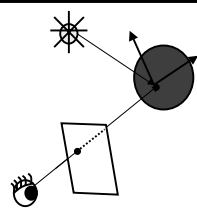


ray tracing

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

2/23/2003 CS155 - Ray Tracing 84

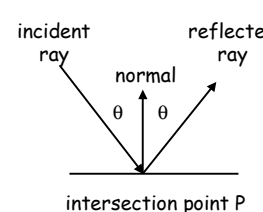
ray tracing



- cast ray through pixel into scene
- find closest intersection (if any)
- compute luminance at intersection
 - direct illumination
 - **reflections**
 - **refraction**

2/23/2003 CS155 - Ray Tracing 85

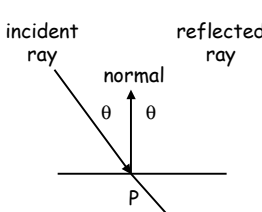
specular reflections



- cast ray reflected at P into scene
- find closest intersection point P' (if any)
- compute luminance at P'
- scale by $msr[g,b]$ and add to luminance at P

2/23/2003 CS155 - Ray Tracing 86

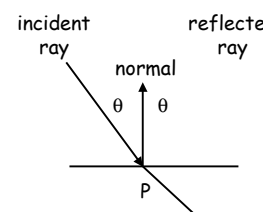
transmission



- cast ray transmitted at P into scene
- find closest intersection point P' (if any)
- compute luminance at P'
- scale by k_{trans} and add to luminance at P

2/23/2003 CS155 - Ray Tracing 87

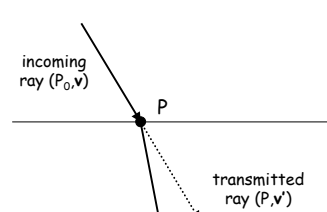
transmission



- **cast ray transmitted at P into scene**
- find closest intersection point P' (if any)
- compute luminance at P' point
- scale by k_{trans} and add to luminance at P

2/23/2003 CS155 - Ray Tracing 88

refraction - snell's law



2/23/2003 CS155 - Ray Tracing 89

thin surface refraction

ignore Snell's law

incoming ray (P_o, v)

transmitted ray (P, v)

2/23/2003 CS155 - Ray Tracing 90

thick surface refraction

incoming ray (P_o, v_{in})

transmitted ray (P, v_{out})

refractive index n_{in}

refractive index n_{out}

θ_{out} satisfies: $n_{out} \sin \theta_{out} = n_{in} \sin \theta_{in}$

$v_{out} = [\beta \cos \theta_{in} - (1 - \beta^2 \sin^2 \theta_{in})^{1/2}] n + \beta v_{in}$ where $\beta = n_{in} / n_{out}$

2/23/2003 CS155 - Ray Tracing 91

thick surface recursion

incoming ray (P_o, v)

transmitted ray (P, v)

inside

What is direct illumination at P' ?

2/23/2003 CS155 - Ray Tracing 92

stopping conditions

recurse until:

- maximum recursive depth specified by user is reached
- contribution to luminance is less than user specified bound

- cast new ray from P into scene
- find closest intersection point P' (if any)
- compute luminance at P'
- scale and add to luminance at P

2/23/2003 CS155 - Ray Tracing 93

implementation issues

offset new ray slightly to make sure you don't find P again!!!

- cast new ray from P into scene
- find **closest** intersection point P' (if any)
- compute luminance at P'
- scale and add to luminance at P

2/23/2003 CS155 - Ray Tracing 94

stopping conditions

recurse until:

- maximum recursive depth specified by user is reached
- contribution to luminance is less than user specified bound

- cast new ray from P into scene
- find closest intersection point P' (if any)
- compute luminance at P'
- scale by $msr/g/b$ or $ktrans$ and add to luminance at P

2/23/2003 CS155 - Ray Tracing 95

ray tracing

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

2/23/2003 CS155 - Ray Tracing 96

modeling transforms

object description: M, p or p' ?

2/23/2003 CS155 - Ray Tracing 97

ray tracing

- cast ray into scene
- **find intersection point (if any) that is closest to eye**
- compute luminance at intersection

2/23/2003 CS155 - Ray Tracing 98

find intersection point

viewpoint sphere

2/23/2003 CS155 - Ray Tracing 99

find intersection point

viewpoint squashed (aka transformed) sphere

2/23/2003 CS155 - Ray Tracing 100

find intersection point

object coordinates world coordinates

2/23/2003 CS155 - Ray Tracing 101

does this make sense?

- is there an inverse transform?
- how do we apply a transform to a ray?
- is a ray in world coordinates a ray in object coordinates?

2/23/2003

CS155 - Ray Tracing

102

Conceptually: scale

What operation inverts a scale by s in the x -direction?

For $s \neq 0$, scale by $1/s$ in the x -direction.

2/23/2003

CS155 - Ray Tracing

103

Any problem?

We are not alone!

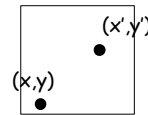
2/23/2003

CS155 - Ray Tracing

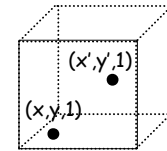
104

we are not alone...

the parallel universe view of homogenous coordinates



we live in this universe



it's not the only one, but it is the only one we can experience!

2/23/2003

CS155 - Ray Tracing

105

scale

$$\begin{pmatrix} s^{-1} & 0 & 0 & 0 \\ 0 & t^{-1} & 0 & 0 \\ 0 & 0 & u^{-1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s & 0 & 0 & 0 \\ 0 & t & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = ?$$

2/23/2003

CS155 - Ray Tracing

106

Conceptually: rotate

What operation inverts a rotate by θ about the x -axis?

Rotate by $-\theta$ about the x -axis.

2/23/2003

CS155 - Ray Tracing

107

rotate about z axis

$$\begin{pmatrix} \cos -\phi & -\sin -\phi & 0 & 0 \\ \sin -\phi & \cos -\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = ?$$

remember $\cos(-\phi) = \cos(\phi)$ and $\sin(-\phi) = -\sin(\phi)$

2/23/2003 CS155 - Ray Tracing 108

rotate about z axis

$$\begin{pmatrix} \cos \phi & \sin \phi & 0 & 0 \\ -\sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \phi & -\sin \phi & 0 & 0 \\ \sin \phi & \cos \phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = ?$$

remember $\cos^2 \phi + \sin^2 \phi = 1$

2/23/2003 CS155 - Ray Tracing 109

Conceptually: translate

What operation inverts a translate by dx in the x-direction?

Translate by -dx in the x-direction.

2/23/2003 CS155 - Ray Tracing 110

translate

$$\begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = ?$$

2/23/2003 CS155 - Ray Tracing 111

does this make sense?

- is there an inverse transform?
- how do we apply a transform to a ray?
- is a ray in world coordinates a ray in object coordinates?

2/23/2003 CS155 - Ray Tracing 112

transform

- Points Done!!
- Vectors
- Rays

2/23/2003 CS155 - Ray Tracing 113

transforms: vector

rectangle shown for reference!

$$v = p - q \text{ and } T(v) = T(p - q) = T(p) - T(q) = Mp - Mq$$

Warnings:

- because of translation we can't ignore $q = (0,0,0)$
- re-unitize unit vectors

2/23/2003 CS155 - Ray Tracing 114

transforms

- Points Done!!
- Vectors Done!!
- Rays

2/23/2003 CS155 - Ray Tracing 115

transforms: ray

- Points
- Vectors
- Rays: $r = (p, v)$

Transform point and transform/unitize vector!

2/23/2003 CS155 - Ray Tracing 116

does this make sense?

- is there an inverse transform?
- how do we apply a transform to a ray?
- is a ray in world coordinates a ray in object coordinates?

2/23/2003 CS155 - Ray Tracing 117

Linear transforms

Linear transforms preserve lines!

2/23/2003 CS155 - Ray Tracing 118

find intersection point

world coordinates M^{-1} local coordinates

intersection point p M intersection point p'

2/23/2003 CS155 - Ray Tracing 119

M and M⁻¹

single transform

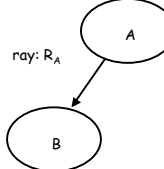
M	M⁻¹
- scale by s	- scale by 1/s
- rotate by θ	- rotate by $-\theta$
- translate by Δ	- translate by $-\Delta$

composite transform

$(M_1 M_2 \dots M_k)^{-1}$	$M_k^{-1} \dots M_2^{-1} M_1^{-1}$
----------------------------	------------------------------------

2/23/2003 CS155 - Ray Tracing 120

scene graph traversal

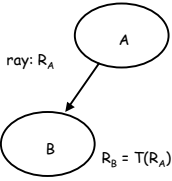


ray: R_A

A sends the ray (represented relative to A's coordinate system) to B.

2/23/2003 CS155 - Ray Tracing 121

scene graph traversal



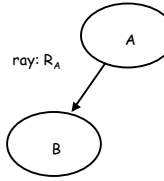
ray: R_A

B converts the ray into its own coordinate system

$R_B = T(R_A)$

2/23/2003 CS155 - Ray Tracing 122

scene graph traversal

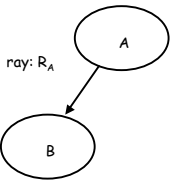


ray: R_A

B computes the intersections of R_B with its objects

2/23/2003 CS155 - Ray Tracing 123

scene graph traversal




ray: R_A

- B sends R_B to its descendents
- Each returns intersection information (represented in B's coordinate system)

2/23/2003 CS155 - Ray Tracing 124

surface normal

is the normal to a transformed surface the transformed normal?



2/23/2003 CS155 - Ray Tracing 125

surface normal

is the tangent plane to a transformed surface the transformed tangent plane?

2/23/2003 CS155 - Ray Tracing 126

The right way ...

N is normal to the tangent plane iff for any points p and q on the tangent plane $N^T \bullet (p-q) = 0$.

2/23/2003 CS155 - Ray Tracing 127

The right way ...

N is normal to the tangent plane iff for any points p and q on the tangent plane $N^T \bullet (p-q) = 0$.

Assume N is normal to the tangent plane and QN is normal to the tangent plane transformed by M.

Q must satisfy the following for any points p and q on the tangent plane:

$$N^T(p-q)=0 \text{ iff } (QN)^T(M(p-q))=0$$

⇕

$$N^T(p-q)=0 \text{ iff } N^T(Q^T M)(p-q)=0$$

Thus $Q=(M^{-1})^T$

2/23/2003 CS155 - Ray Tracing 128

ray tracing

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

2/23/2003 CS155 - Ray Tracing 129