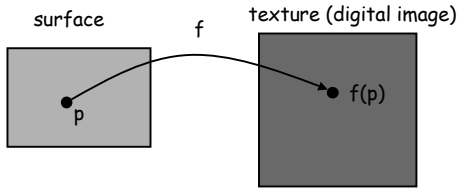
cheap tricks

- texture mapping
- procedural texture mapping
- bump mapping
- transparency mapping
- cut-out mapping
- depth of field
- lens effects
- jittering
- soft shadows

texture mapping

See cs155 gallery for examples

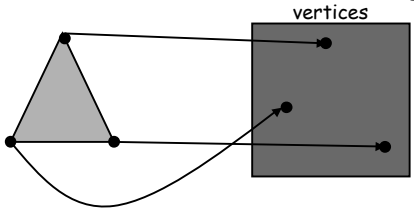
texture mapping



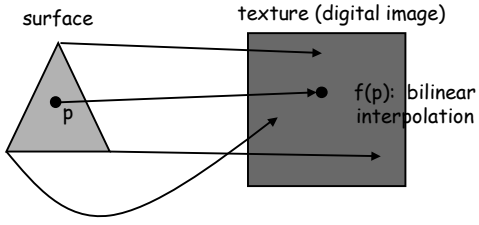
f(p): coordinates in the texture map corresponding to surface point p

texture mapping triangle

Input specifies texture coordinates of triangle vertices

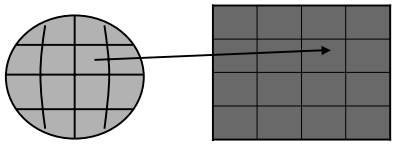


texture mapping triangle

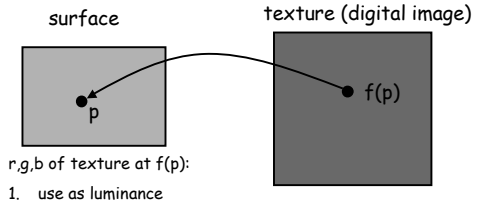


texture mapping sphere

e.g. latitude/longitude



texture mapping



r,g,b of texture at f(p):

1. use as luminance
2. use as diffuse and spectral coefficient
3. use as diffuse coefficient

cheap tricks

- texture mapping
- **procedural texture mapping**
- bump mapping
- transparency mapping
- cut-out mapping
- depth of field
- lens effects
- jittering
- soft shadows

procedural texture mapping

See gallery for example

procedural texture mapping

- create 3d "noise" procedure
- sample to find texture for surface

advantages

- don't need to find a mapping from a (complex) 3d surface to a 2d texture image
- concise representation of texture

invented by Ken Perlin: see his web site at NYUfor more details

cheap tricks

- texture mapping
- procedural texture mapping
- **bump mapping**
- transparency mapping
- cut-out mapping
- depth of field
- lens effects
- jittering
- soft shadows

bump mapping

See gallery for example

bump mapping intuition

a surface appears to have a coarse texture
- jagged silhouette
- surface normals fluctuate across surface

simulate this by randomly perturbing normals in the lighting calculations

bump mapping intuition

a surface appears to have a coarse texture
- jagged silhouette
- surface normals fluctuate across surface

Perturbing normals can also create some illusion of this

cheap tricks

- texture mapping
- procedural texture mapping
- bump mapping
- **transparency mapping**
- cut-out mapping
- depth of field
- lens effects
- jittering
- soft shadows

Texture specifies transparency of surface

cheap tricks

- texture mapping
- procedural texture mapping
- bump mapping
- transparency mapping
- **cut-out mapping**
- depth of field
- lens effects
- jittering
- soft shadows

Texture specifies "existence" of surface

cheap tricks

- texture mapping
- procedural texture mapping
- bump mapping
- transparency mapping
- cut-out mapping
- **depth of field**
- lens effects
- jittering
- soft shadows

blur based on distance from viewpoint

cheap tricks

- texture mapping
- procedural texture mapping
- bump mapping
- transparency mapping
- cut-out mapping
- depth of field
- **lens effects**
- jittering
- soft shadows

See paper mentioned in assignment

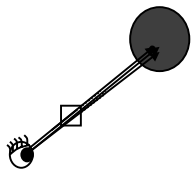
cheap tricks

- texture mapping
- procedural texture mapping
- bump mapping
- transparency mapping
- cut-out mapping
- depth of field
- lens effects
- **jittering**
- soft shadows

jittering: antialiasing technique

Run rt for example

jittering: anti-aliasing technique



- cast several ray through pixel neighborhood into scene
- for each:
  - find intersection point (if any) that is closest to eye
  - compute luminance at intersection
- compute average luminance

cheap tricks

- texture mapping
- procedural texture mapping
- bump mapping
- transparency mapping
- cut-out mapping
- depth of field
- lens effects
- jittering
- **soft shadows**

soft shadows

See gallery for example

soft shadows

use jittering in occlusion test

ray tracing

- simple ray casting
- recursive ray tracing
- modeling transforms
- cheap tricks
- **optimizations**

ray tracing complexity

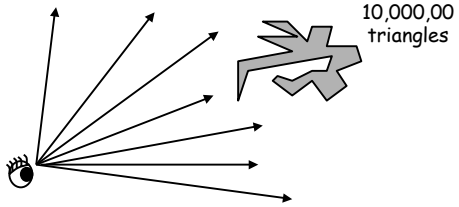
O(# of intersection tests)
= O(#pixels x # objects)

Can we reduce the number of intersection tests?

optimization

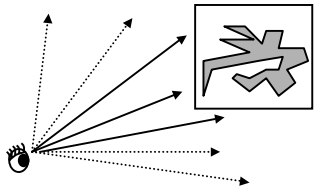
- bounding boxes
- oct-trees
- BSP-trees

bounding boxes: intuition



most rays miss the object

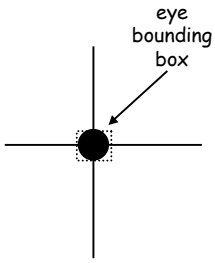
bounding boxes



1. rule out rays by simple intersection test with bounding box
2. perform exhaustive test on remaining rays

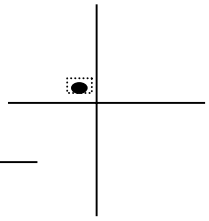
bounding boxes & hierarchical coordinates

body xfm
body description
  head translate wrt body
  head rotate
  head description
    {eye1 translate wrt head
    eye1 scale
    eye1 description}
    {eye2 translate wrt head
    eye2 scale
    eye2 description}



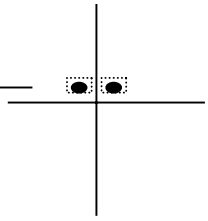
hierarchical coordinates

body xfm
body description
  head translate wrt body
  head rotate
  head description
    {eye1 translate wrt head
    eye1 scale
    eye1 description}
    {eye2 translate wrt head
    eye2 scale
    eye2 description}



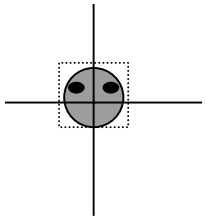
hierarchical coordinates

body xfm
body description
  head translate wrt body
  head rotate
  head description
    {eye1 translate wrt head
    eye1 scale
    eye1 description}
    {eye2 translate wrt head
    eye2 scale
    eye2 description}



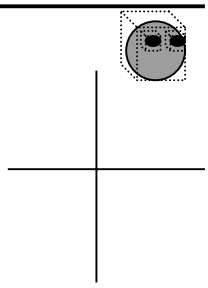
hierarchical coordinates

body xfm
body description
  head translate wrt body
  head rotate
  head description
    {eye1 translate wrt head
    eye1 scale
    eye1 description}
    {eye2 translate wrt head
    eye2 scale
    eye2 description}



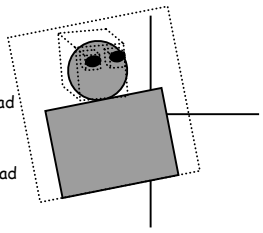
hierarchical coordinates

body xfm
body description
  head translate wrt body
  head rotate
  head description
    {eye1 translate wrt head
    eye1 scale
    eye1 description}
    {eye2 translate wrt head
    eye2 scale
    eye2 description}



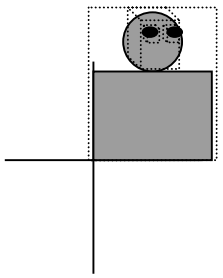
hierarchical coordinates

body xfm
body description
  head translate wrt body
  head rotate
  head description
    {eye1 translate wrt head
    eye1 scale
    eye1 description}
    {eye2 translate wrt head
    eye2 scale
    eye2 description}



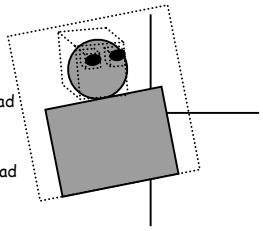
hierarchical coordinates

body xfm
body description ←
    head translate wrt body
    head rotate
    head description
        {eye1 translate wrt head
        eye1 scale
        eye1 description}
        {eye2 translate wrt head
        eye2 scale
        eye2 description}



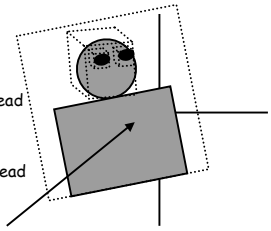
hierarchical coordinates

body xfm ←
body description
    head translate wrt body
    head rotate
    head description
        {eye1 translate wrt head
        eye1 scale
        eye1 description}
        {eye2 translate wrt head
        eye2 scale
        eye2 description}



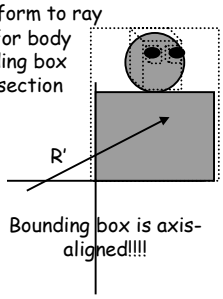
intersection

body xfm ←
body description
    head translate wrt body
    head rotate
    head description
        {eye1 translate wrt head
        eye1 scale
        eye1 description}
        {eye2 translate wrt head
        eye2 scale
        eye2 description}



intersection

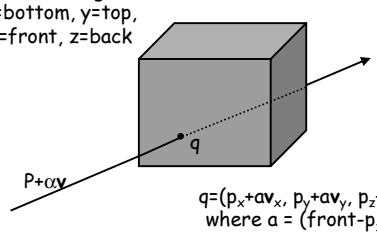
body xfm ←Apply inverse transform to ray
body description ← Test for body bounding box intersection
    head translate wrt body
    head rotate
    head description
        {eye1 translate wrt head
        eye1 scale
        eye1 description}
        {eye2 translate wrt head
        eye2 scale
        eye2 description}



Bounding box is axis-aligned!!!!

bounding box intersection

x=left, x=right,
y=bottom, y=top,
z=front, z=back



$q=(p_x+av_x, p_y+av_y, p_z+av_z)$
where $a = (front-p_z)/v_z$

optimization

- bounding boxes
- oct-trees
- BSP-trees

We won't cover these. Look on the web for details.