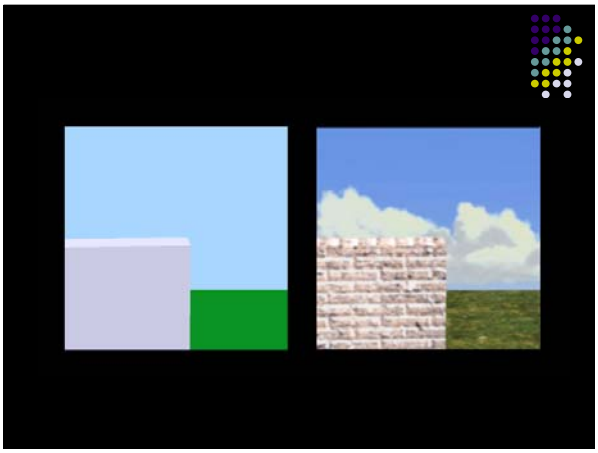
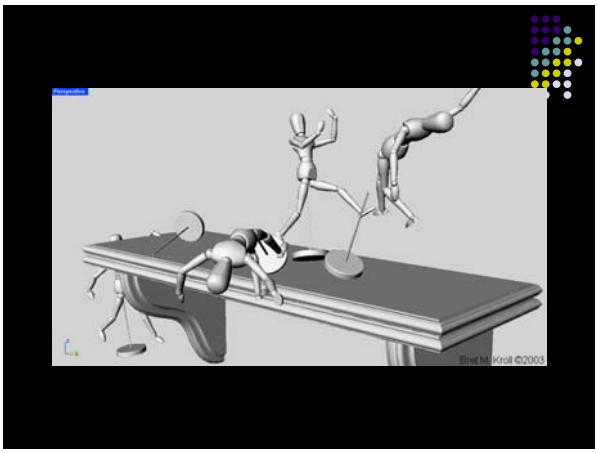
Texture Mapping

Advantages

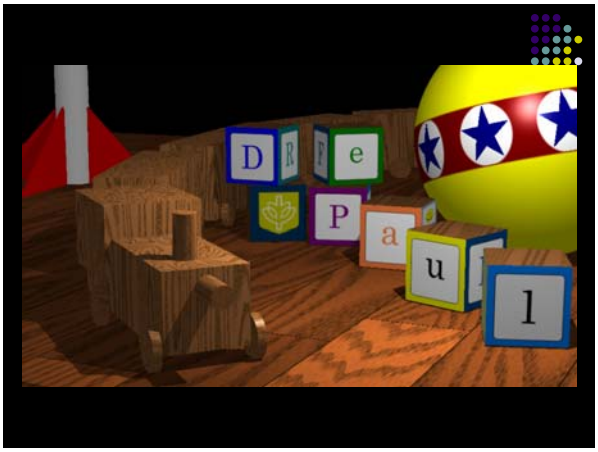
- Adds interest, detail
- Realism
- Cost

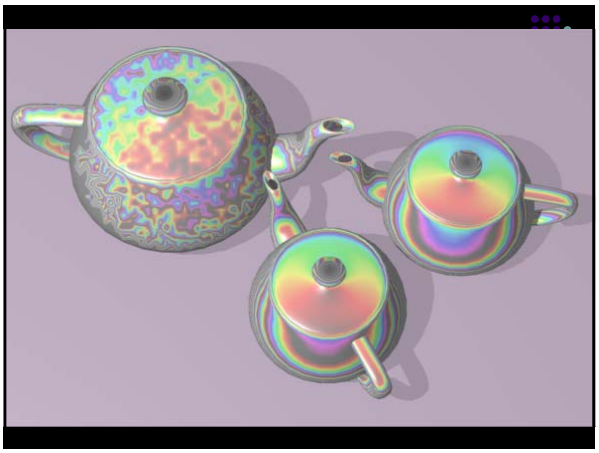




Visual Cues

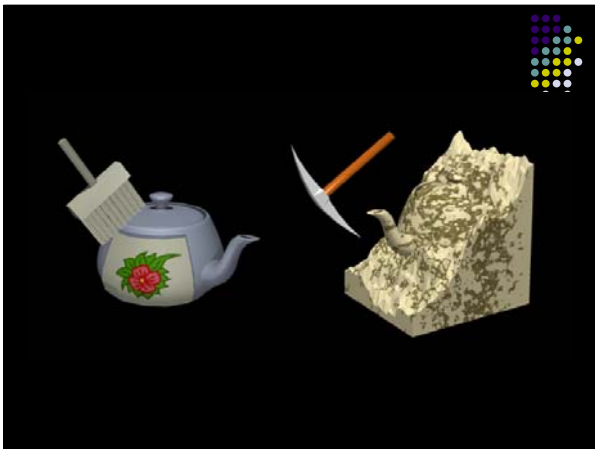
- Images, patterns “glued” onto object
- Wood, stone
- Repeating patterns
- Mottling


D
e
P
a
u
l

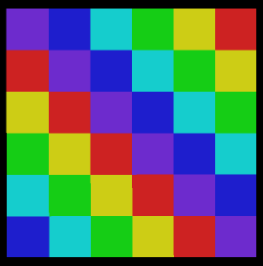


Two basic categories

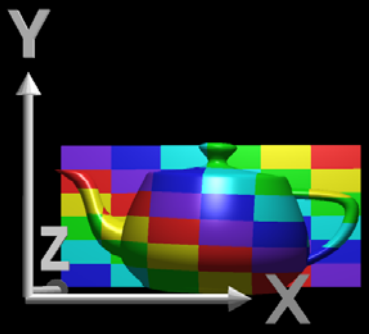
- 2D (image-based) texture mapping
- 3D texture mapping

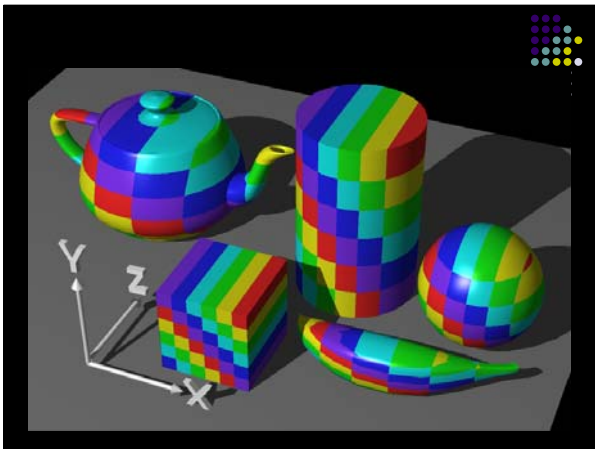


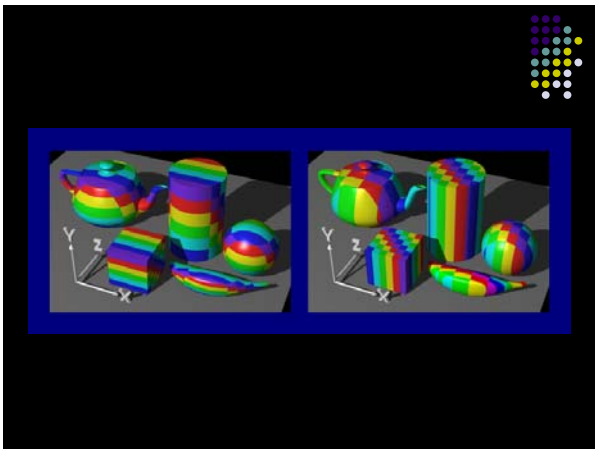
Our pattern



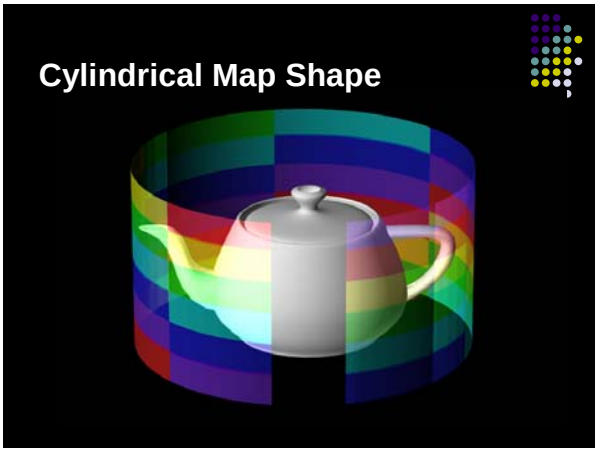
Planar map shape

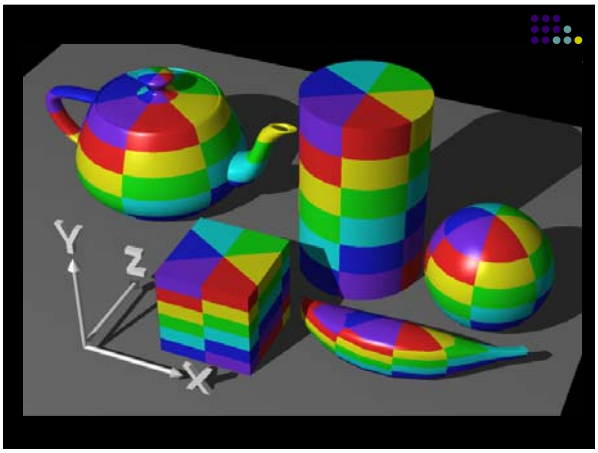


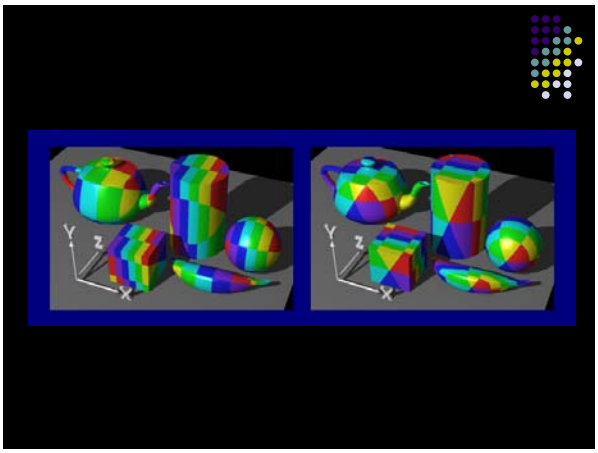




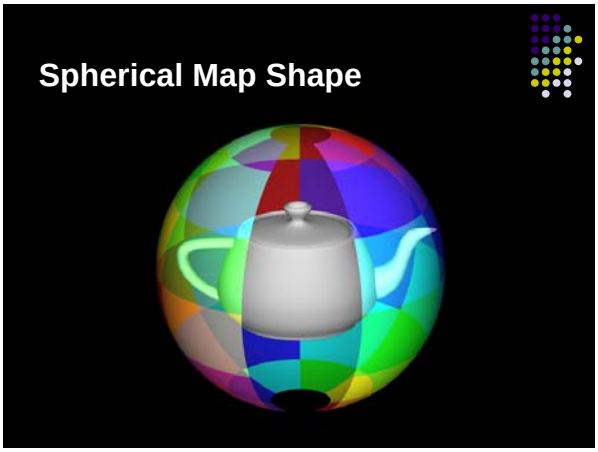
Cylindrical Map Shape

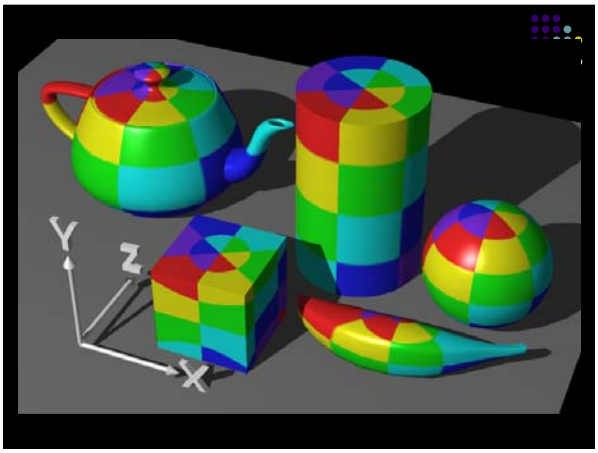




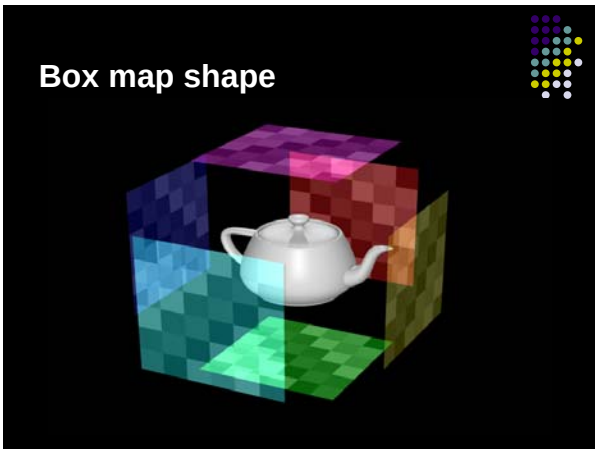


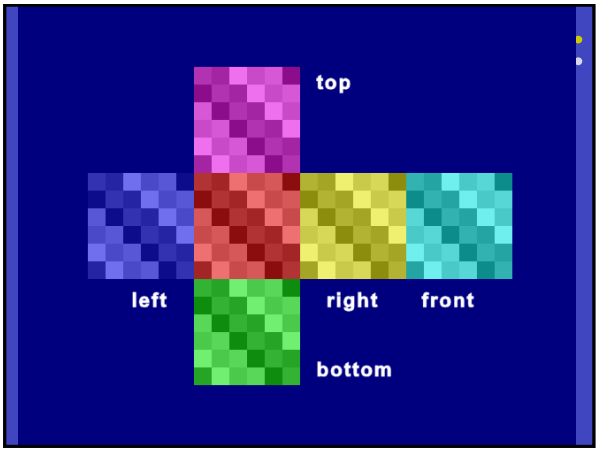
Spherical Map Shape

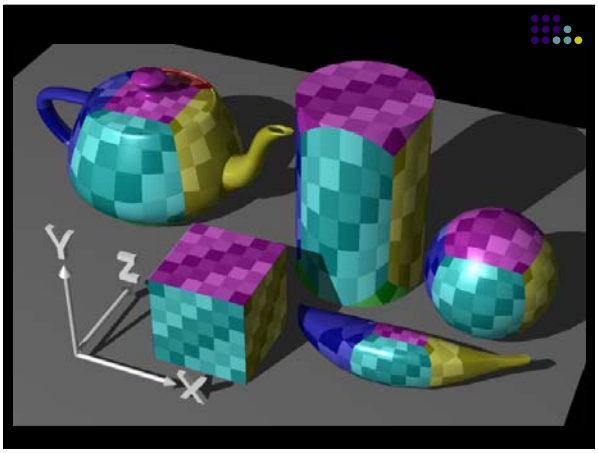


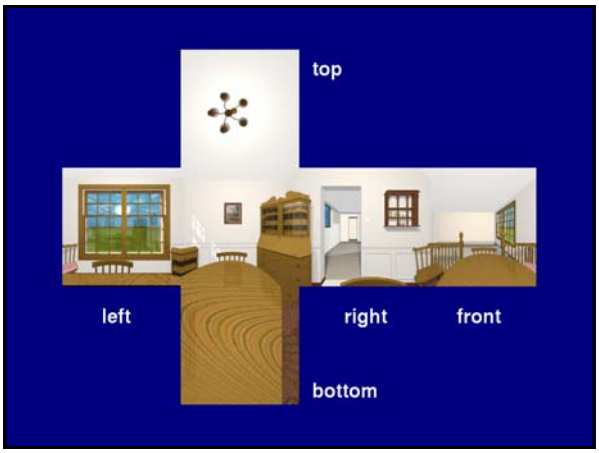


Box map shape









Texture Mapping in POV-Ray

Pigment statement

- Defines the color or pattern of colors for an object

```
pigment {
  [PIGMENT_IDENTIFIER]
  [PIGMENT_TYPE]
  [PIGMENT_MODIFIER...]
}
PIGMENT_TYPE:
  PATTERN_TYPE | COLOR |
  image_map { BITMAP_TYPE "bitmap.ext"
    [IMAGE_MAP_MODS...] }
```

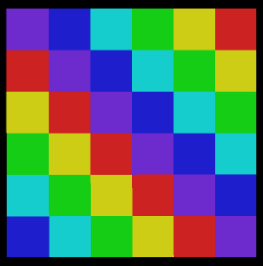
Image_map

- wraps a 2-D bit-mapped image around your 3-D objects

```
pigment {
  image_map
  {
    [BITMAP_TYPE] "bitmap.ext"
    [IMAGE_MAP_MODS...]
  }
  [PIGMENT_MODIFIERS...]
}
```

BITMAP_TYPE: gif | tga | iff | ppm | pgm | png | jpeg | tiff | sys

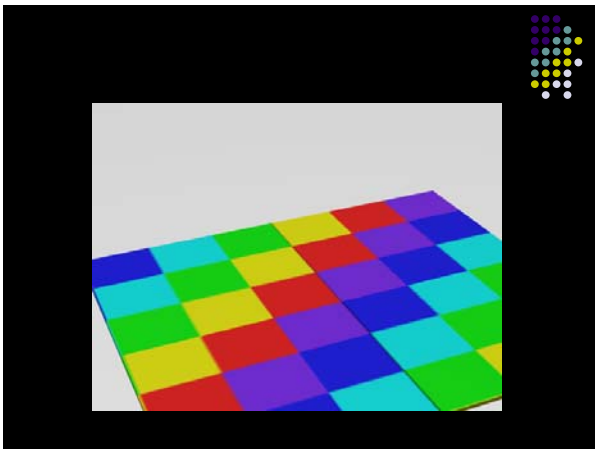
Our pattern: “ourpat.jpg”



Modifiers

- **`map_type 0`**
  - Planar map.
  - Image is projected onto x-y plane.
  - The image fills a square in (x,y) coordinates from (0,0) to (1,1)

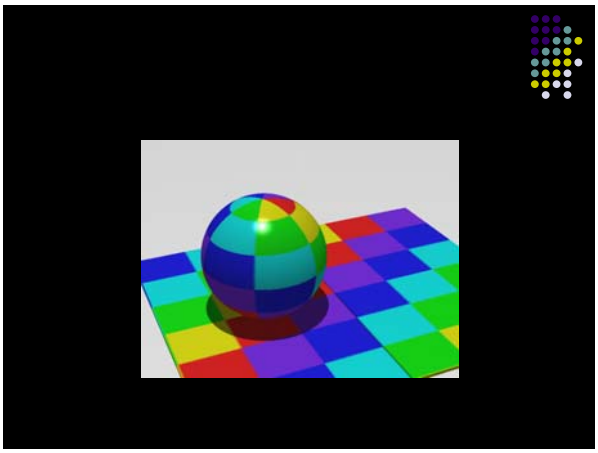
```
// table top
box{<-1,-0.1,-1>, <1,0, 1>
    scale <6, 1, 5>
    translate <2, 0, 1.5>
    texture{
        pigment{
                image_map { sys "ourpat.bmp"
                            map_type 0
              }
              quick_color red 0.8 green 0.4 blue 0
        }
        finish{diffuse 0.6 ambient 0.4 }
    }
}
```



Modifiers

- **`map_type 1`**
  - Spherical map.
  - Center is at origin.
  - Y-axis is texture axis.
  - Top of texture is a “North Pole”.
  - Texture starts at x-axis; goes west to east

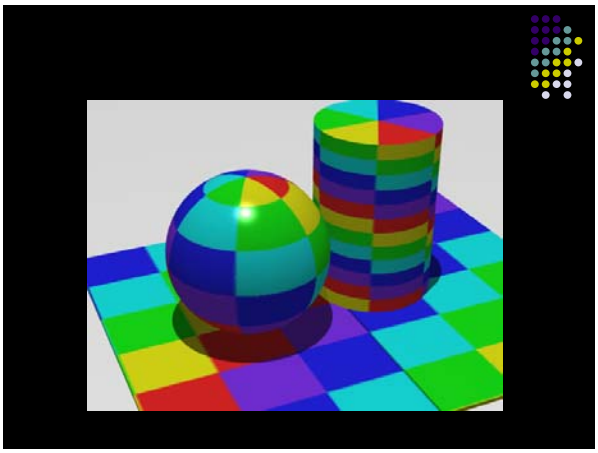
```
sphere{<0, 0, 0>, 1
    texture{
        pigment
        {
            image_map { sys "ourpat.bmp"
                        map_type 1
                      }
        }
        finish
        {
            diffuse 0.5
            ambient 0.5
            specular 1.0
            roughness 0.005
        }
    }
    scale<2.,2.,2.>
    translate<-0.5, 2., 1.62>
}
```



Modifiers

- **`map_type 2`**
  - Cylindrical map.
  - Center is at origin.
  - Y-axis is texture axis.
  - Texture starts at x-axis; goes west to east.
  - Bottom of image is at y = 0; top of image at y = 1

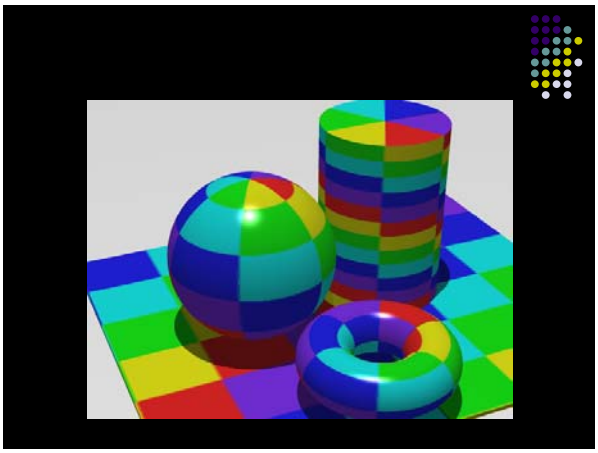
```
cylinder{ <0,-1, 0>, <0, 1, 0>, 1
    texture{
        pigment{
            image_map { sys "ourpat.bmp"
                        map_type 2
                }
        }
        finish{
            diffuse 0.5
            ambient 0.5
            specular 1.0
            roughness 0.005
        }
    }
    scale<1.6, 2.33, 1.6>
    translate<3.22, 2.33, 2.11>
}
```



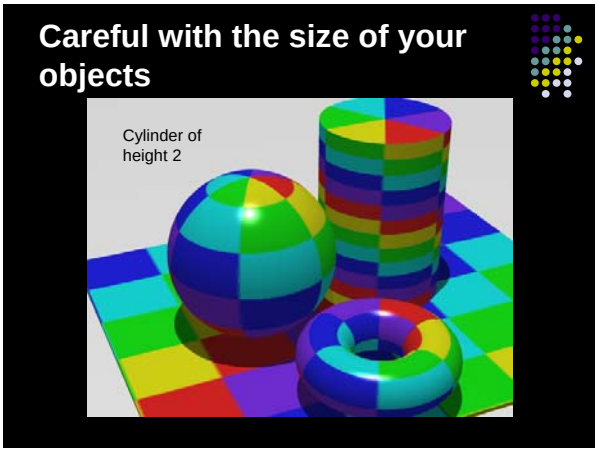
Modifiers

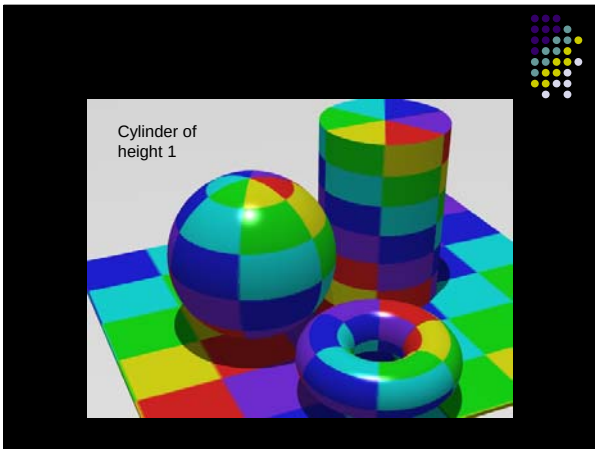
- **`map_type 5`**
  - Torus map.
  - Center is at origin.
  - It assumes that a torus of major radius one sits at the origin in the x-z-plane.
  - The image is wrapped around similar to spherical or cylindrical maps.
  - However the top and bottom edges of the map wrap over and under the torus where they meet each other on the inner rim

```
torus{1., .5
     texture{
              pigment {image_map { sys "ourpat.bmp"
                                   map_type 5
                       }
              }
              finish{
                    diffuse 0.5
                    ambient 0.5
                    specular 1.0
                    roughness 0.005
              }
     }
     scale 1.2
     translate <1,1,-2>
}
```



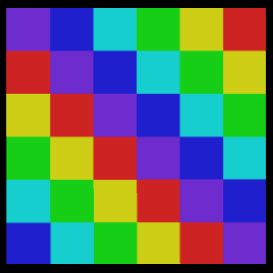
Careful with the size of your objects



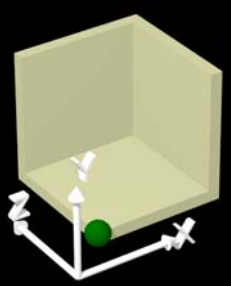


Example

Texture

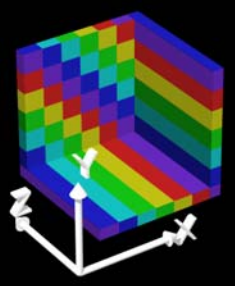


Scene



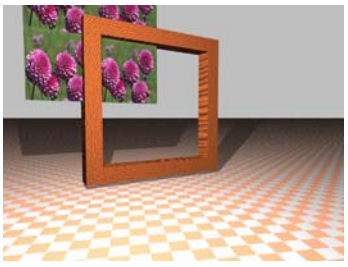
```
box{<0,-0.1,0>, <1,0, 1>
    pigment {
      image_map {"ourpat.bmp"  map_type 0}
    }
    finish {ambient .7 diffuse .3}
}
box {<0,-0.1, 1>, <1,1,1.1>
    pigment {
      image_map {"ourpat.bmp"  map_type 0}
    }
    finish {ambient .7 diffuse .3}
}
box {<1,-0.1, 0>, <1.1,1, 1.1>
    pigment {
      image_map {"ourpat.bmp"  map_type 0}
    }
    finish {ambient .7 diffuse .3}
}
```

What happened here?



Repetition, Scaling, Translating

- can specify “once” in image_map
- can scale, rotate, translate image_map



Transformations and texture

```
Object myObject{
   texture {…}
   rotate <…>
   translate <…>
}
```

versus

```
Object myObject{
      rotate <…>
      translate <…>
      texture {…}
}
```

Object Maps

```
pigment {
 object {
   … object … // text, etc.
   color rgb <1,0,0>,      // outside object
   color rgb <0,0,1>       // inside object
 }
```

Example