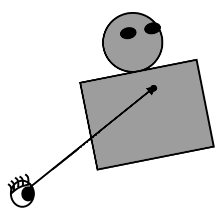
cs155 – z sweedyk

ray tracing

ray tracing

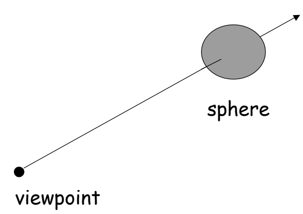
- ray casting
  - rays
  - intersection tests
  - **intersection with scene graph**
  - lighting and material properties
- recursive ray tracing
- cheap tricks
- optimizations

ray tracing



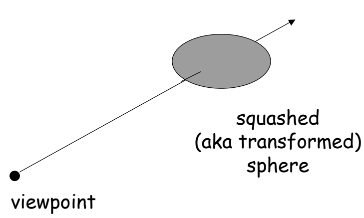
- cast ray into scene
- **find intersection point (if any) that is closest to eye**
- compute luminance at intersection

find intersection point

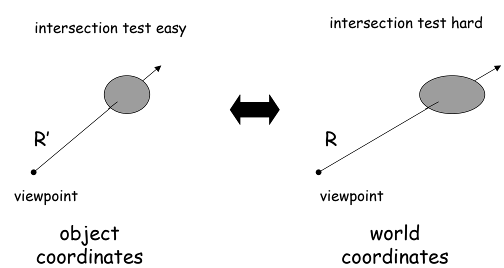


find intersection point

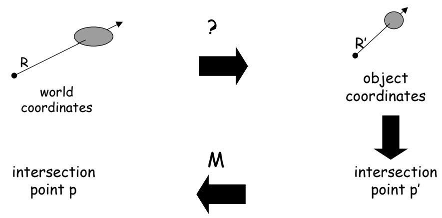
intersection test hard!



find intersection point



find intersection point



does this make sense?

- is there an inverse transform M^{-1} for points?

Conceptually: scale

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

Conceptually: scale

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

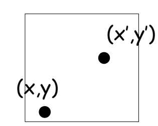
For $s \neq 0$, scale by $1/s$ in the x-direction.

Any problem?

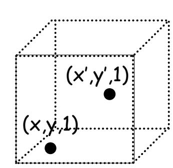
We are not alone!

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!

scale

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

Conceptually: rotate

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

Conceptually: rotate

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

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

rotate about z axis

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

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

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

Conceptually: translate

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

Conceptually: translate

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

Translate by -dx in the x-direction.

translate

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

does this make sense?

- is there an inverse transform M^{-1} for points?

YES!

To invert composite $M_1 M_2 \dots M_n$ use $M_n^{-1} \dots M_2^{-1} M_1^{-1}$

does this make sense?

- is there an inverse transform for points?

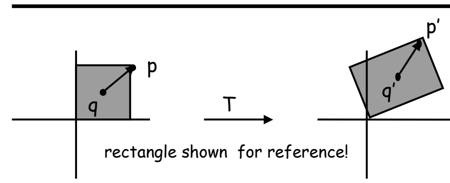
YES!

- how do we apply a transform to a ray?

R=(p,v) T(R) = (T(p),T(v))

How about this?

transforming vectors

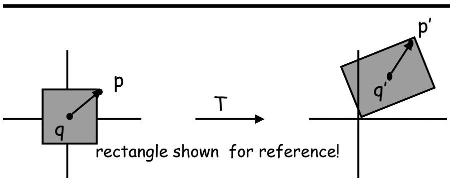


rectangle shown for reference!

v=p-q and T(**v**)=T(p-q)=T(p)-T(q)

don't worry about homogeneous coordinates until here

transforming vectors (tricky part)



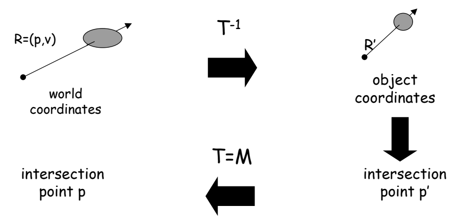
rectangle shown for reference!

v=p-q where q=(0,0,0,1)

and T(**v**)=T(p-q)=T(p)-T(q)

if translation involved then T(q) $\neq$ (0,0,0,1)

find intersection point



$T^{-1}(p) = M^{-1}p$, $T^{-1}(v)=(M^{-1}v-M^{-1}(<0,0,0,1>))\,)/||(M^{-1}v-M^{-1}(<0,0,0,1>))||$

ray tracer skeleton has built in call for this

M and M^{-1}

single transform

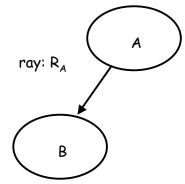
M
- scale by s
- rotate by θ
- translate by Δ

M^{-1}
- scale by 1/s
- rotate by $-\theta$
- translate by $-\Delta$

composite transform

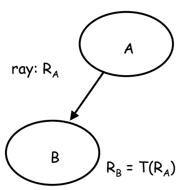
$(M_1 M_2 \ldots M_k)^{-1}$ $\quad$ $M_k^{-1} \ldots M_2^{-1} M_1^{-1}$

scene graph traversal



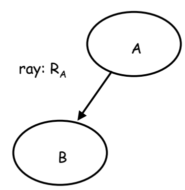
A sends the ray (represented relative to A's coordinate system) to B.

scene graph traversal



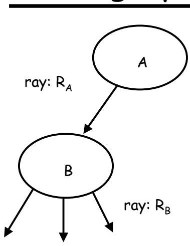
B converts the ray into its own coordinate system

scene graph traversal



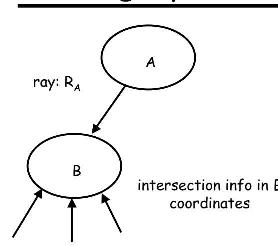
B computes the intersections of R_B with its objects

scene graph traversal



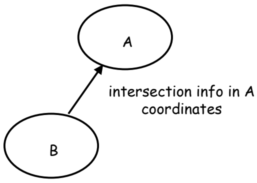
- B sends R_B to its children

scene graph traversal



- Each child returns its intersection info in B's coordinate system

scene graph traversal



- B computes intersection closest to viewer, converts it to A's coordinate system and returns the info to A

Intersection info

- distance between intersection point and viewer
- point of intersection
- normal at point of intersection
- material properties of surface
- etc.

Intersection info

- distance between intersection point and viewer
- point of intersection
- normal at point of intersection
- **material properties of surface**
- **etc.**

no problem

Intersection info

- distance between intersection point and viewer
- **point of intersection**
  - **$M^{-1}p$ where M is the transform from A to B and p is the closest intersection point in B's coordinates**
- normal at point of intersection
- material properties of surface
- etc.

Intersection info

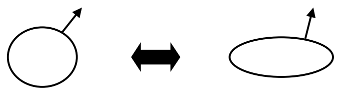
- **distance between intersection point and viewer**
  - **distance= $|| M^{-1}p - p_0||$ where $R_A=(p_0,v)$**
- point of intersection
- normal at point of intersection
- material properties of surface
- etc.

Intersection info

- distance between intersection point and viewer
- point of intersection
- **normal at point of intersection**
- material properties of surface
- etc.

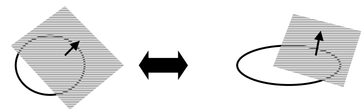
surface normal

is the normal to a transformed surface the transformed normal?



surface normal

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



The right way …

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

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:

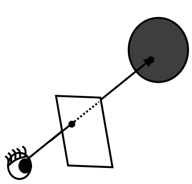
$$N^T(p-q)=0 \text{ iff } (QN)^T(M(p-q))=0$$

$$\Updownarrow$$

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

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

ray casting



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