Survival Kit for Project 2

1. Casting Rays
2. Computing Intersection Points
   - Triangle
   - Sphere
3. Computing color
4. Recursive ray tracing

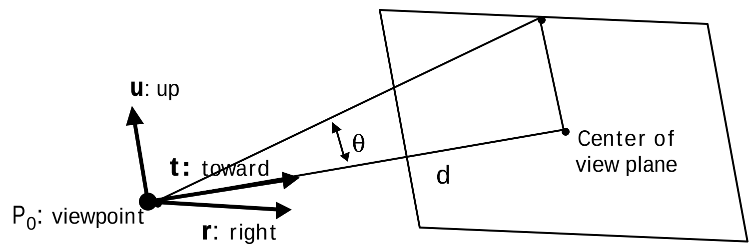
1

Survival Kit for Project 2

1. **Casting Rays**
2. Computing Intersection Points
   - Triangle
   - Sphere
3. Computing color (without shadows)

2

World View



P_o is the viewpoint

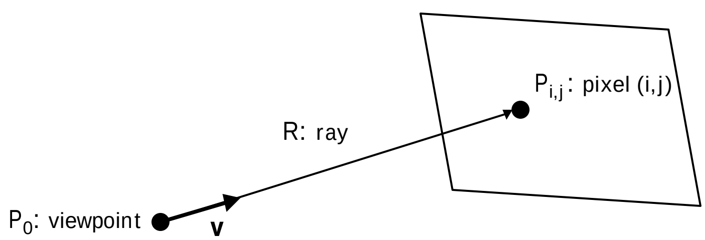
u, **t**, and **r** are unit vectors

d is the distance between P_0 and view plane

θ is the half-height angle of the frustum

3

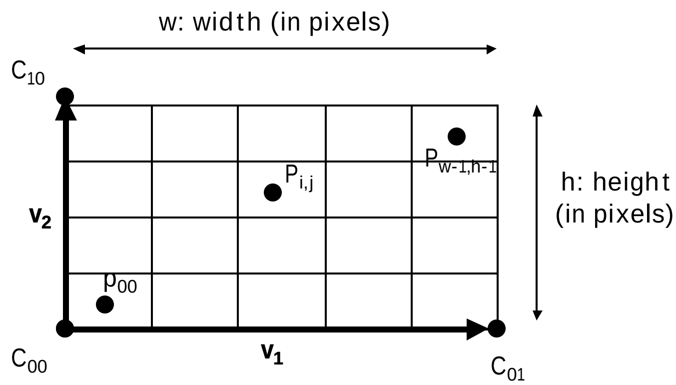
Casting Rays



Computing R: R= (P_0, **v**) where $\mathbf{v} = (P_{i,j} - P_0) / || P_{i,j} - P_0 ||$

4

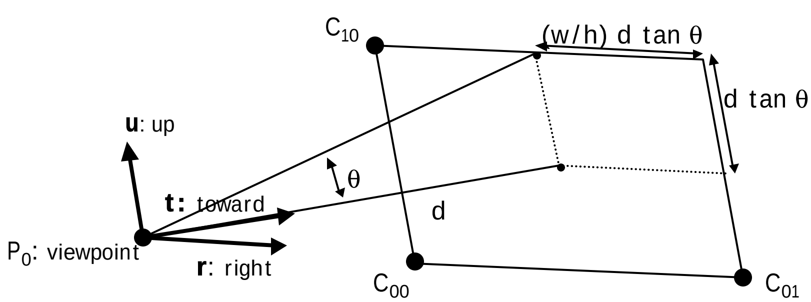
Casting Rays



Computing P_{ij} : $P_{i,j} = C_{00} + (i+0.5)/w \ \mathbf{v_1} + (j+0.5)/h \ \mathbf{v_2}$

5

Casting Rays



Computing C_{00}, C_{01}, C_{10} :

$C_{00} = P_0 + d\mathbf{t} - (d \tan\theta)\mathbf{u} - (w/h)(d \tan\theta)\ \mathbf{r}$

$C_{01} = P_0 + d\mathbf{t} - (d \tan\theta)\mathbf{u} + (w/h)\ (d \tan\theta)\ \mathbf{r}$

$C_{10} = P_0 + d\mathbf{t} + (d \tan\theta)\mathbf{u} - (w/h)(d \tan\theta)\ \mathbf{r}$

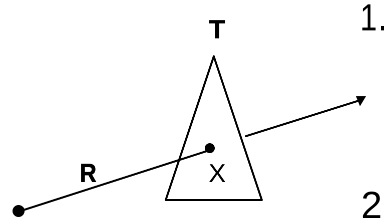
6

Survival Kit for Project 2

1. Casting Rays
2. **Computing Intersection Points**
   - **Triangle**
   - Sphere
3. Computing color

7

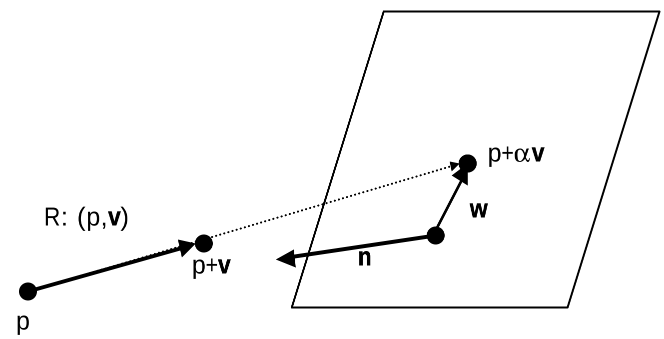
Intersection of R and T



1. Find intersection point X of R and the plane containing T
2. Check if X is inside T

8

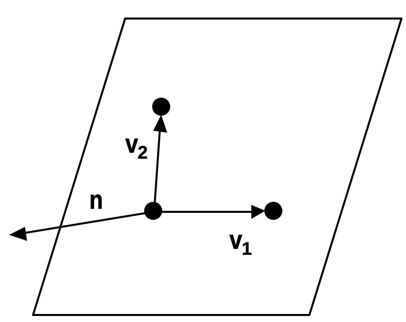
Intersection of a Ray and a Plane



Intersection point : X= p+α**v** where α is such that **w • n = 0**

9

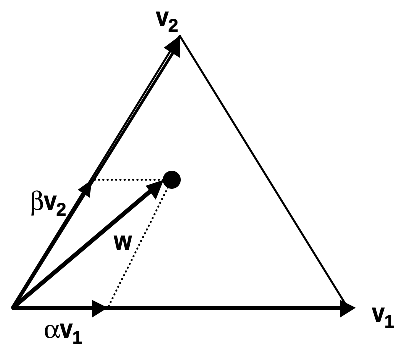
Normal to a Plane



$\mathbf{n} = \mathbf{v_1} \times \mathbf{v_2}$

10

Inside Test

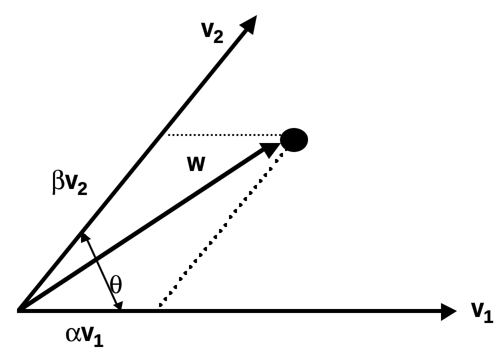


Find α and β such that $\mathbf{w} = \alpha\, \mathbf{v_1} + \beta\, \mathbf{v_2}$

Inside iff $\alpha \geq 0$, $\beta \geq 0$, and $\alpha+\beta \leq 1$

11

Inside Test



To find α and β solve:

$\alpha(\mathbf{v_1} \cdot \mathbf{v_1}) + \beta\,(\mathbf{v_2} \cdot \mathbf{v_1}) = \mathbf{w} \cdot \mathbf{v_1}$

$\alpha\,(\mathbf{v_1} \cdot \mathbf{v_2}) + \beta\,(\mathbf{v_2} \cdot \mathbf{v_2}) = \mathbf{w} \cdot \mathbf{v_2}$

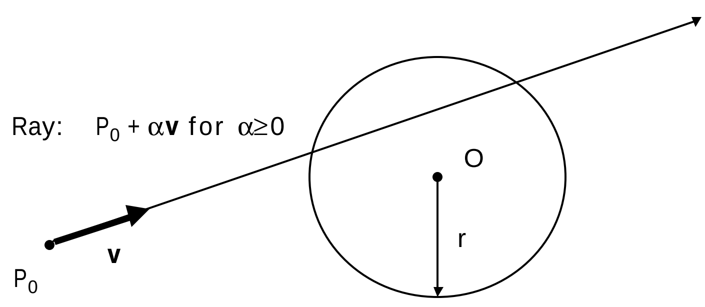
12

Survival Kit for Project 2

1. Casting Rays
2. **Computing Intersection Points**
   - Triangle
   - **Sphere**
3. Computing color (no shadows)

13

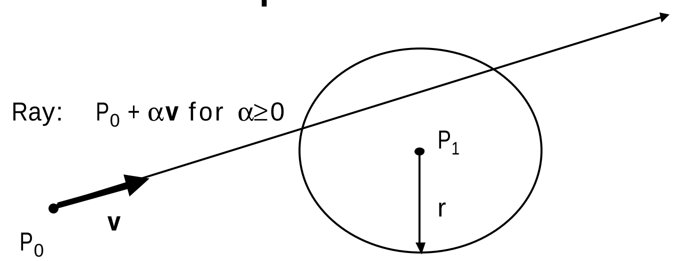
Intersection of a Ray and a Sphere



$P_0+\alpha\mathbf{v}$ is an intersection point if $|P_0 + \alpha\mathbf{v} - O|^2 = r^2$

14

Intersection of a Ray and a Sphere



$P_0 + \alpha\mathbf{v}$ is an intersection point if $\|\mathbf{w} + \alpha\mathbf{v}\|^2 = r^2$, where $\mathbf{w} = P_0 - P_1$

Solve for α in $A\alpha^2 + B\alpha + C = 0$ where

$A = \mathbf{v}\cdot\mathbf{v}$, $B = 2\,\mathbf{v}\cdot\mathbf{w}$, $C = \mathbf{w}\cdot\mathbf{w} - r^2$

15

Survival Kit for Project 2

1. Casting Rays
2. Computing Intersection Points
   - Triangle
   - Sphere
3. **Computing color (no shadows)**

16

Computing Color

(no shadows)

The color of a point on a surface depends on

- Lights in scene
- Material properties of surface
- Geometry of scene

17

Computing Color

(no shadows)

The contribution of each channel (r,g,b) is the result of five terms

- Ambient
- Diffuse
- Specular
- Emissive
- Transmission

18

Red Ambient

- The red ambient term is *r* * *ar* where
  - *r* is the intensity of the red ambient light. *r* is specified by the #ambient directive in the ray file.
  - *ar* is the response of the surface to red ambient light. *ar* is specified by the #material directive in the ray file .

19

Red Diffuse

- The red diffuse term is $\Sigma R_{L,D}$ where
  - The summation is taken over all lights L
  - $R_{L.D}$ is the intensity of the red, diffuse reflection of light L at the intersection point

20

$R_{L,D}$ for directional light L

$R_{L,D}$ = *dr* • *r* • max(0,(**n** · **d**)) where

1. *dr* is the response of the surface to red, diffuse light. *dr* is specified in the #material directive of the ray file.
2. *r* is the red intensity of light L. *r* is given in the #light_dir directive of the ray file.
3. **n** is the unit normal of the surface at the point of intersection.
4. **d** is the vector (-1) • <dx,dy,dz>/||<dx,dy,dz>>||. The direction vector <dx,dy,dz> is specified in the #light_dir directive of the ray file.

21

$R_{L,D}$ for point light L

$R_{L,D}$ = A • *dr* • *r* • max(0,(**n** · **p**)) where

1. *dr* , *r* , and **n** are defined in slide 19
2. **p** is the unit vector **from** the intersection point **to** the light position (*px,py,pz*). Position (*px,py,pz*) is given by the #light_point directive of the ray file.
3. A is the attenuation term 1/(*ca* + *la* • *dist* + *qa* * $dist^2$). *ca, la* and *qa* are specified by the #light_point directive of the ray file. *dist* is the distance between the intersection point and the light position (*px,py,pz*).

22

$R_{L,D}$ for spot light L

$R_{L,D}$ = A •SP • *dr* • *r* • max(0,(**n** · **d**))
where

1. A is defined in slide 21
2. *dr* , *r* , **n** and **d** are defined in slide 20
3. SP is the "spot light effect." (See Princeton precept notes.)

23

Red Specular

- The red specular term is $\Sigma R_{L,S}$ where
  - The summation is taken over all lights L
  - $R_{L.D}$ is intensity of the red, specular reflection of Light L at the intersection point

24

$R_{L,S}$ for directional light L

$R_{L,S} = sr \bullet r \bullet \max(0,(\mathbf{n} \cdot \mathbf{s}))^N$ where

1. *sr* is the response of the surface to red, specular light. *sr* is specified in the #material directive of the ray file.
2. *r* and **n** are defined as in slide 20.
3. *N* (not to be confused with **n** – sorry) is given (as “n”) in the #material directive of the ray file.
4. **s** is the vector of reflection. See Princeton precept notes.

25

$R_{L,S}$ for point light L

$R_{L,S} = A \bullet dr \bullet r \bullet \max(0,(\mathbf{n} \cdot \mathbf{s}))^N$ where

1. *dr* , *r* , and **n** are defined in slide 20
2. A is defined in slide 21
3. **s** and *N* are defined in slide 24

26

$R_{L,S}$ for spot light L

$R_{L,S}$ = A • SP • *dr* • *r* • max(0,(**n** · **s**)) ^{N}
where

1. A is defined in slide 21
2. *dr* , *r* , **n** are defined in slide 20
3. SP is defined in slide 22
4. **s** and *N* are defined in slide 23

27

Red Transmission

- The red light transmitted through the surface is R_{neg} • *ktran* where
  - *Ktran* is the transmission coeffecient given in the #material directive
  - R_{neg} is the light falling on the “backward facing intersection point” of the surface. To compute R_{neg} you repeat the lighting calculation but use **–n** as the normal to the surface.

28

Recursive Ray Tracing

- The radiosity of a surface point depends on
  - the direct contribution of the light sources
  - light reflected from other objects in the scene
  - light transmitted from other objects in the scene

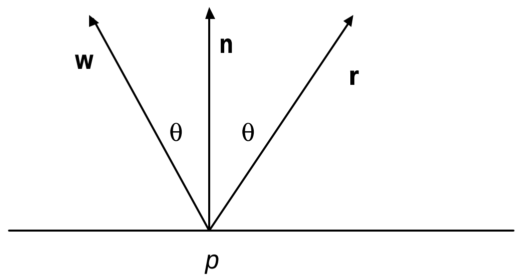
29

Red Recursive Reflections at point *p*, rDepth = m

- Recursively compute the red light along the ray (*p*,**r**) using rDepth m-1
  - **r** is the unit vector in the direction of mirror reflection
- Add the contribution scaled by s_r to the color at *p*.
  - s_r is specified by the #material directive

30

Mirror Reflection



w is the negative of the unit direction vector of the incoming ray

p is the point of intersection with unit normal **n**

r is the unit vector of mirror reflection:

$$\mathbf{r} = 2(\mathbf{n}\cdot\mathbf{w})\mathbf{n} - \mathbf{w}$$

31

Red Recursive Transmission at point p, rDepth = m

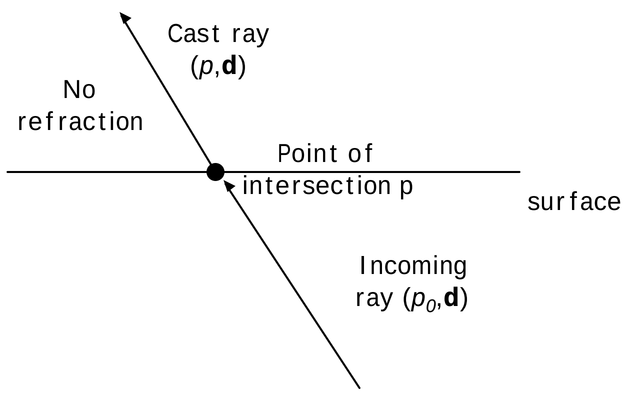
(ignoringSnell's law)

- Recursively compute the red light along the ray (*p*,**d**) using rDepth m-1
  - **d** is the unit vector in the direction of the incoming ray
- Add the contribution scaled by *ktran* to the color at *p*.
  - *ktran* is specified by the #material directive

32

Transmission

Ignoring Snell

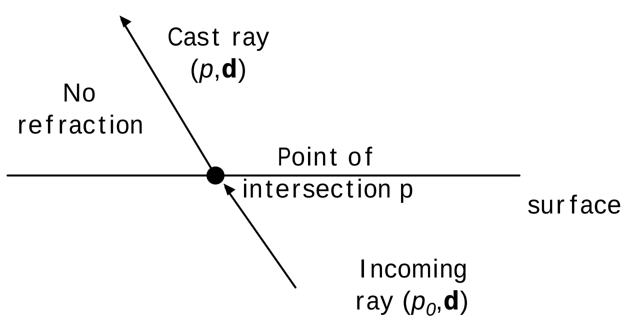


33

Thin SurfaceTransmission

with Snell

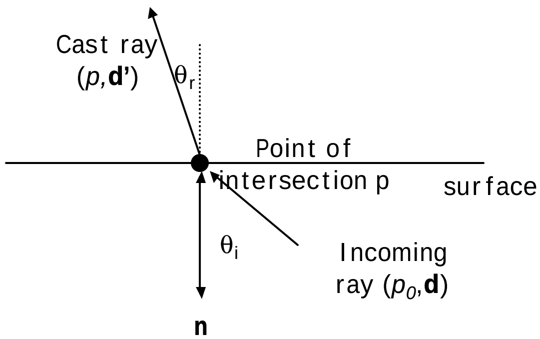
For "thin" (i.e. triangles) objects we can ignore Snell



34

Thick SurfaceTransmission

with Snell



θ_r satisfies: $\eta_r \sin \theta_r = \eta_i \sin \theta_i$

$\mathbf{d'} = ((\eta_i/\eta_r) \cos \theta_i - \cos \theta_r)\mathbf{n} - (\eta_i/\eta_r)\mathbf{d}$

35