*Department of Applied Mathematics*
*and Computational Sciences*
*University of Cantabria*
***UC-CAGD Group***

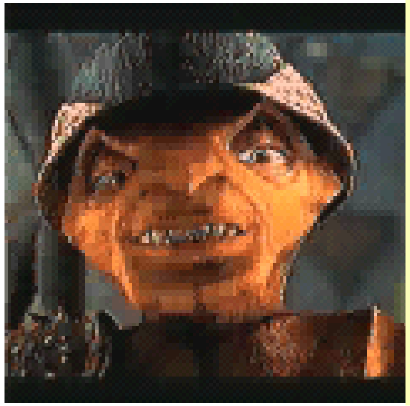
# COMPUTER-AIDED GEOMETRIC DESIGN AND COMPUTER GRAPHICS:

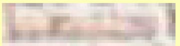
# TEXTURE AND BUMP MAPPING

***Andrés Iglesias***
**e-mail: iglesias@unican.es**
**Web pages: http://personales.unican.es/iglesias**
**http://etsiso2.macc.unican.es/~cagd**

# Texture

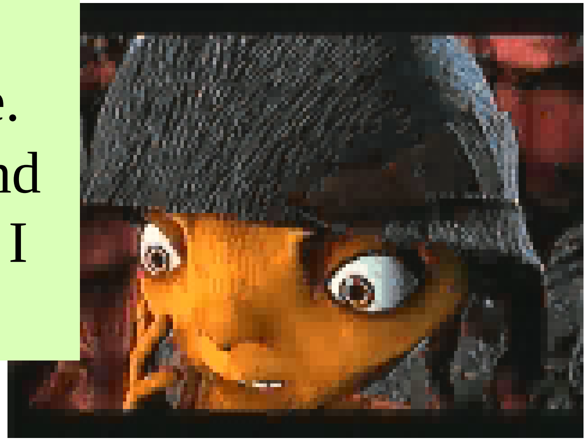
## *Motivation*



Today, we're going to build a brick wall. It is easy! Each ant must take a brick like this  With some hundreds of ants we'll finish the wall before the battle. Come on!

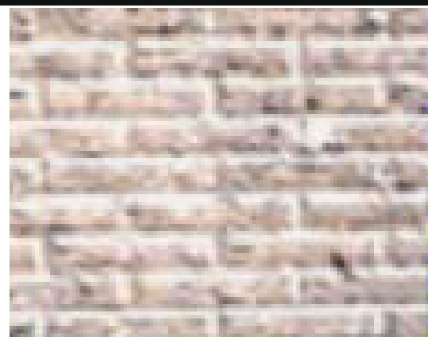
1 brick,
2 bricks,
3 bricks,
....

*My God!* We won't finish the wall in time. They arrive before, and we'll die!!! What can I do alone?



Don't worry! I'm so strong...

I have a trick!! We only need to apply a single textured polygon. Something like this:



# *Texture*

In computer graphics, the fine surface detail on an object is generated using *textures*.

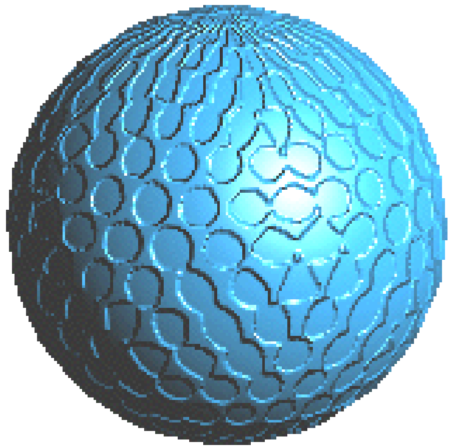
Three aspects of texture are generally considered:

1. *Texture mapping:* the addition of a separately specified pattern to a smooth surface. After the pattern is added, the surface remains smooth.

Also known as *patterns* or *colour detail*

2. *Bump mapping:* the addition of roughness to the surface. This is obtained perturbation function that changes the geometry of the surface.

Also known as *roughness*



3. *Simulating environments:* for example, shadows and lighting using textures.

# Texture

## Texture mapping

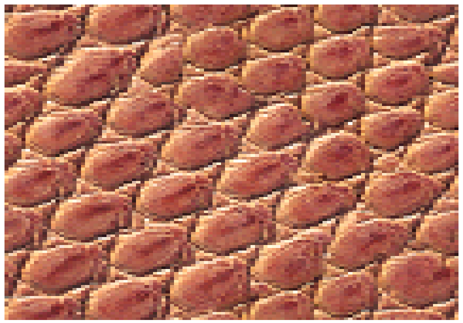
The basis of adding texture patterns is mapping:

### Texture Space

$v$

$(u,v)$

$u$



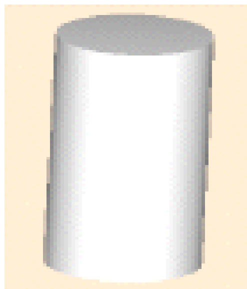
### Object Space

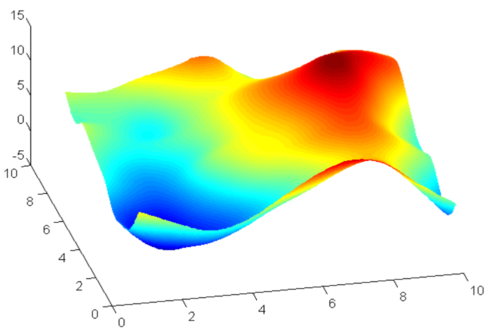
$z_o$

$(x_o, y_o, z_o)$

$x_o$

$y_o$



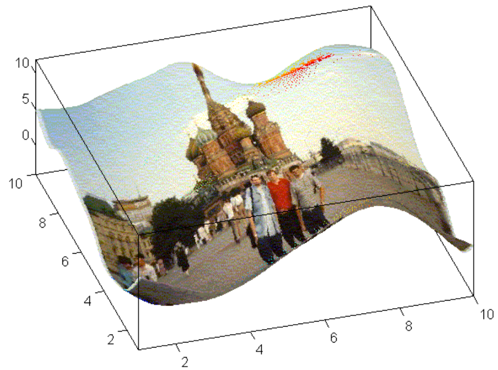



### Image Space

$(x_s, y_s)$

$y_s$

$x_s$



# *Texture*

## *Texture mapping*

Object space mapping:

We map an image onto the surface of an object.

Texture pattern defined in an orthogonal coordinate system *(u,v)* in texture space

The surface is defined in a second orthogonal coordinate system *(x,y,z)* represented in a parametric space

The surface is represented in parametric space *(s,t)* as: $x(s,t),\ y(s,t),\ z(s,t)$

Therefore, we need to determine the mapping function between the texture space and the parametric space:

$$s=f(u,v) \quad , \quad t=g(u,v)$$

The inverse mapping from parametric space to texture space is:

$$u=F(s,t) \quad , \quad v=G(s,t)$$

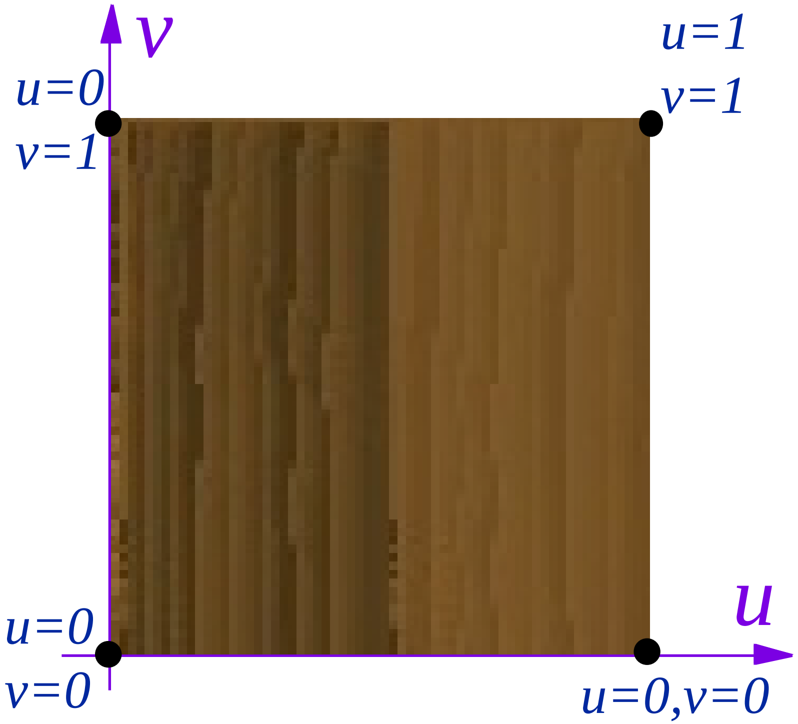
# Texture

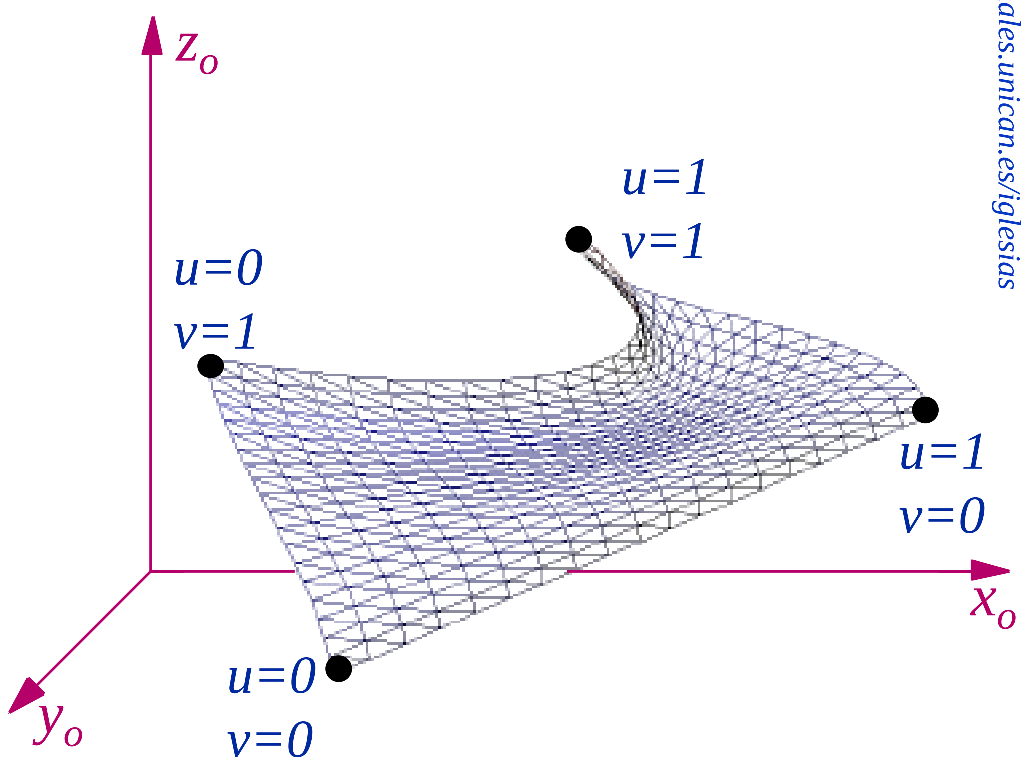
## Texture mapping

**Texture Space** | **Object Space**

Given a particular *(x,y,z)*:

- Compute the corresponding *(u,v)*
- shade the surface with the colour pointed to in the image by *(u,v)*



# Texture

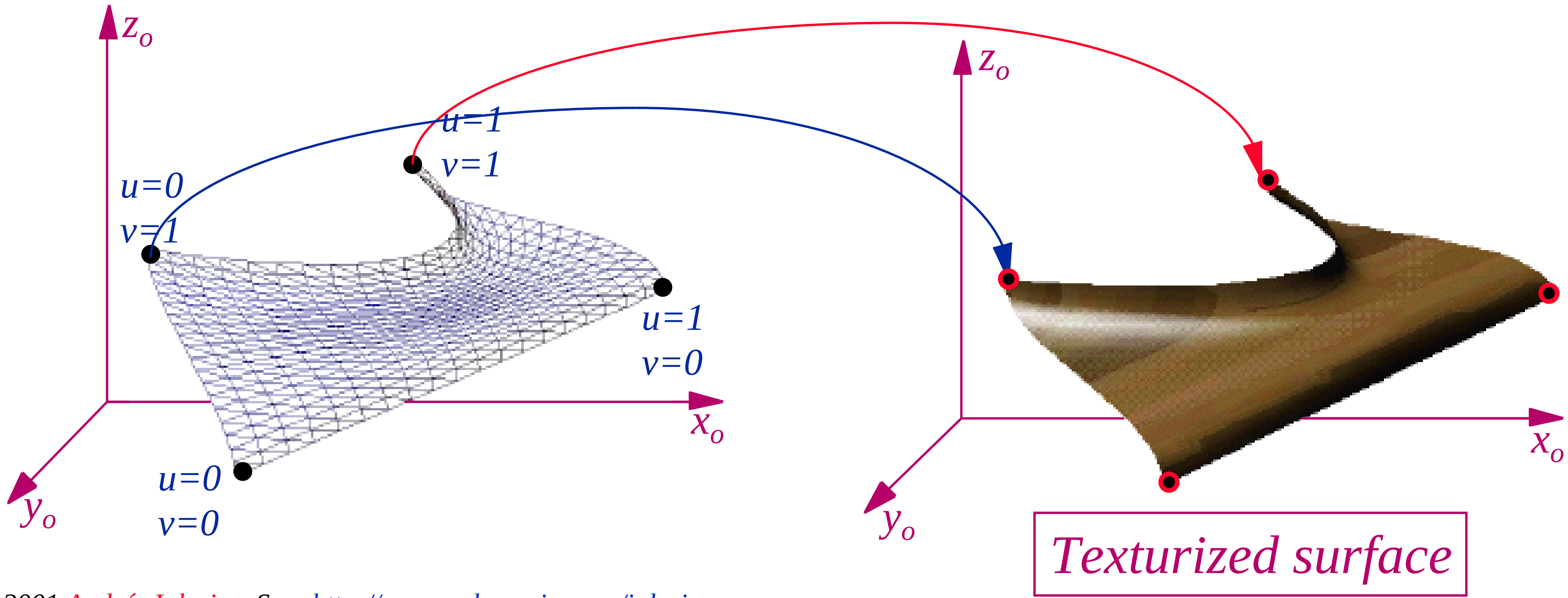
## Texture mapping

Mapping functions: $s=f(u,v) \quad , \quad t=g(u,v)$

The mapping functions are frequently assumed to be linear:

$$s=A\,u+B$$
$$t=C\,v+D$$

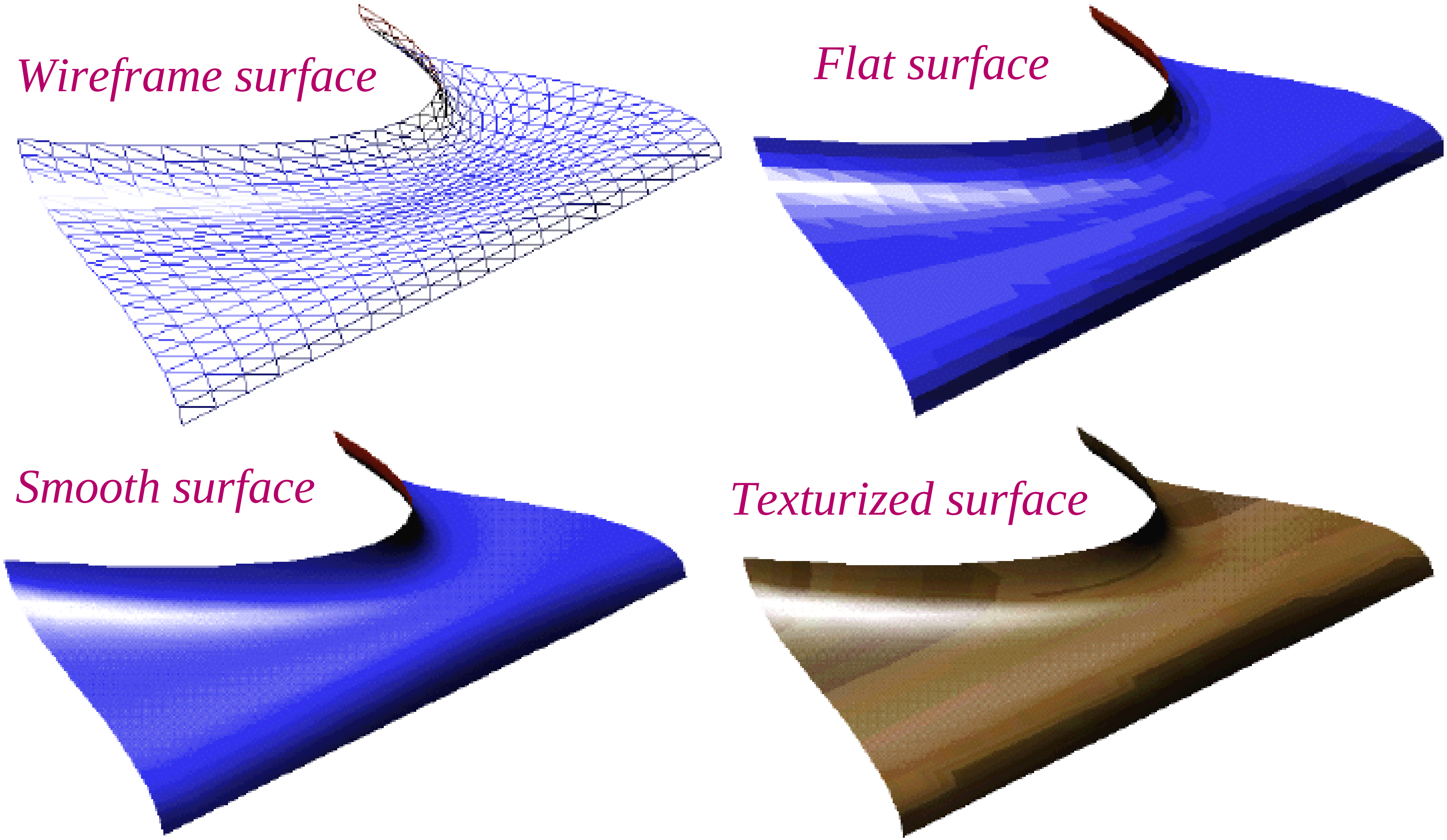
where the constants $A$, $B$, $C$ and $D$ are obtained from the relationship between known points in the two systems.



# *Texture*

## *Comparing models*





# Texture

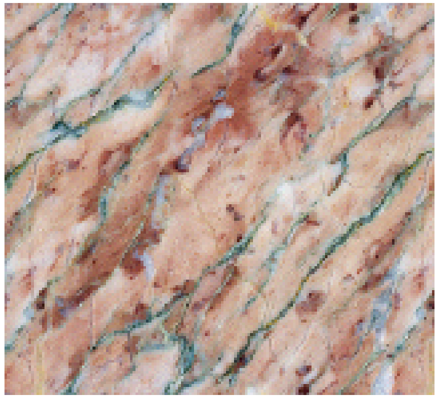
## Texture mapping

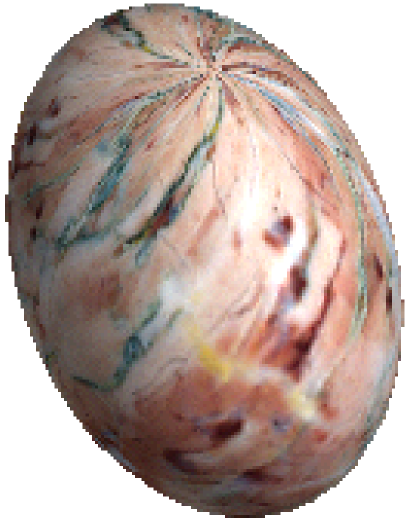
However, linear mapping functions may lead to *unsatisfactory results*.

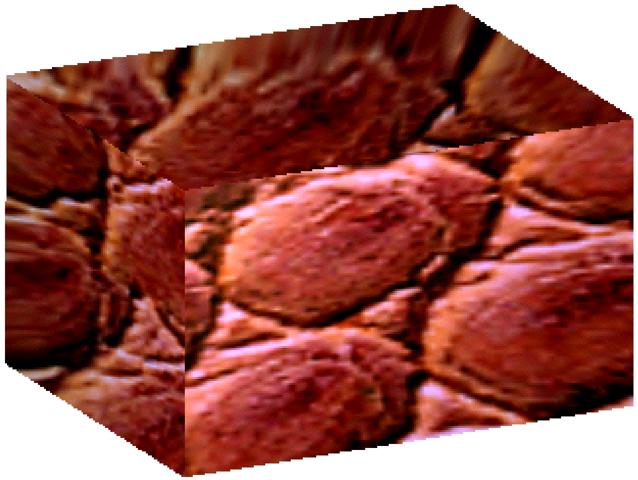
It's better to use nonlinear functions (because the mapping between parametric space and object space is nonlinear).

*Texture Space*

*Object Space*







# Texture

## Texture mapping

**Bier, E.A., Sloan, K.R. *Two-part Texture Mappings*, IEEE Comput. Graph. and Appl., Vol. 6, Nº 9, pp. 40-53, 1986.**

### Two-part mapping

Overcomes the mapping problems introducing an *"easy" intermediate surface*.

- First, mapping the texture image onto a simple three-dimensional surface (a plane, a cylinder, a sphere or a box). This is known as the *S mapping*.

- Then mapping the result onto the final three-dimensional surface. This is referred to as the *O mapping*.

$$(u,v) \xrightarrow{\;S\;} T'(x_i, y_i) \xrightarrow{\;O\;} O(x_s, y_s, z_s)$$

*Texture Space* → *Intermediate space* → *Object space*

# Texture

## *Texture mapping*

### Two-part mapping

For the *S mapping* the authors described four intermediate surfaces:

1.- A plane at any orientation

To align the texture with the plane require:

3 rotations + 3 translations

Then, the pattern must be scaled. Ignoring rotations and translations, the transformation is given by:

$$(u,v) \longrightarrow (a\, x_i, d\, y_i)$$

$a$, $d$ - scaling factors

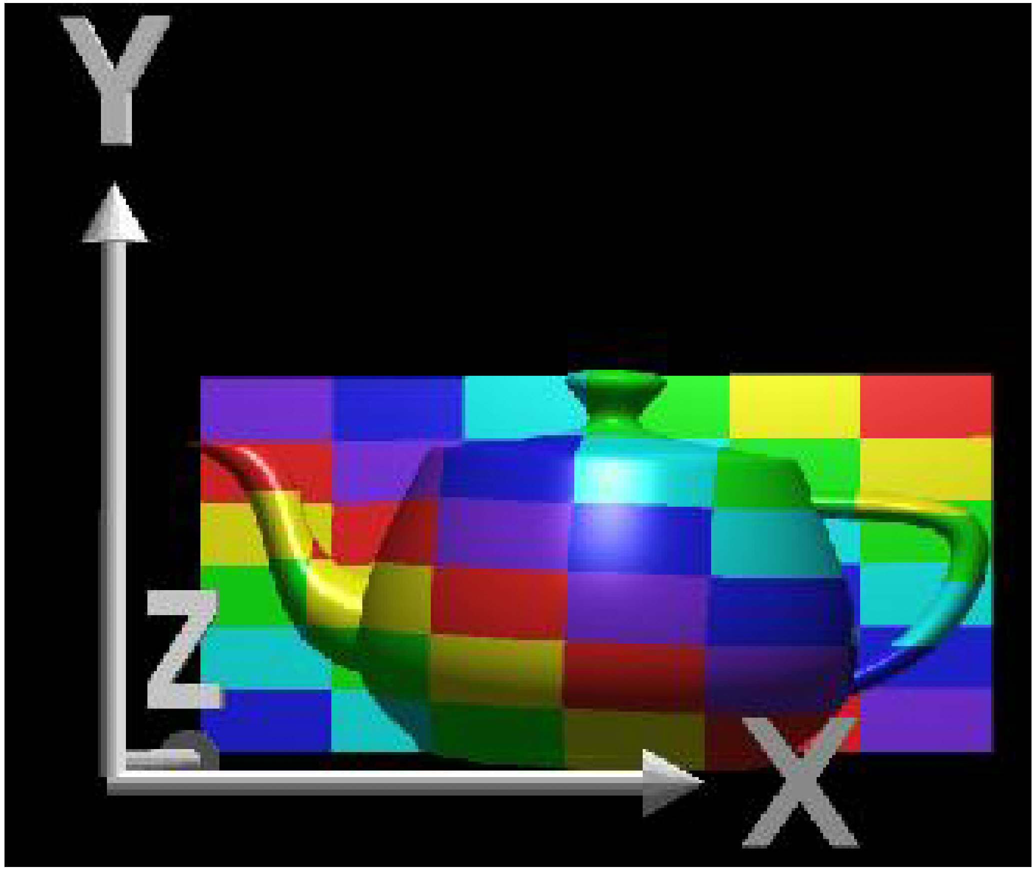



*Image from: © SIGGRAPH'97 R. J. Wolfe (DePaul University)*

# Texture

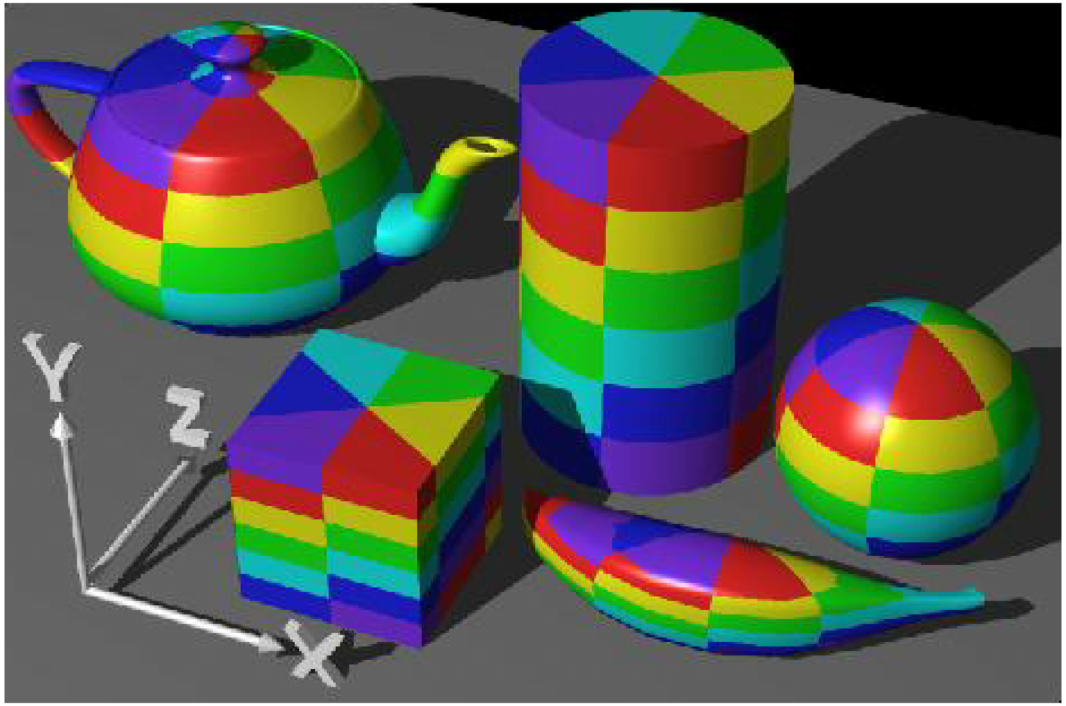
## Texture mapping

Two-part mapping

2.- The curved surface of a cylinder (Useful for surfaces of revolution)

$$(u,v) \longrightarrow [a\, r\, (\theta-\theta_0), d\, (h-h_0)]$$

where $a$, $d$ are scaling factors, and $\theta_0$ and $h_0$ position the texture on the surface of the cylinder of radius $r$.




*Images from: © SIGGRAPH'97 R. J. Wolfe (DePaul University)*

# Texture

## *Texture mapping*

Two-part mapping

3.- The surface of a sphere

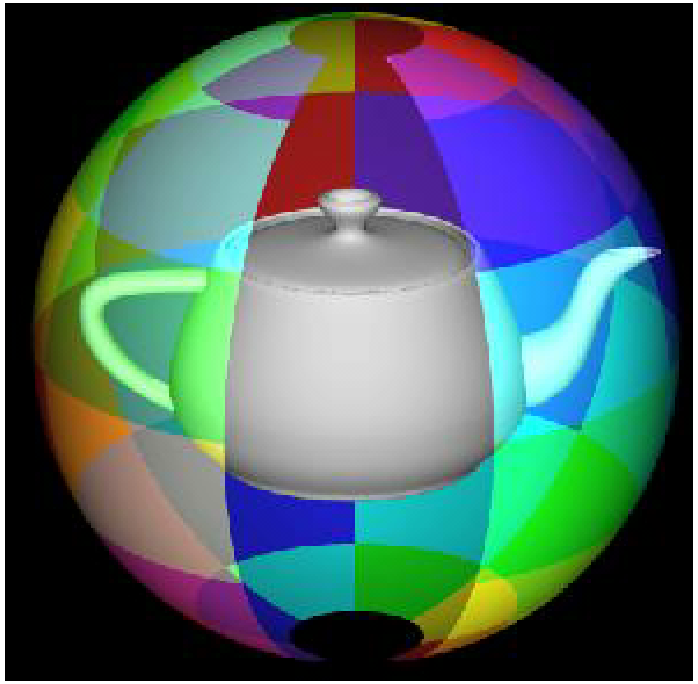
Using the stereographic projection:

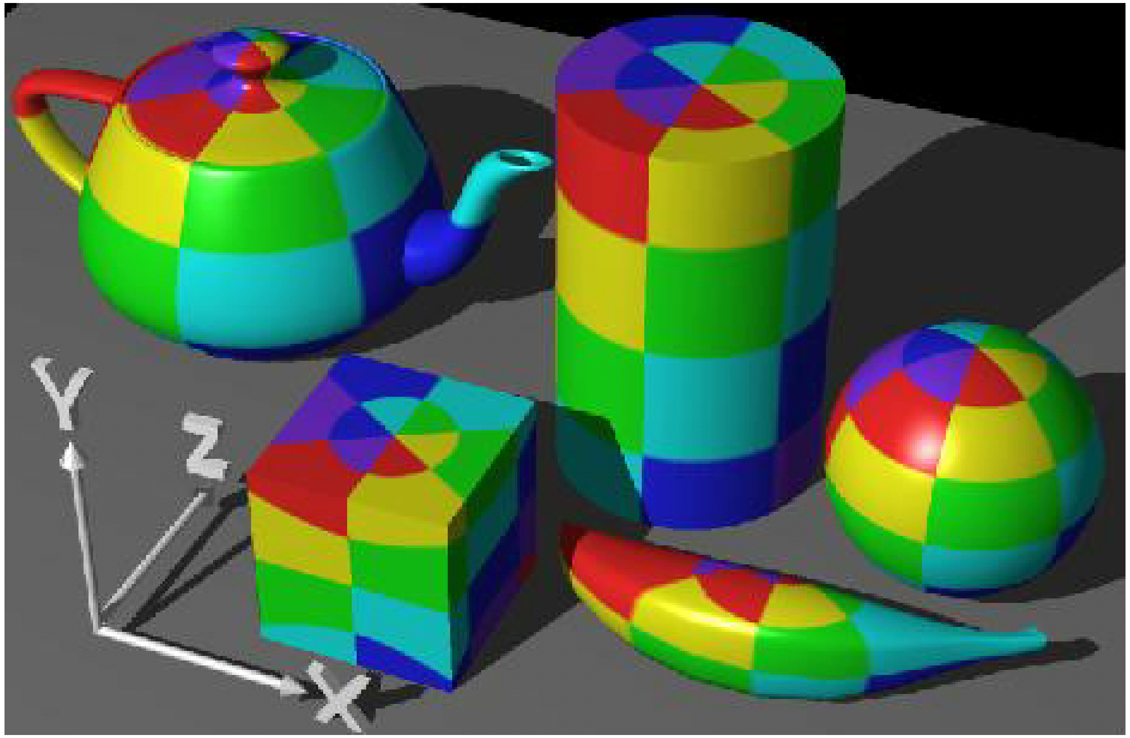
$$(u,v)=(2p/C,2q/C) \longrightarrow (\theta, \varphi)$$

where $(\theta, \varphi)$ are the equatorial and polar variables for the sphere and:

$$C=1+\mathrm{sqrt}(1+p^2+q^2)$$

$$p=\tan(\theta)\cos(\varphi)\ ,\quad q=\tan(\theta)\cos(\varphi)$$






*Images from: © SIGGRAPH'97 R. J. Wolfe (DePaul University)*

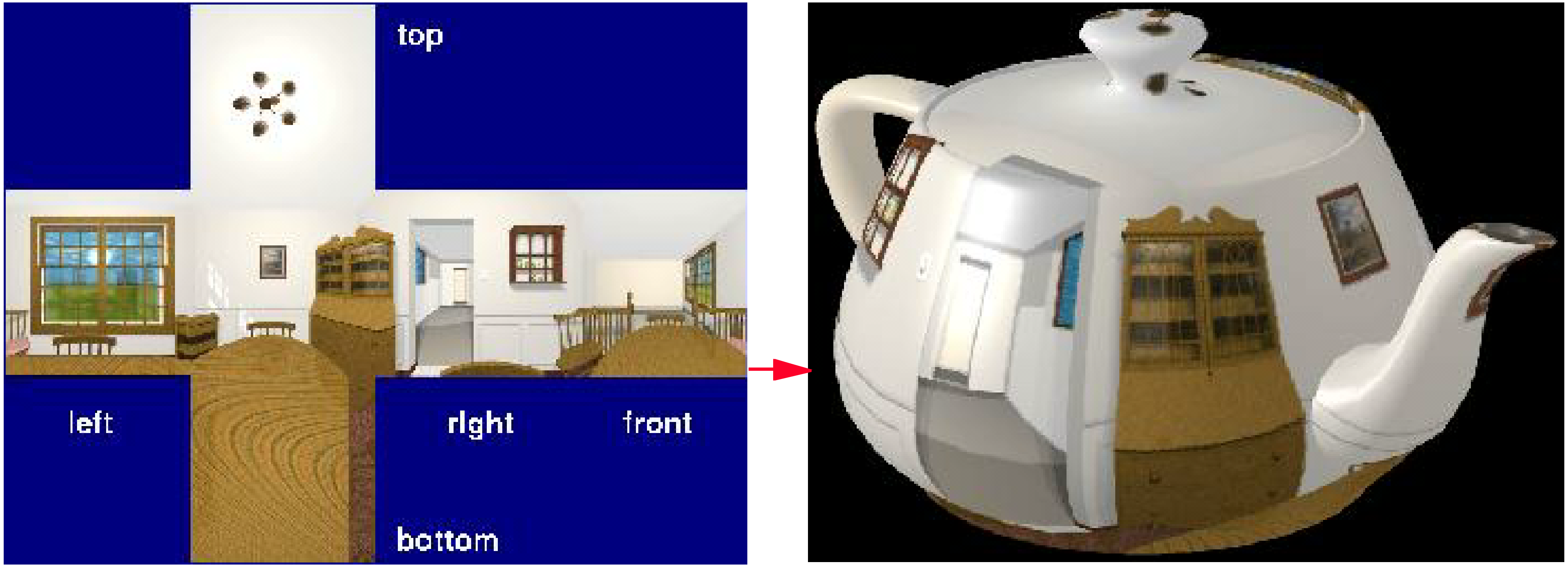
# Texture

## Texture mapping

### Two-part mapping

#### 4.- The faces of a cube

Interesting enough, since a box is topologically equivalent to an sphere.




Images from: © SIGGRAPH'97 R.J. Wolfe (DePaul University)

Shortcoming: nonadjacent pieces of the texture are now adjacent both on the box surface and the final three-dimensional surface, leading to possible discontinuities.

# Texture

## *Texture mapping*

### Two-part mapping

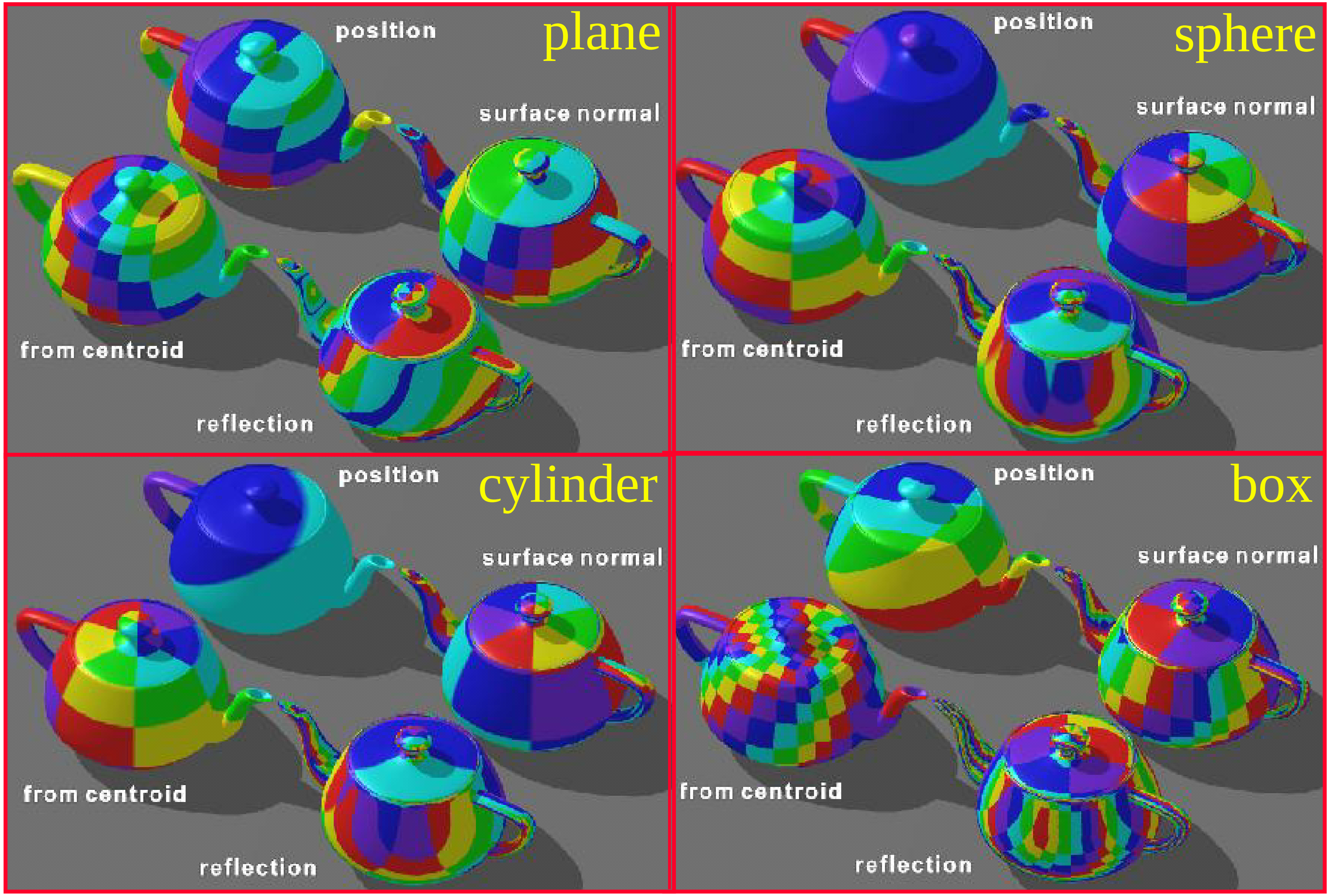
For the *O mapping* we also have four mapping techniques (which map the texture from the intermediate surface to the object):

- *Reflected ray:* trace a ray from the viewpoint of the object and then trace the resulting reflected ray from the object to the intermediate surface. This is in fact the *environment mapping*.
- *Object normal:* find the intersection of the normal to the object surface with the intermediate surface.
- *Object centroid:* intersect the line defined by the centroid of the object and a point on the object surface with the intermediate surface.
- *Intermediate surface normal (ISN):* trace a ray in the direction of the normal at a point on the intermediate surface to find its intersection with the object.

Reflected ray is ignored because is viewpoint dependent and not very useful.

# *Texture*

**Two-part mapping** three *O mapping* x four *S mapping* = 12 combinations



# Texture

## Texture mapping

Two-part mapping

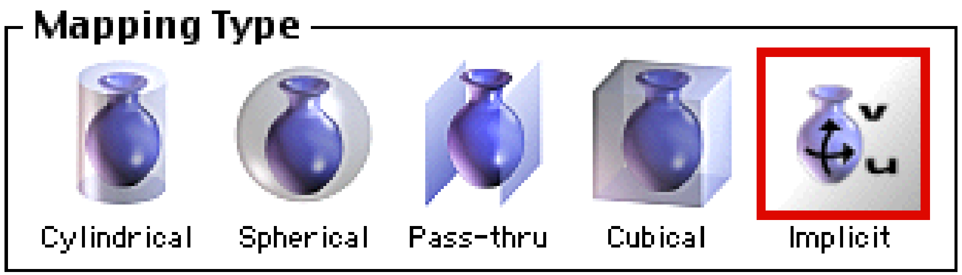
**Rogers, D.F.. *Procedural Elements for Computer Graphics*, 2nd. Edition, McGraw-Hill, Boston, 1998.**

However, only five mappings are really useful:

| | Plane | Cylinder | Sphere | Box |
|---|---|---|---|---|
| Object normal | Redundant | Poor | OK | OK |
| Object centroid | Redundant | Poor | centroid/sphere | centroid/box |
| I.S.N. | slide projector | shrinkwrap | Redundant | ISN/box |

Commercial software already incorporates two-part mapping techniques:

***Painter 3D***




***Amorphium***

Flat
Cylindrical
Spherical

# Texture

## Bump mapping

Adding texture patterns to smooth surfaces produces smooth surfaces.

Using a rough-textured pattern to add the appearance of roughness to a surface is not a good idea. Rough-textured surfaces hava a small random component in the surface normal and hence in the light reflection direction.

**Blinn, J.F.,** ***A scan line algorithm for the computer display of parametrically defined surfaces*****, Comput. Graph., Vol. 12, 1978 (supplement SIGGRAPH'78).**

Blinn developed a method to for perturbing the surface normal.

At any point of the surface S, the partial derivatives are $S_u$ and $S_v$. The surface normal n is given by the cross-product: $n = S_u \times S_v$

Blinn defined a new surface S' as: $$S'(u,v) = S(u,v) + P(u,v)\frac{n}{|n|}$$

where P(u,v) is a perturbation function in the direction of the normal to the original surface. The new normal vector is: $n' = S'_u \times S'_v$

# Texture

## Bump mapping

The perturbed normal can be written as:

$$n' = n + \frac{P_u\,(n \times S_v)}{|n|} + \frac{P_v\,(S_u \times n)}{|n|}$$

normal of the unperturbed surface

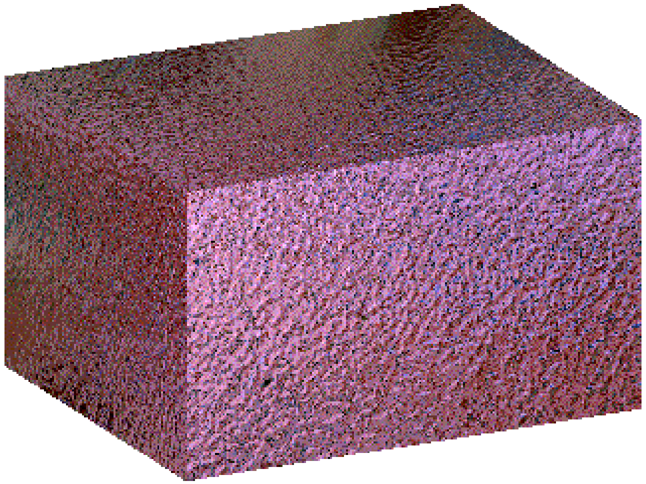
effect of the perturbation on the surface normal (hence, on the illumination model)



The perturbation P(u,v) can be defined either analytically or as a lookup table.

Different effects can be achieved:

random function gives rough surface



smoother functions give more regular feature

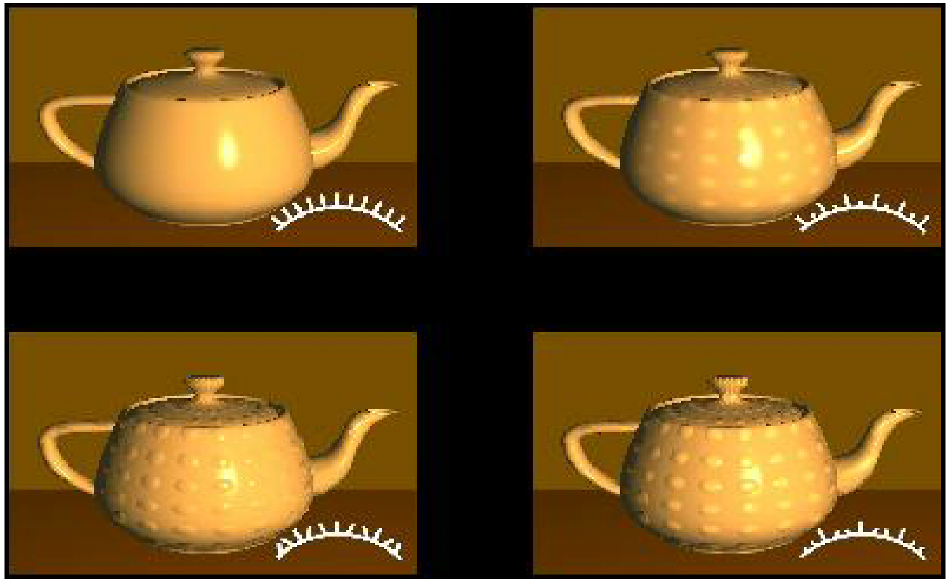
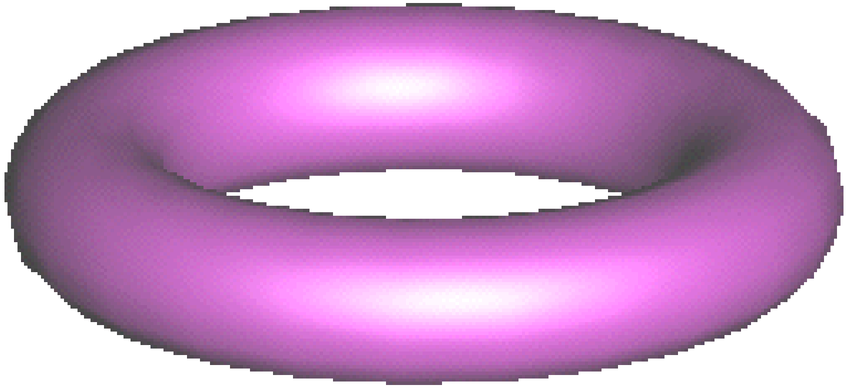


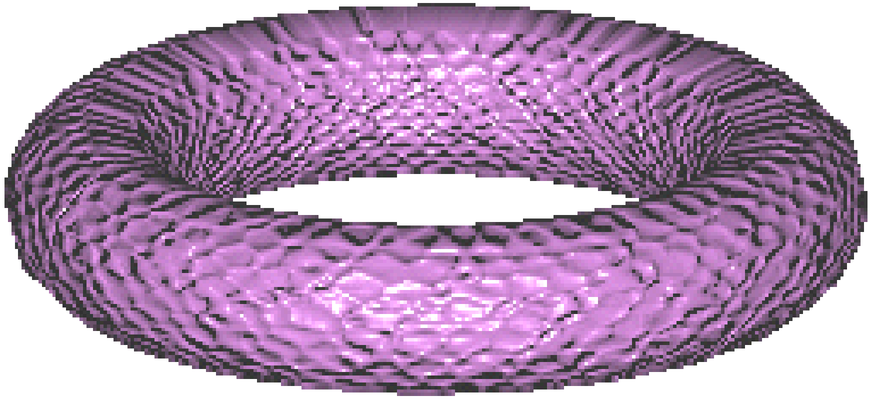
*Image from: © SIGGRAPH'97 R.J. Wolfe (DePaul University)*

# Texture

## *Bump mapping*

An example:





Note that: *Roughening only becomes apparent when the shading model is applied*

Examples of use:

- the surface of an orange
- texture of a granitic stone
- granulated effects
- outer cover of a tyre
- etc...