraph/mapping/r_wolfe/r_wolfe_mapping_1.htm)*
- ***SIGGRAPH 1997 Course Notes  
Teaching Texture Mapping Visually  
by Rosalee Wolfe***



12 Environment mapping: (*right*) an environment-mapped teapot produced from the synthetic environment map shown on the left.



2D Image Bump Mapping Using a 24-bit Bitmap



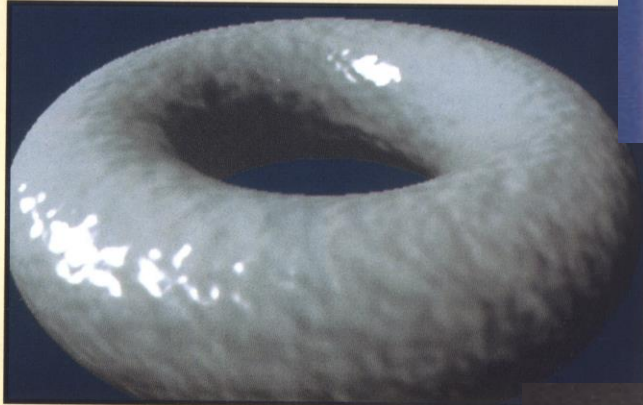


(a)

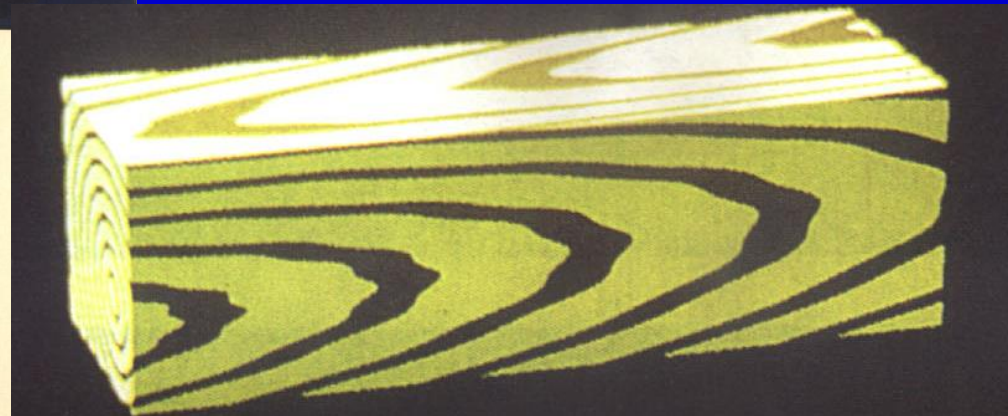
IV.7 Solid textures (a–d). The stucco doughnut is particularly effective. (Courtesy of Ken Perlin.)



(b)



(c)





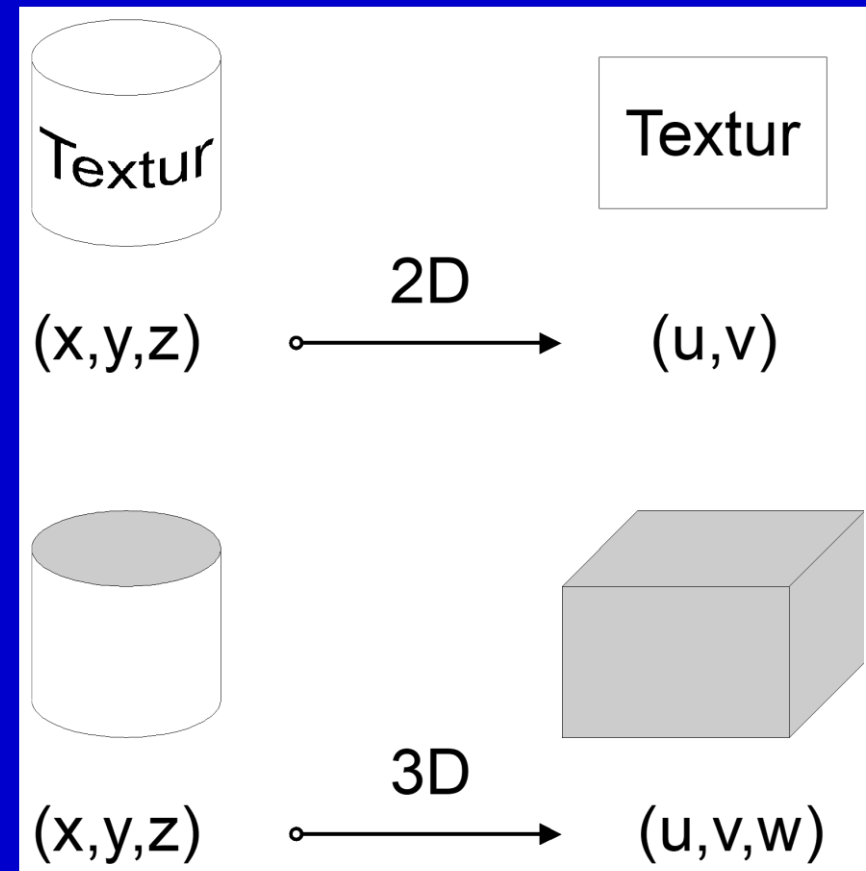
# Texture Assignment

## □ Mapping

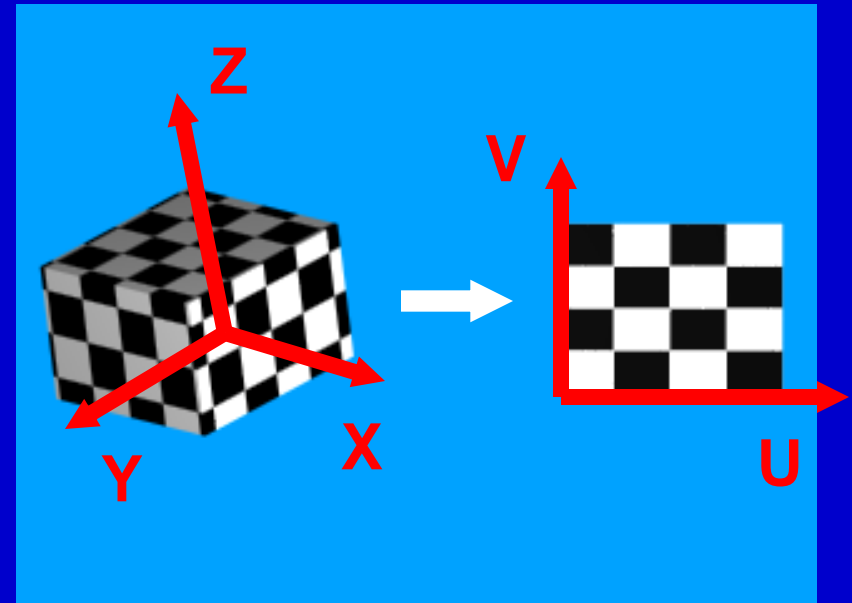
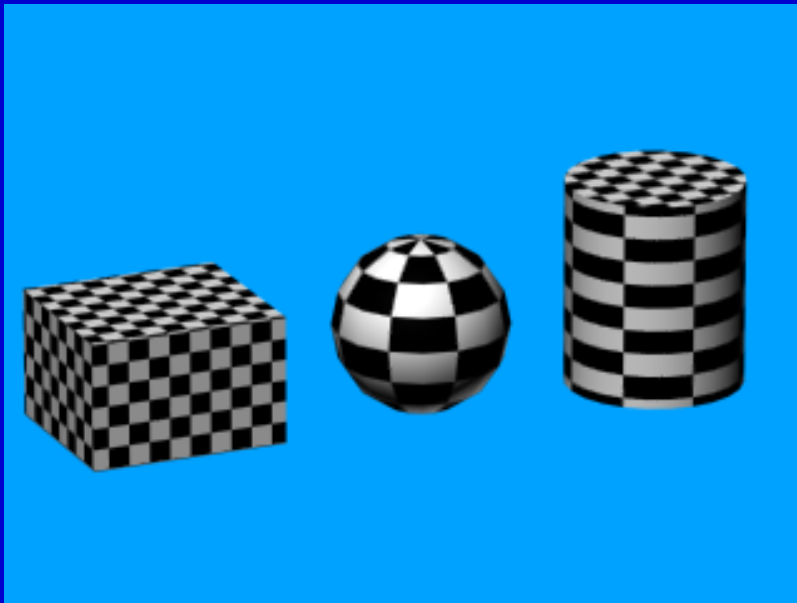
- $R^3$  in  $R^2$  for the image textures
- $R^3$  in  $R^3$  for „solid textures“

## □ Spaces

- Object space  $(x,y,z)$
- Texture space  $(u,v)$  or  $(u,v,w)$



# *Problem Definition*



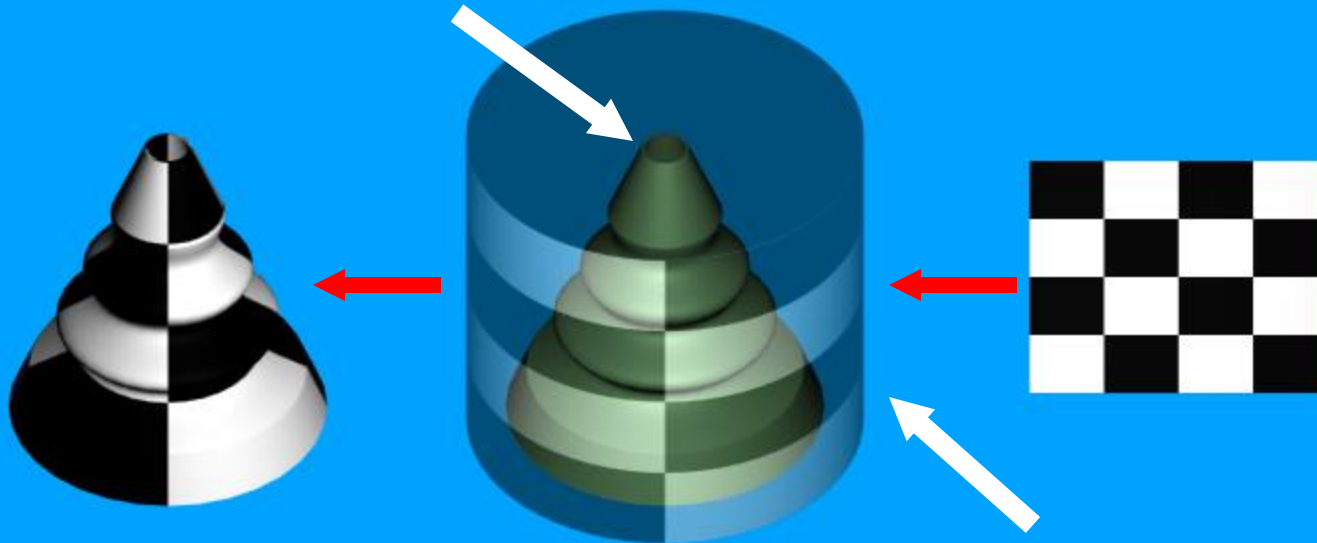
# *Practical Set Up*

- *Why? More expensive than Z-buffer, but still cheaper than other methods*
- *Problem*
  - $(u,v) = F(x,y,z)$  - to find
  - considering in the shader
- *Mapping*  
*(assigning vertices - image points)*
  - Implicit with modeling (solid textures)
  - 2-stage method
  - Reverse projection, backward mapping

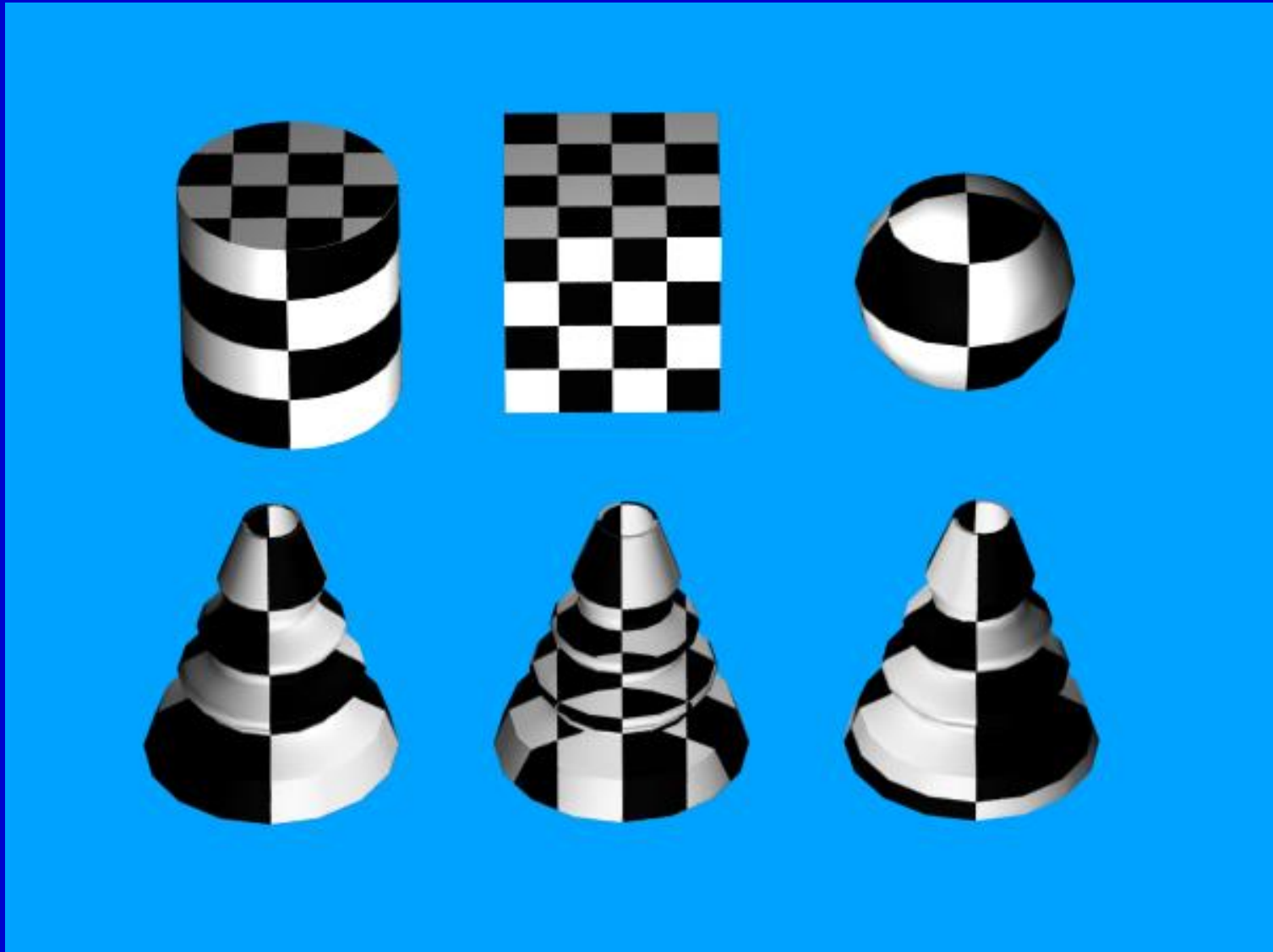
# ***Two-stage Method***

- **1. stage: „S-mapping“ (Simple)**
  - *Use a simple intermediate surface (tmp)*
  - *Projection of texture to a surface*
- **2. stage: „O-mapping“**
  - *Projection intermediate - object*
  - **4 possibilities of implementation:**
    - *Reflected ray*
    - *Surface normal*
    - *Object centroid*
    - *Intermediate surface normal*

**Object  
Surface**



**Intermediate  
Surface**



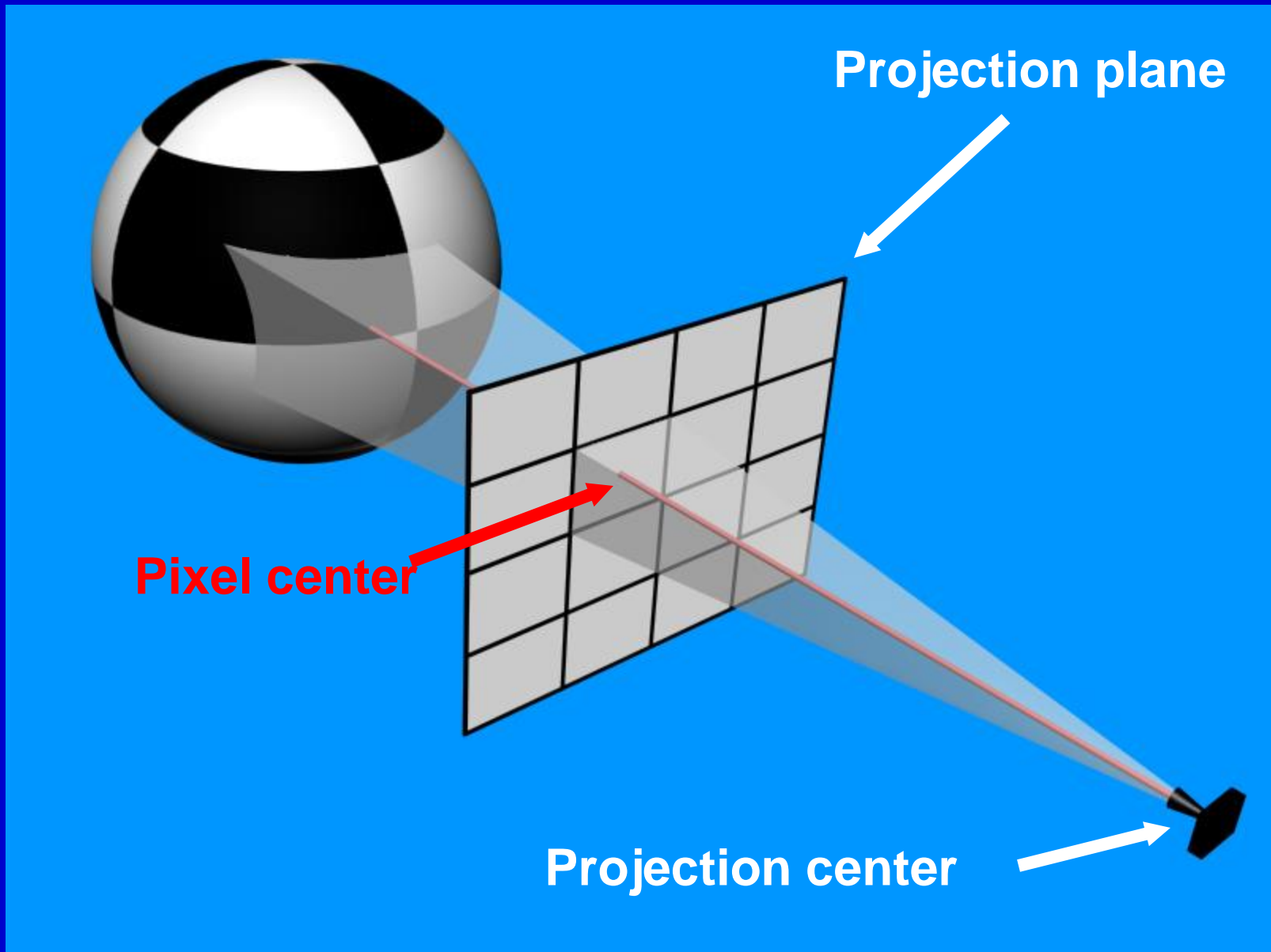
# *Texturing & Anti-Aliasing*

## □ *Problem:*

- *pixel area is not zero - it is an area*
- *spatial variation*
- *object scaling*

## □ *Method*

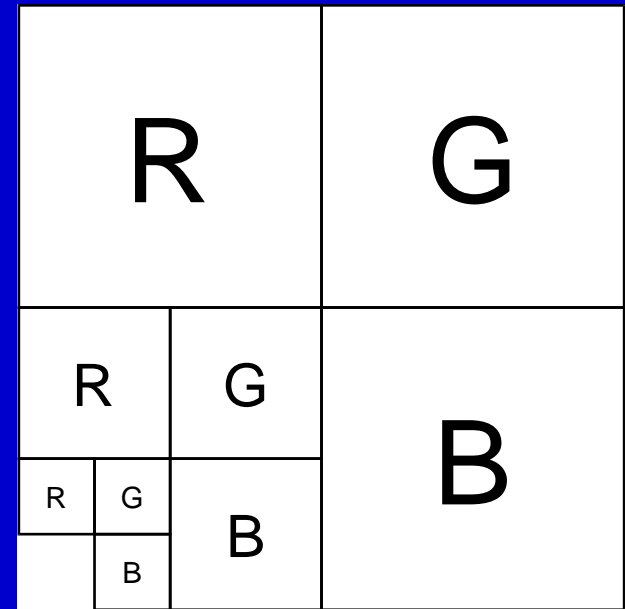
- *mip-mapping*
- *summed area tables*
- *space variant filters*





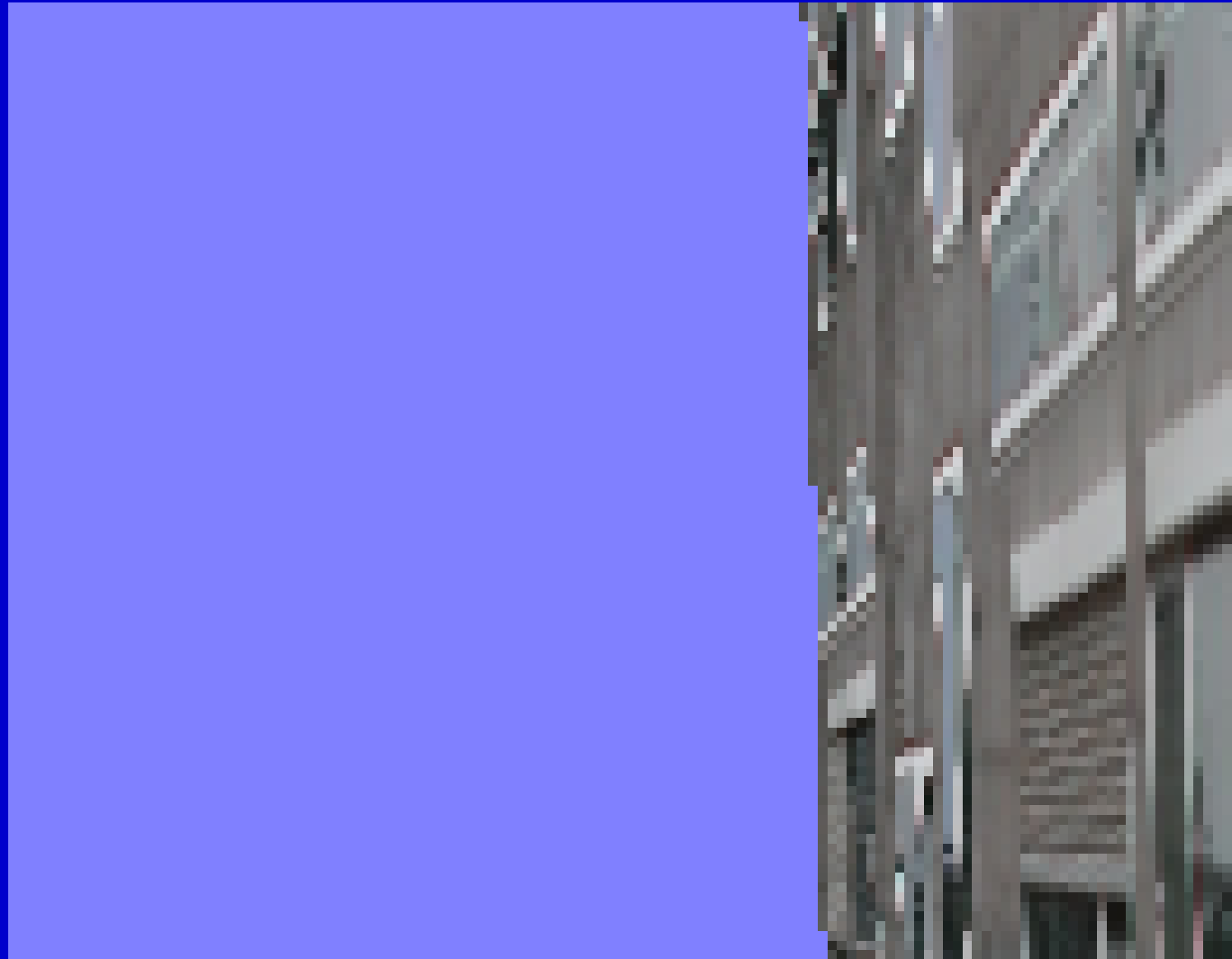
# Mip-Mapping

- **Create image pyramid**
- **Additional parameter „d“  
(u,v,d)**
- **The area of given pixel in texture space gives the scale factor**
- **Linear interpolation of 2 mip-maps**
- **Bilinear interpolation in the mip-map**
- **MIP = multos in parvo (lat.)**





# *Problem*



# Mipmapping



# Ripmapping



# *Hans Holbein jr.*

## *The Ambassadors*

*Mipmap fails*

*Ripmap - better*

*Fipmap Experiment by Alex Bornik*

*Morale: textures are hierarchic*

*low resolution for tuning,  
hi res for final result*



H

sadors



ring & Animation  
Leberl & Andrej Ferko

H  
Fi

sa  
DI

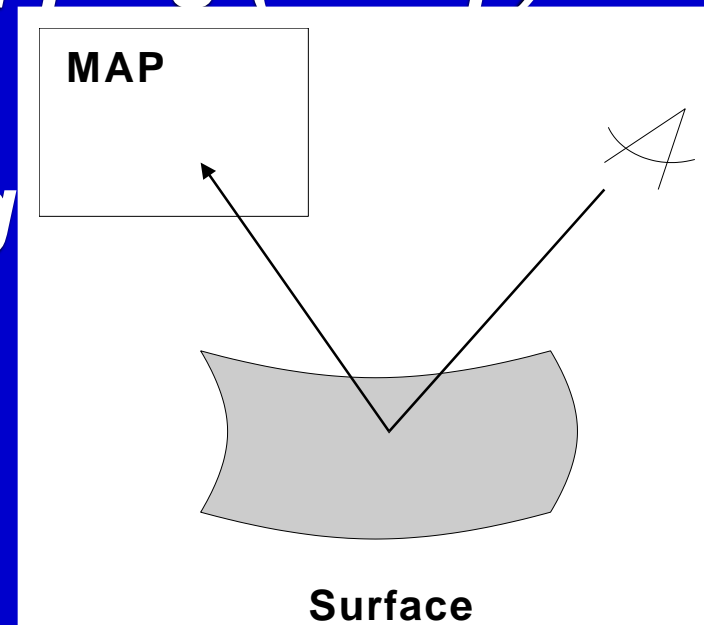


rin  
Leb



# *View-Dependent Mapping*

- *Texture in given surface point changes with the camera position*
- *Method*
  - *Chrome/reflection mapping (cheap)*
  - *Refraction mapping*
  - *Environment mapping*



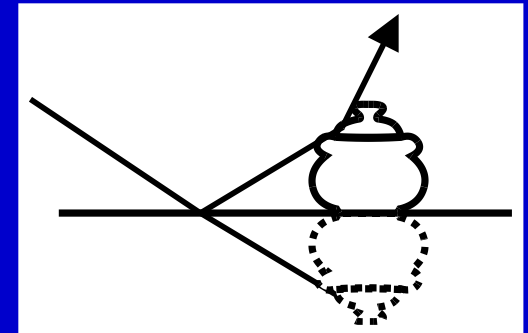
# Environment Mapping

- **Map-Production (6 times planar)**

- Synthetic generated
- Photographs

- **Extension**

- Compute the reflection on bounding surfaces (multiple mapping)
- Composition with the generated image



# Bump Mapping

Idea: Simulation of rough surfaces  
varying the surface normals

Formalism (Blinn 78):

$$N = P_u \times P_v$$

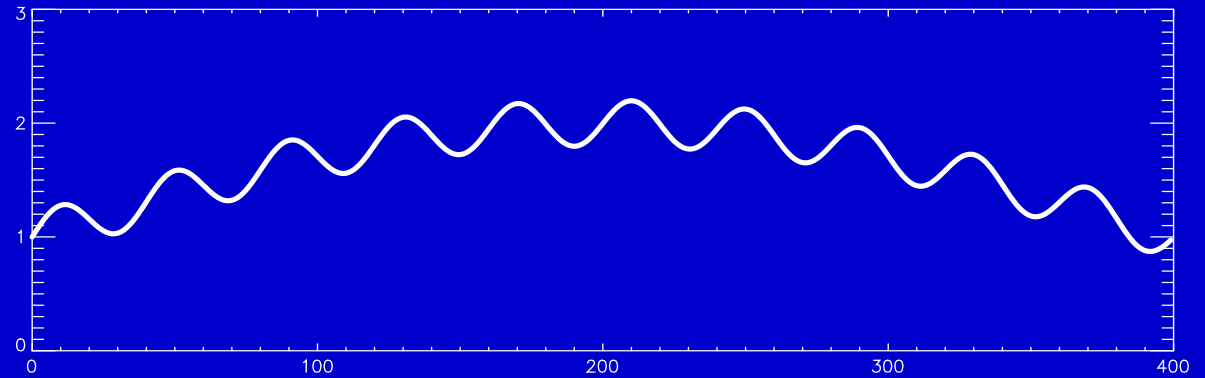
$$P' = P + B(u, v)N$$

$$P'_u = P_u + B_u N + B N_u \quad \text{analogously for } v$$

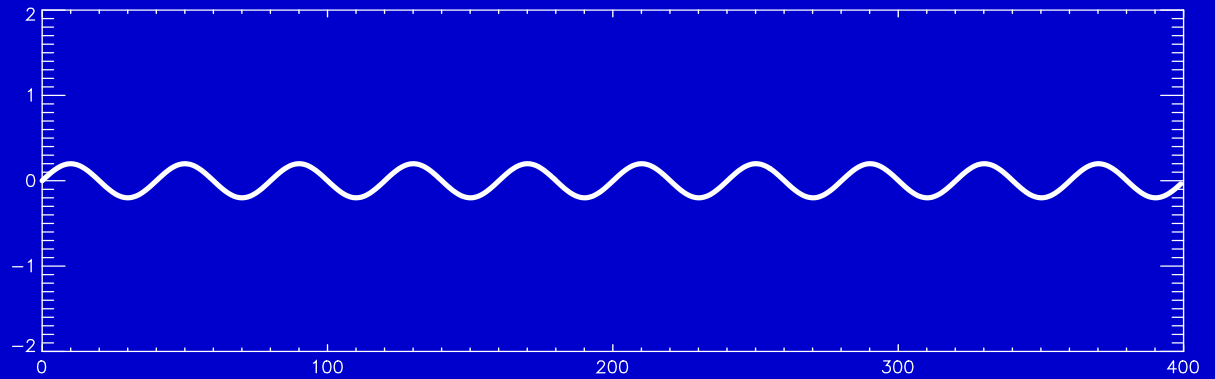
$$N' \cong P_u \times P_v + B_u (N \times P_v) + B_v (P_u \times N) + B_u B_v (N \times N)$$

$$N' \cong N + D$$

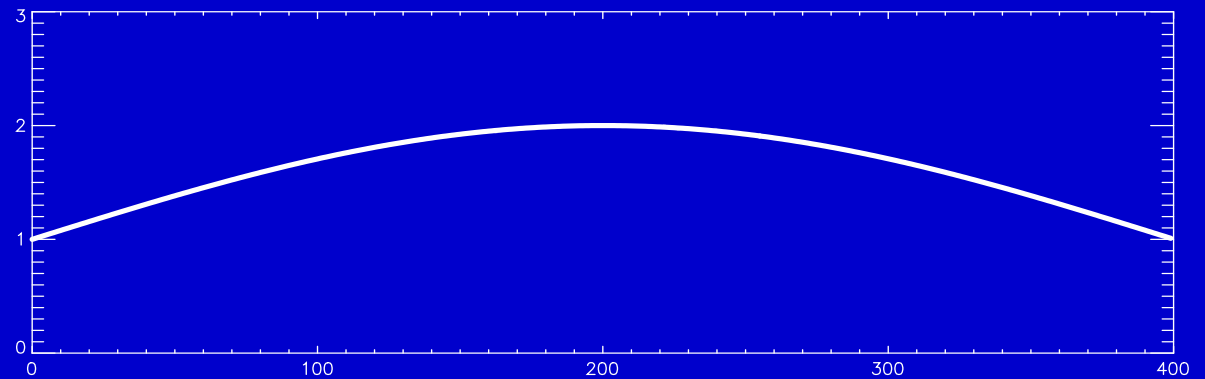
$$P' = P + B$$



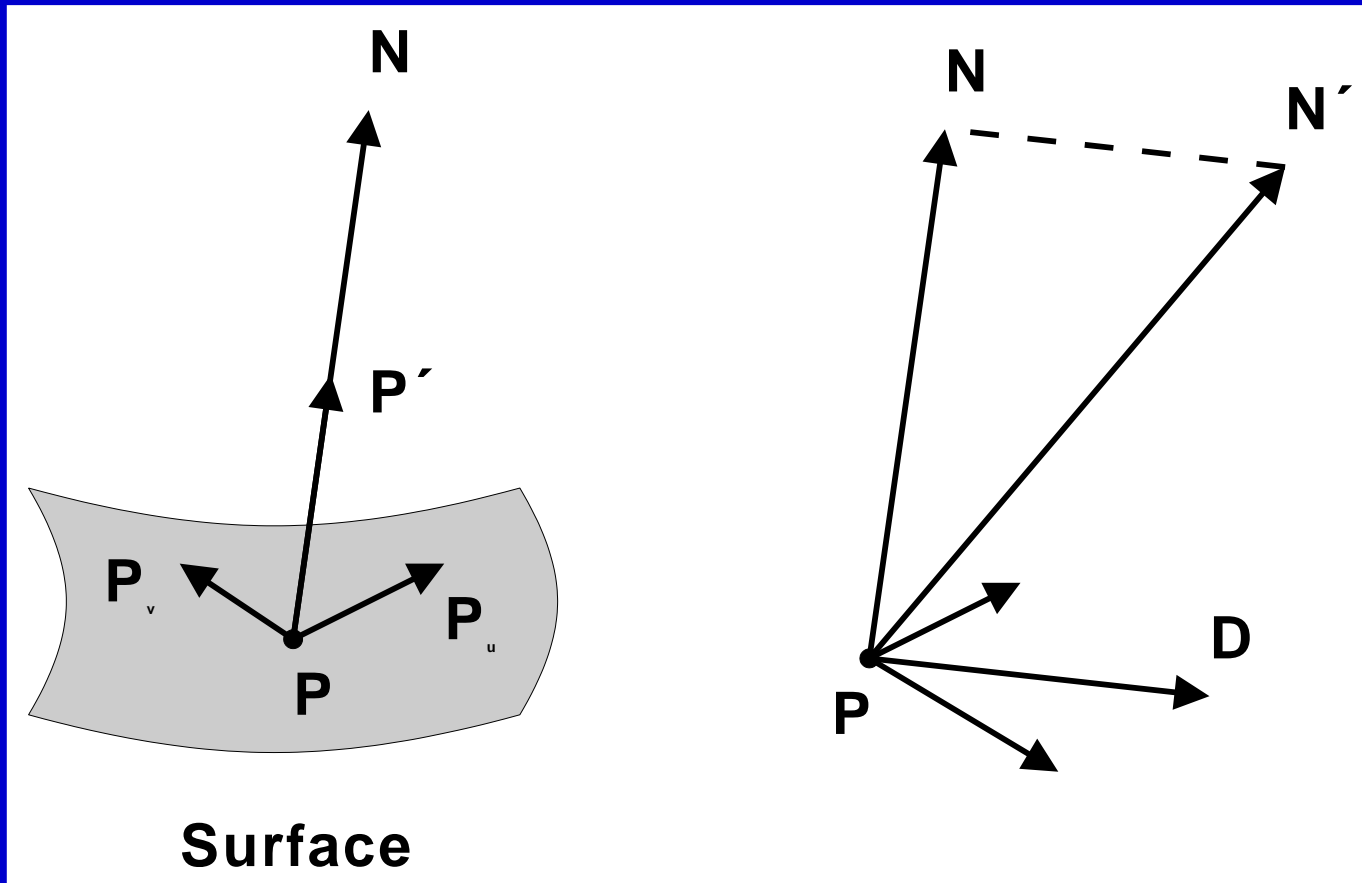
$B$



$P$



# Geometric Situation



# *Summary (Bump-Map)*

- *Explicite substitution in details modeling*
- *Silhouette edges remain unchanged*
- *Scaling of scene objects*
- *Anti-Aliasing*
- *Alternative: displacement mapping*



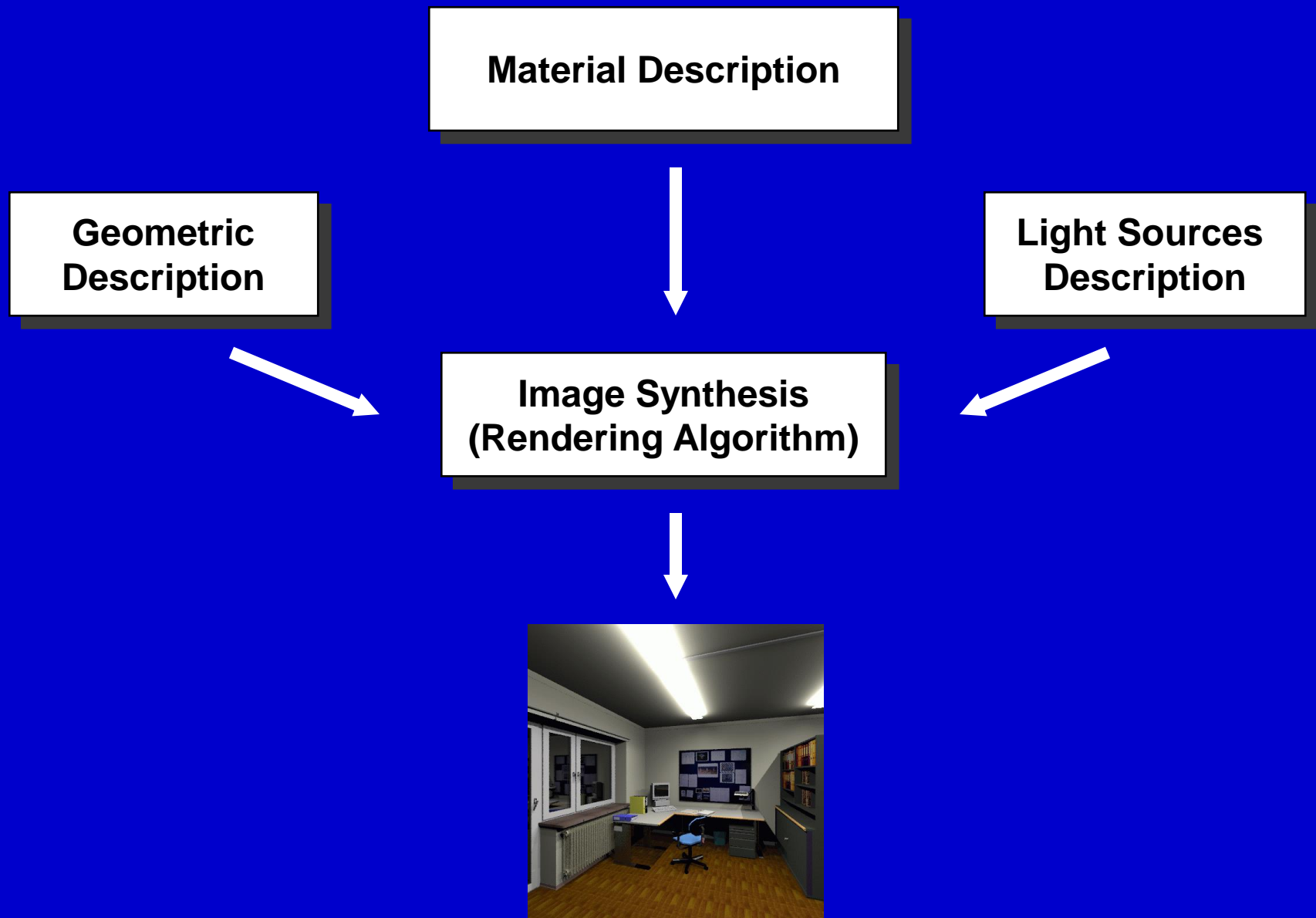






Figure 12. A still life image showing examples of procedural and scanned textures and patterns.

# *Perlin 1989 et al.: procedural approach*



# *Summary*

## *(Procedural Textures)*

### □ *Pros*

- *Less memory required*
- *No fixed resolution*
- *None limitations in propagation*
- *One type contains the whole class*

### □ *Cons*

- *Building and verifying complex*
- *Anti-Aliasing more complex*

# *Applications (Surface texturing)*



# *Applications (Solid Textures)*

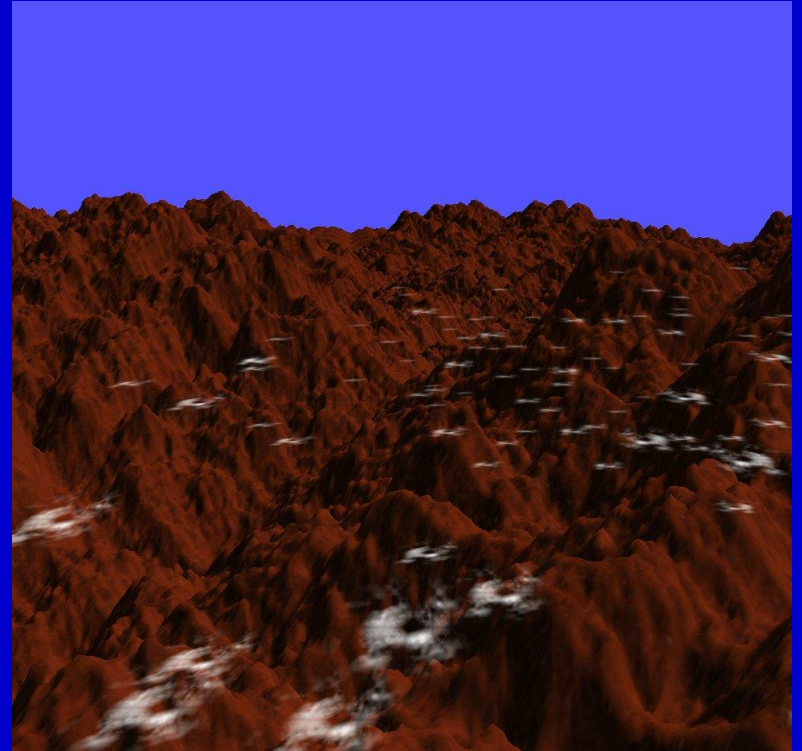
□ **Solids**

□ **Gases**

## Problems:

- *Objects within the gases must be visible*
- *Gasiform shapes may cast shadows*
- *Self shadowing*





# Summary (Textures)

- *Fundamental drawing tool*
- *Hardware implemented*
- *Extremely useful for realism: Toy Story, Doom, Final Fantasy, Shrek*
- *In happy cases - no 3D model, just the procedural textures (NYU torch by Perlin)*
- *Best way for fast rendering and VR*
- *Alternatives: billboards, sprites...*





# *NYU Torch by Ken Perlin*



□ <http://mrl.nyu.edu/~perlin/>

***Thank You...***

*... for Your attention.*



# *Visualisation, Rendering and Animation*

*2 VO / 1 KU (2001-2004)*

---

*Heinz Mayer, Franz Leberl & Andrej Ferko*

[ferko@icg.tu-graz.ac.at](mailto:ferko@icg.tu-graz.ac.at)

Short podcast version 2020

