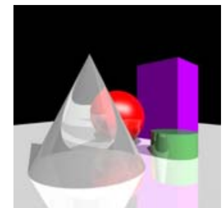
cs155 – z sweedyk

ray tracing

ray tracing

```
<scene>
  <cone material="glass">
  <sphere color="red">
  <box color="purple">
  <floor material="marble">
</scene>
```

→

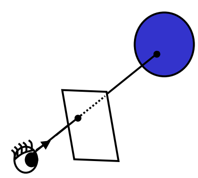


peter henry

ray tracing

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

ray casting



1. cast ray through pixel into scene
2. find intersection point (if any) that is closest to eye
3. compute luminance at intersection

ray specification

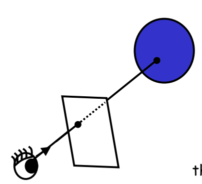
a ray is a half-line defined by a point and a unit vector

(p,v)

starting point → p

direction (unit vector) → v

ray casting

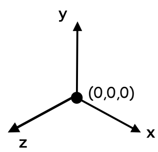


1. cast ray through pixel into scene

Construct the ray that starts at the viewpoint and goes through pixel i,j.

world coordinates

in the beginning there was (0,0,0) and x, y, z



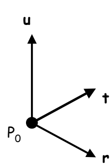
input scene specification

- Viewpoint position P_0 and orientation **t**, **u**

⋮

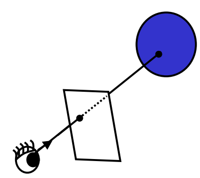
viewpoint specification: P_0, **u**, **t**

3d world coordinates



- P_o is the viewpoint
- **u**, **t**, and **r**=**t**×**u** are orthogonal unit vectors in the up, toward, and right directions

ray casting

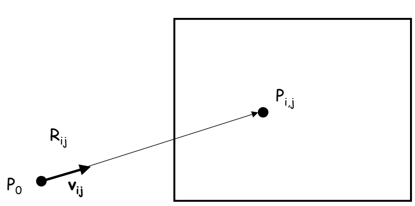


1. cast ray through pixel into scene

Construct the ray that starts at the viewer and goes through pixel i,j.

(P_0, ____)

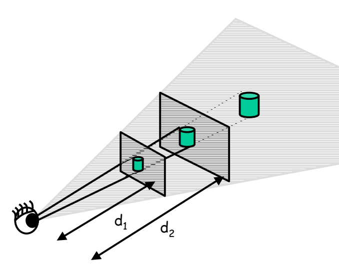
casting rays



R_{ij} = (P_0, $\mathbf{v}_{ij}$) where $\mathbf{v}_{ij}$ =($P_{i,j}$ – P_0)/|| $P_{i,j}$ – P_0||

Problem: Compute P_{ij} in *world coordinates*.

Where are the pixels in world coordinates?



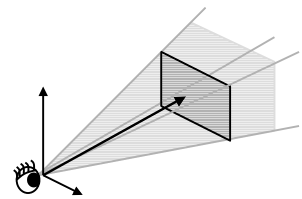
Don't worry … it won't matter!

input scene specification

- Viewpoint position P_0 and orientation **t, u**
- Image width *w* and height *h* in pixels
- Half-height angle θ

⋮

view volume



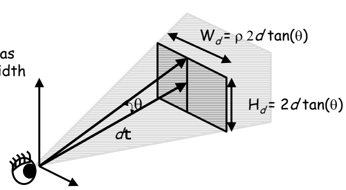
The world that is visible to the viewer is called the *view volume.*

The view volume is a pyramid that is aligned with the toward, up, and right vectors.

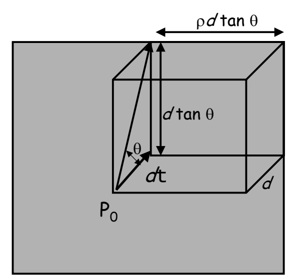
view volume specification: θ, ρ

The view volume is specified by the half-height angle θ and the aspect ratio ρ=w/h.

At a distance *d* from the viewer, the *view window* has height $H_d = 2d\tan(\theta)$ and width $W_d = \rho H_d = \rho\, 2d\tan(\theta)$.



pixel coordinates



What are the coordinates of the center of the view window at distance *d*?

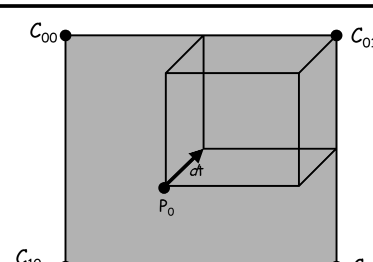
Ans: $P_0 + d\mathbf{t}$

point + vector

p + **v** is the point you get to by walking along the vector **v** from point p.

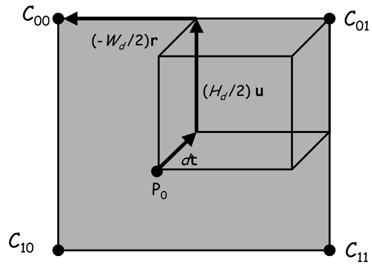
$q = p + v = (p_x+v_x, p_y+v_y, p_z+v_z)$

corner coordinates

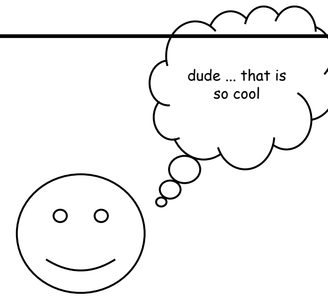


What are the coordinates of C_{00}

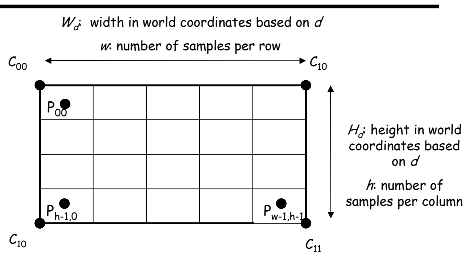
corner coordinates



$C_{00} = P_0 + d\mathbf{t} + (H_d/2)\,\mathbf{u} - (W_d/2)\,\mathbf{r}$



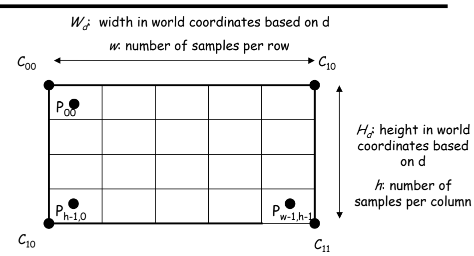
pixel coordinates



What are the height and width of a pixel "square" in world coordinate?

Ans: H_d/h and W_d/w

pixel coordinates



$P_{ij} = C_{00} + (i+\tfrac{1}{2})(W_d/w)\ \mathbf{r} - (j+\tfrac{1}{2})\,(H_d/h)\ \mathbf{u}$

choosing d

$P_{ij} = C_{00} + (i+\tfrac{1}{2})(W_d/w)\ \mathbf{r} - (j+\tfrac{1}{2})\,(H_d/h)\ \mathbf{u}$

Let's choose d so that $H_d = h$.

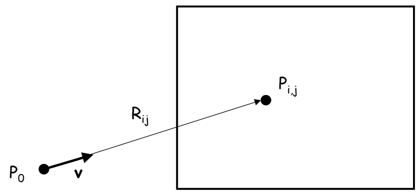
So $d = h/(2\tan(\theta))$.

Then $W_d = \rho H = \rho h = w$.

specification

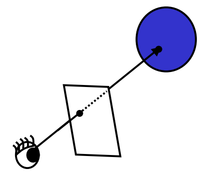
- Input specification
  - Viewpoint position P_0 and orientation $\mathbf{t}$, $\mathbf{u}$
  - Image width w and height h in pixels
  - Half-height angle θ
- Compute
  - Right vector $\mathbf{r} = \mathbf{t} \times \mathbf{u}$
  - Aspect ratio $\rho = w/h$
  - $d = h/(2\tan\theta)$
  - $H_d = h$, $W_d = w$

casting rays



$R_{ij} = (P_0, \mathbf{v}_{ij})$ where $\mathbf{v}_{ij} = (P_{i,j} - P_0)/||P_{i,j} - P_0||$

ray casting



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

intersection

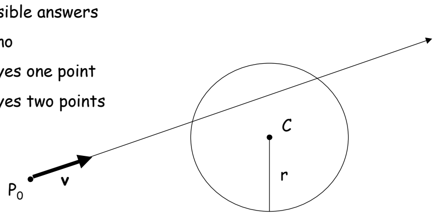
- sphere
- triangle
- box
- cylinder
- cone
- torus

intersection

- **sphere**
- triangle
- box
- cylinder
- cone
- torus

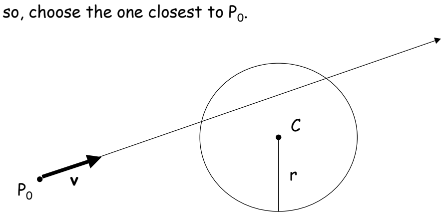
sphere intersection

- Is there a point that lies on the sphere and on the ray (P_0,**v**)?
- Possible answers
  - no
  - yes one point
  - yes two points



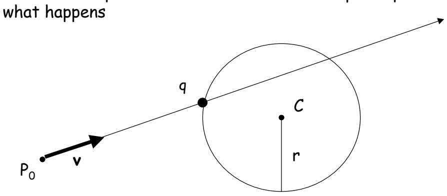
sphere intersection

- Is there a point that lies on the sphere and on the ray (P_0,**v**)?
- If so, choose the one closest to P_0.



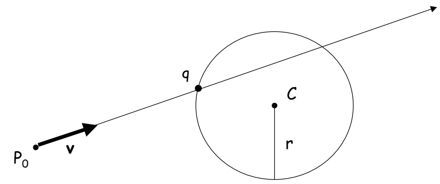
sphere intersection

- Is there a point that lies on the sphere and on the ray (P_0,**v**)?
- To answer this question we'll assume there is a point q and see what happens



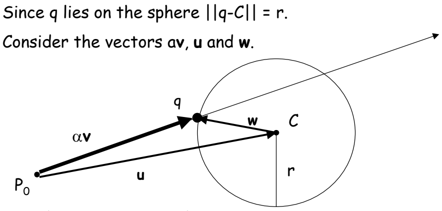
sphere intersection

- Since q lies on R, there is some $\alpha \geq 0$ such that $q = P_0 + \alpha \mathbf{v}$.
- Since q lies on the sphere $||q-C|| = r$.



sphere intersection

- Since q lies on R, there is some $\alpha \geq 0$ such that $q = P_0 + \alpha \mathbf{v}$.
- Since q lies on the sphere $||q-C|| = r$.
- Consider the vectors a**v**, **u** and **w**.



Then $\mathbf{w} = \alpha\mathbf{v} - \mathbf{u}$ and

$$r^2 = ||w||^2 = w \bullet w = (\alpha\mathbf{v} - \mathbf{u}) \bullet (\alpha\mathbf{v} - \mathbf{u})$$
$$= \alpha^2 ||\mathbf{v}||^2 - 2\alpha(\mathbf{u} \bullet \mathbf{v}) + ||\mathbf{u}||^2$$

sphere intersection

Does the ray intersect the sphere?

⬍

Does the quadratic (in α)

$$||\mathbf{v}||^2 \alpha^2 - 2(\mathbf{u} \bullet \mathbf{v})\alpha + ||\mathbf{u}||^2 - r^2$$

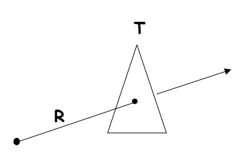
have any real, non-negative roots?

- No: no intersection
- Yes: intersection point is $q = P_0 + r\mathbf{v}$ where r is the smallest non-negative root.

intersection

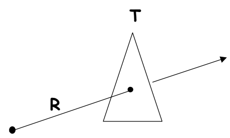
- sphere
- **triangle**
- box
- cylinder
- cone
- torus

triangle intersection



- do R and T intersect?
- if so, where?

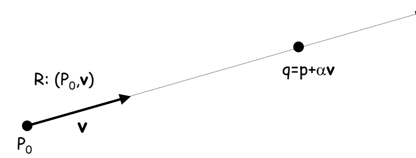
triangle intersection



1. Find the intersection point (if any) of R and the plane containing T.
2. Determine if the intersection point is inside the triangle.

ray: parametric form

a point q lies on R=(p,**v**) iff q=p+α**v** for some $\alpha \geq 0$



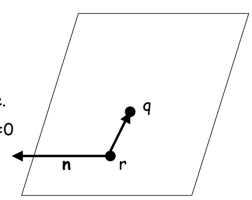
plane

Let **n** be a normal to the plane.

Let r be a point on the plane.

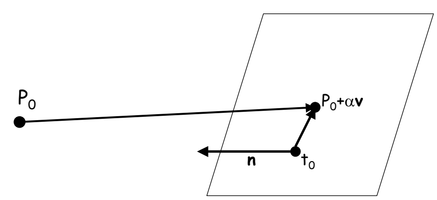
Let q be an arbitrary point in space.

Then q lies on the plane iff **n•**(q-r)=0



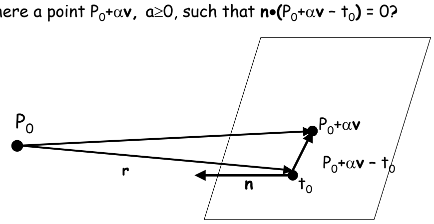
plane intersection

Is there a point P_0+α**v**, a≥0, such that **n•(**P_0+α**v** – t_0**)** = 0?



plane intersection

Is there a point P_0+α**v**, a≥0, such that **n•(**P_0+α**v** – t_0**)** = 0?



Assume there is and solve for α.

Then 0 = **n•(**P_0+α**v** – t_0**)** = **n•(**α**v** – **r)** where **r** is the vector from P_0 to t_0.

plane intersection

Is there a point P_0+α**v**, a≥0, such that **n•(**P_0+α**v** – t_0**)** = 0?

Is there an α such that **n•(**α**v** – **r)=0?**

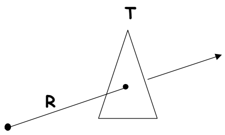
If **n•v**=0 then R is either parallel to plane or it lies in the plane.

If **n•v**≠0 then α= (**n•r**)/(**n•v**).

If $\alpha \geq 0$ then R intersects the plane at point P_0+α**v**.

If α<0 then R does not intersect plane.

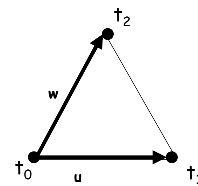
triangle intersection



1. Find the intersection point (if any) of R and the plane containing T.
2. Determine whether the point lies inside the triangle.

triangle: parametric form

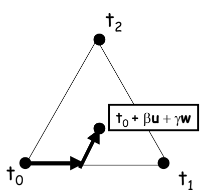
a point q lies on the *plane* containing the triangle T iff $q = t_0 + \beta\mathbf{u} + \gamma\mathbf{w}$ for some β and γ.



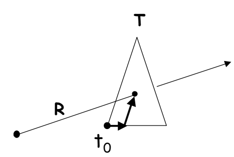
triangle: parametric form

a point q lies on the triangle T iff $q = t_0 + \beta\mathbf{u} + \gamma\mathbf{w}$ where

$\beta \geq 0$, $\gamma \geq 0$, $\beta + \gamma \leq 1$



triangle intersection



Find β, γ such that

$P_0 + \alpha\mathbf{v} = t_0 + \beta\mathbf{u} + \gamma\mathbf{w}$

$\beta, \gamma \geq 0$

$\beta + \gamma \leq 1$

triangle intersection

let $\mathbf{s} = P_0 + a\mathbf{v} - t_0$.

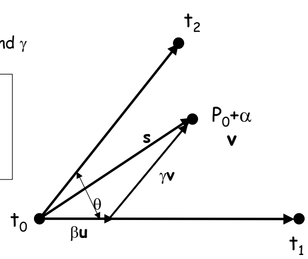
then $\mathbf{s} = \beta\mathbf{u} + \gamma\mathbf{w}$ for some β and γ

To find β and γ solve system of equations:

$\beta(\mathbf{u} \bullet \mathbf{w}) + \gamma(\mathbf{w} \bullet \mathbf{w}) = \mathbf{s} \bullet \mathbf{w}$

$\beta(\mathbf{u} \bullet \mathbf{u}) + \gamma(\mathbf{w} \bullet \mathbf{u}) = \mathbf{s} \bullet \mathbf{u}$

if $\beta, \gamma \geq 0$ and $\beta + \gamma \leq 1$ then the intersection point lies in the triangle T



intersection

- triangle
- sphere
- box
- cylinder
- cone
- torus

Etc. – you get to do these

ray tracing

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