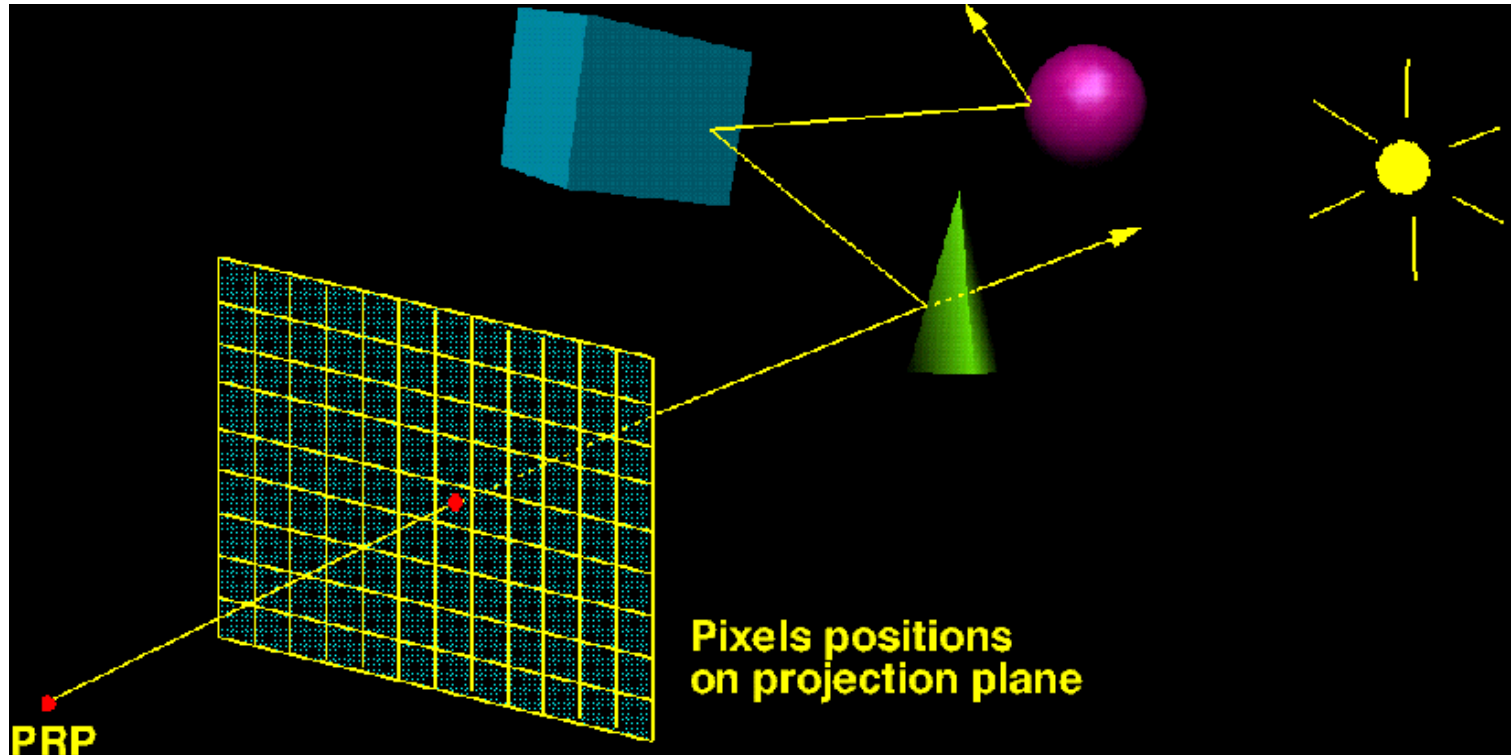


Sledenje žarka (Ray Tracing)



Sledenje žarka (ray tracing)

Metoda upodabljanja prizora.

Dobra za upodabljanje površin, ki so tako odbojne kot prosojne.

S sledenjem žarka lahko realiziramo **globalno osvetlitev**.

Sledenje žarka (kot ga poznamo danes), je razvil Turner Whitted (1980).

Sledenje žarka (ray tracing)

Na naš pogled prizora vplivajo le žarki svetlobe, ki od objektov prihajajo do kamere.

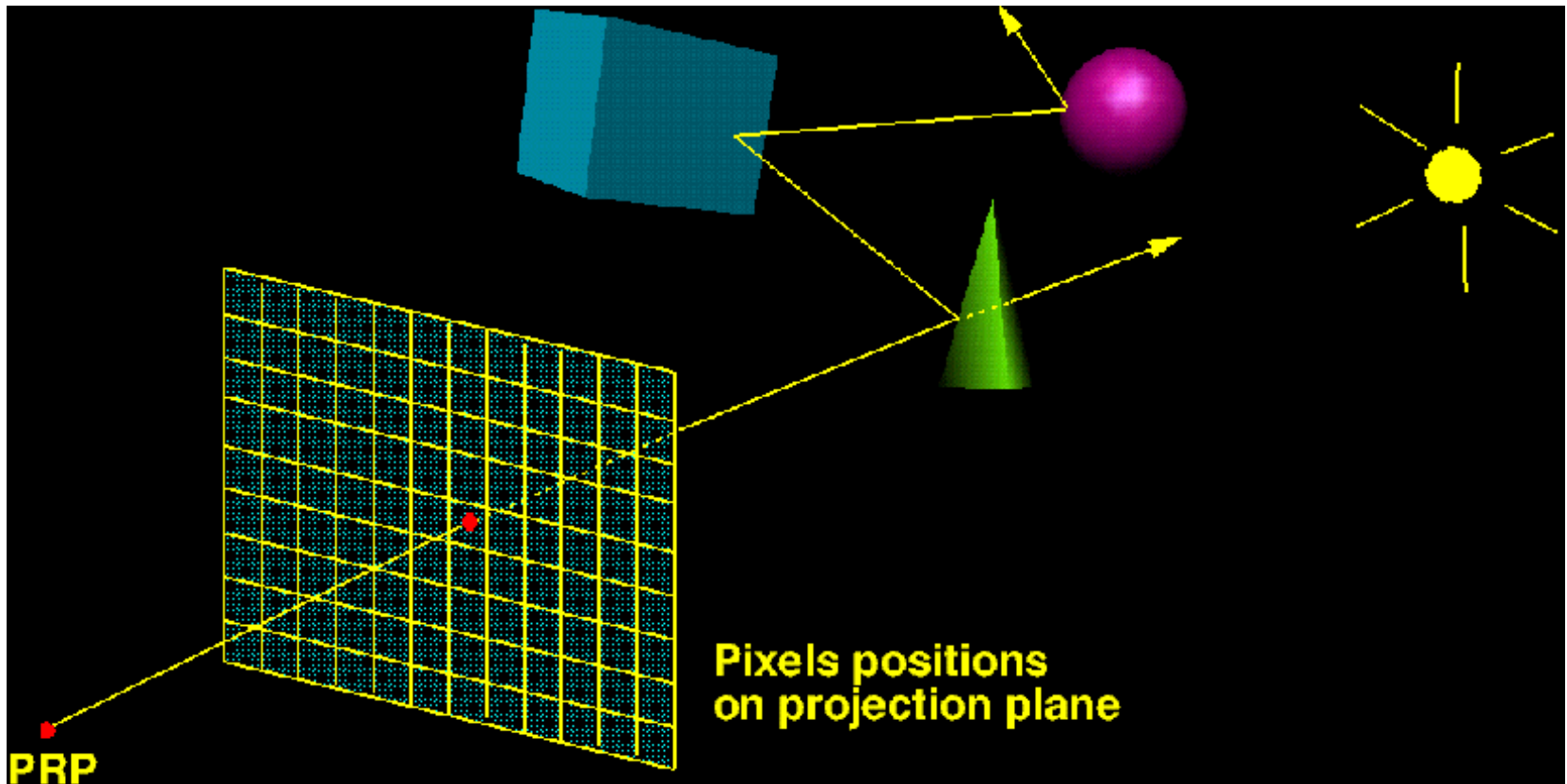
Žarek je definiran kot vektor v določeni začetni točki (kamera).

Računanje poti vseh žarkov svetlobe
je računsko kompleksno.

Večina žarkov iz virov svetlobe nikoli ne dosežejo kamere; zato računamo le pot žarkov,
ki dosežejo kamero.

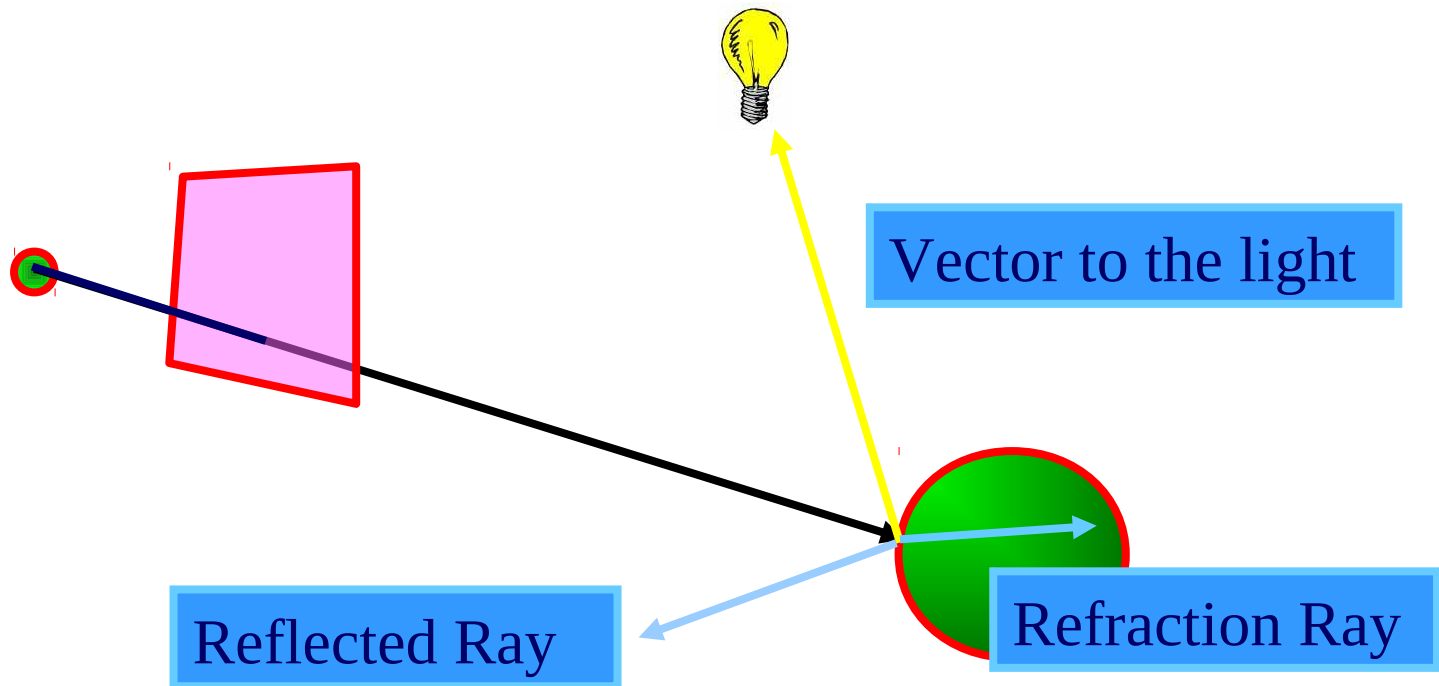
Sledenje žarka (ray tracing)

Računanje izvaja vzvratno. Začne pri kameri in ugotavlja kaj zadane drugi konec žarka.



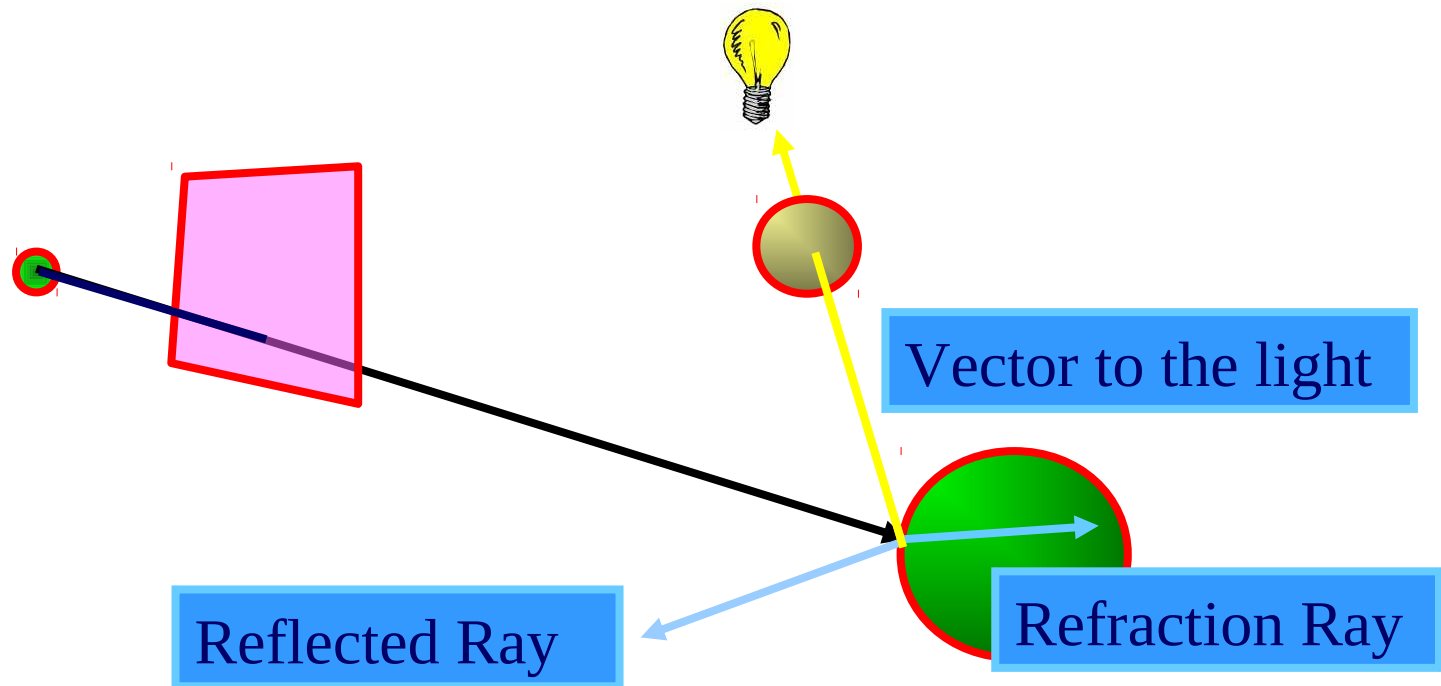
Rekurzivno sledenje žarka

How would we compute lighting? Let's *reverse* the light ray that got to our eye



Rekurzivno sledenje žarka

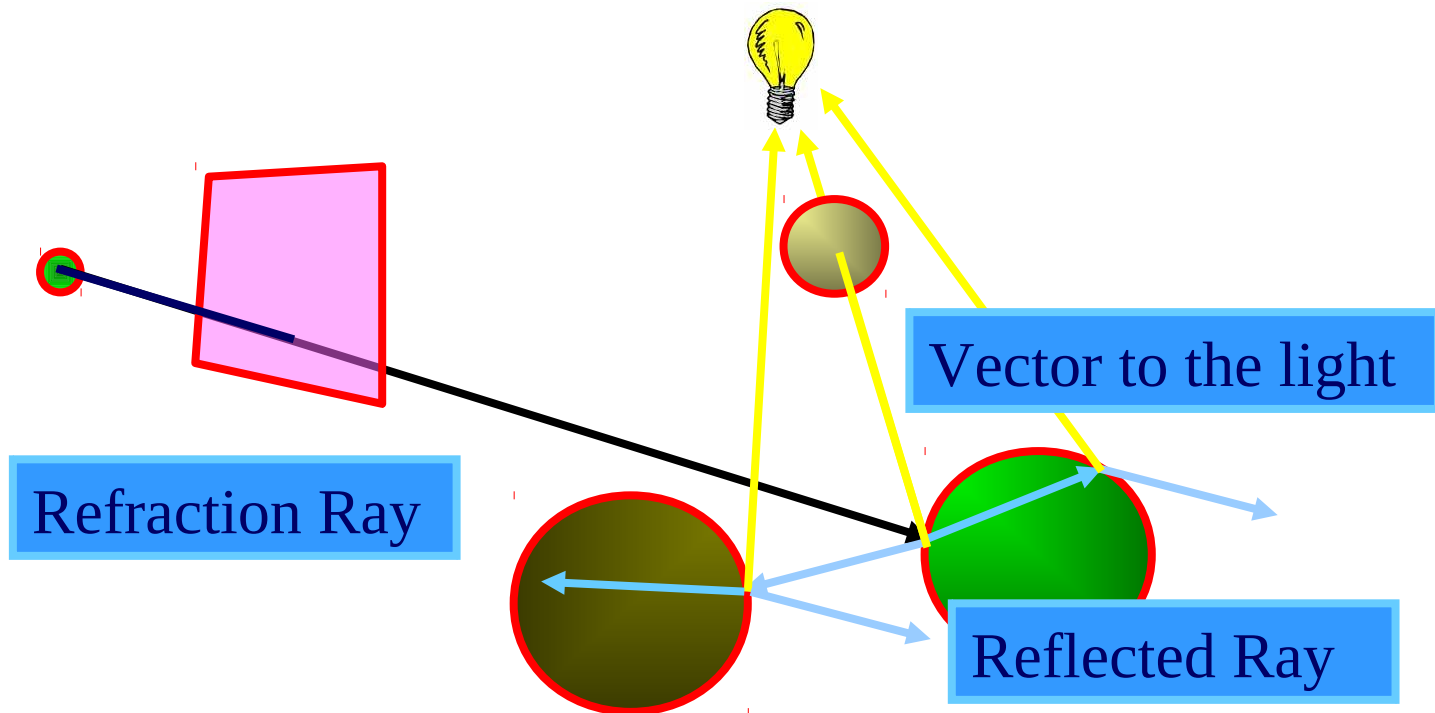
Shadows



Rekurzivno sledenje žarka

For each ray, we repeat the process

How many “levels of recursion” should we do?



Algoritem sledenja žarka

```
for (j=0;j<IMAGE_HEIGHT;j++)  
    for (i=0;i<IMAGE_WIDTH;i++)  
        result=CheckForIntersection(i,j)  
        SetPixel(i,j,result);
```

CheckForIntersection(i,j)

ray = vector from eye through pixel on display plane

if (DetermineIntersection(ray, color))

return color;

- $O(i*j*testi\ sekanja)$

Algoritem sledenja žarka

- Določimo koordinatni sistem, kjer so položaji pikslov označeni v u - v ravnini kamere.
- Iz položaja optičnega središča določimo pot žarka, ki prehaja skozi središče posameznega piksla.
- Za vsak žarek testiraj posamezno površino v prizoru, da ugotoviš, ali je žarek seka. Vsak žarek bodisi
 - seka površino
 - seka vir svetlobe
 - gre v neskončnost

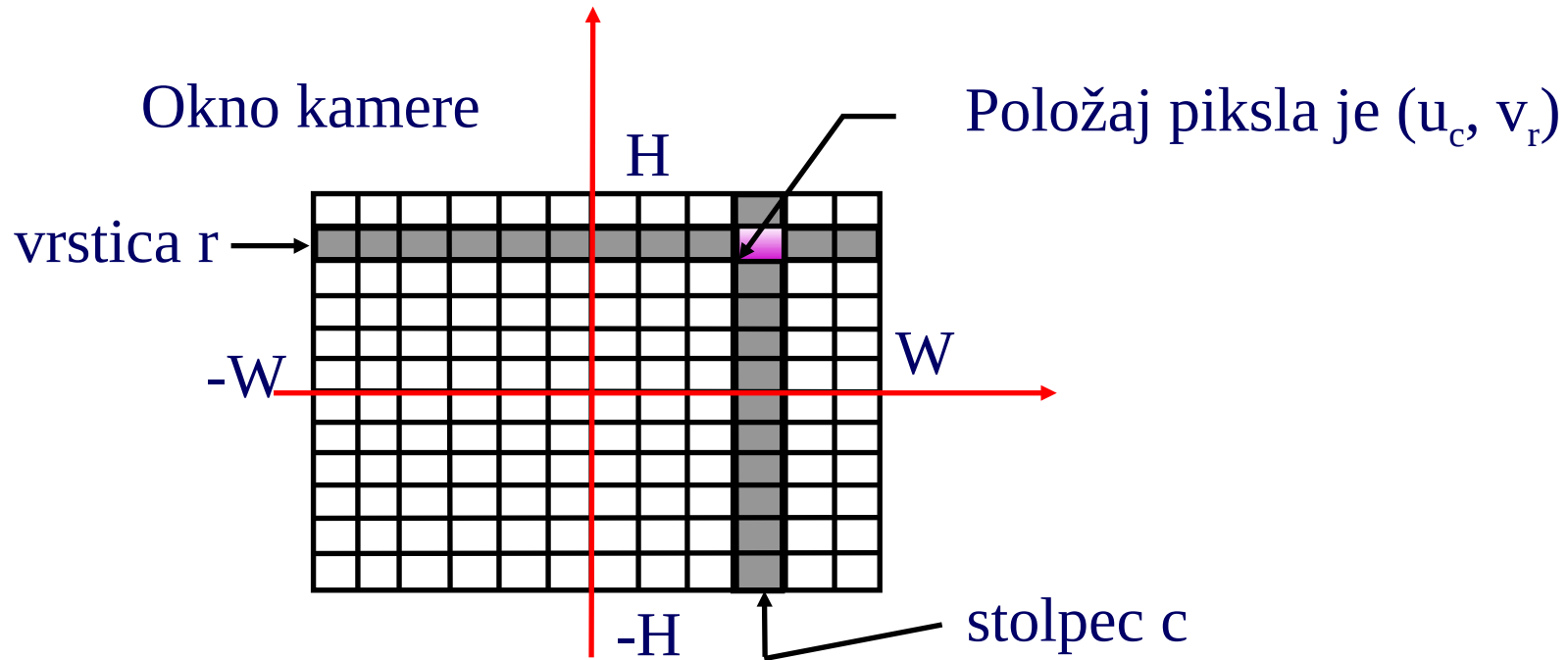
Sledenje žarka (ray tracing)

- e je sekana površina:
 - Dolo i njen as zadetka t . Shrani vse ase zadetka za vsak žarek za vsako površino. Objekt z najmanjšim $asom$ zadetka je najbližje o esu.
 - Dolo i to ko zadetka na objektu.
 - V to ki zadetka na površini objekta dolo i normalni vektor.
 - Izra $unaj$ barvo svetlobe, ki se odbije od površine v smeri o esa in jo shrani v piksel.

Sledenje žarka (ray tracing)

- Izra unaj odbite (zrcalne površine) in lomljene (prosojne površine) žarke.
- Za vsak žarek ponovi: tako primarno kot sekundarno.
- Jakost osvetlitve, ki jo dolo imo pikslu dobimo s seštevanjem prispevkov jakosti njegovih žarkov (drevo sledenja žarka).

Računanje položaja posameznega piksla



$$u_c = c(\text{širina enega stolpca}) - W$$

$$u_c = c \left(\frac{2W}{nCols} \right) - W$$

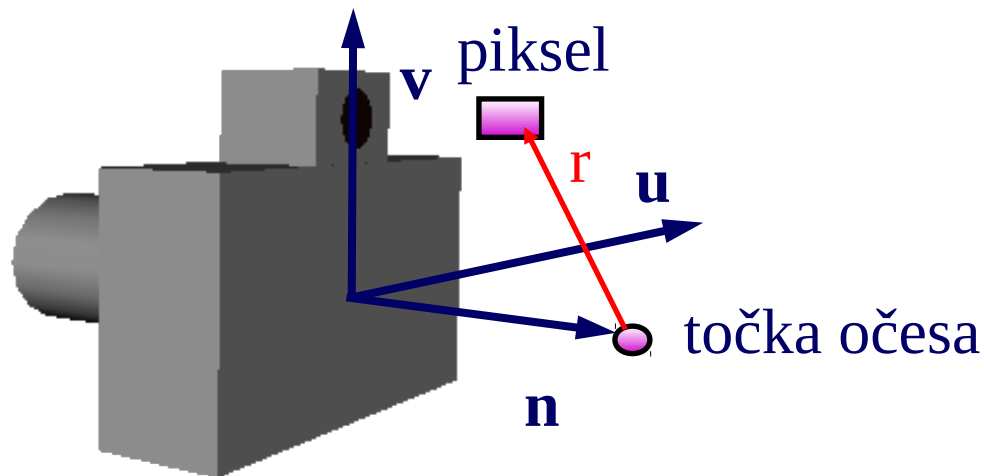
$$v_r = r \left(\frac{2H}{nRows} \right) - H$$

Računanje poti žarka skozi vsak posamezen piksel

Da najdemo 3D koordinate piksla, začnemo v točki očesa (eyepoint) in gremo vzdolž n , u in v smeri kamere.

$$r = eye - Nn + u_c u + v_r v$$

(N je razdalja od točke očesa do bližnje ravnine.)



Računanje poti žarka skozi vsak posamezen piksel

Izraz za pot žarka parametriziraj s t , tj. časom zadetka.

Naj bo $r(t) = \text{eye}$ za $t = 0$ in $r(t) = r$ za $t = 1$.

$$\text{Za } r = \text{eye} - Nn + u_c u + v_r v$$

$$r(t) = \text{eye}(1-t) - (\text{eye} - Nn + u_c u + v_r v)t$$

Enačbo lahko zapišemo tudi kot:

$$r(t) = \text{eye} + dir_{rc} t$$

Sekanje žarka z objektom

Implicitna oblika enačbe krogle

$$F(x, y, z) = x^2 + y^2 + z^2 - 1$$

Točka leži na površini, če $F(x, y, z) = 0$

Mesto zadetka se zgodi ob t_{hit} .

t_{hit} poiščemo z rešitvijo enačbe:

$$F(\text{eye} + \text{dir}_{rc} t_{\text{hit}}) = 0$$

Sekanje s transformiranim objektom

Če ima objekt funkcijo v implicitni obliki $F(P)$,
potem ima transformirani objekt funkcijo v implicitni obliki
 $F(M^{-1}(P))$

To se izpelje v:

$$r(t) = M^{-1}eye + M^{-1}dir_{rc} t$$

Sekanje s transformiranim objektom

Izračunaj t_{hit} na transformiranem objektu z

uporabo: $F(\mathit{eye} + \mathit{dir}_{rc} t_{hit}) = 0$

kjer je $F(x,y,z)$ funkcija originalnega,
netransformiranega objekta

Svetloba, ki zadene površino

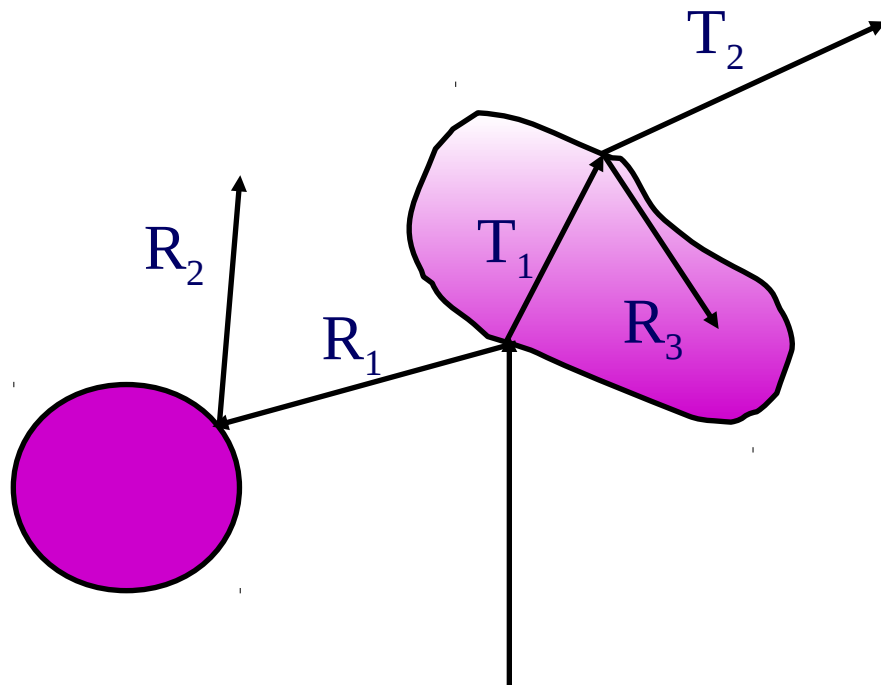
Če je izvir svetlobe viden v točki sekanja:

- Z uporabo standardnega modela odboja izračunamo prispevek od vira svetlobe
- Izračunamo odbiti žarek
- Izračunamo prepuščene žarke

Sestavljena jakost osvetlitve

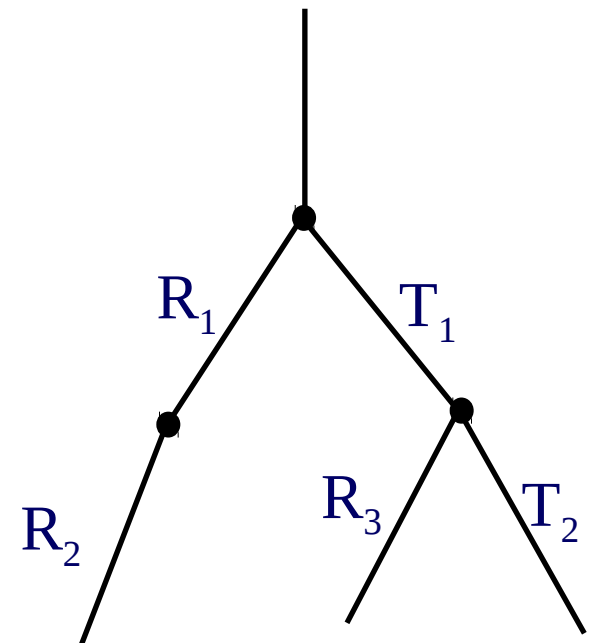
Jakost osvetlitve, ki je prirejena pikslu je sestavljena iz jakosti osvetlitev vseh objektov, ki jih zadene žarek.

Drevo sledenja žarka

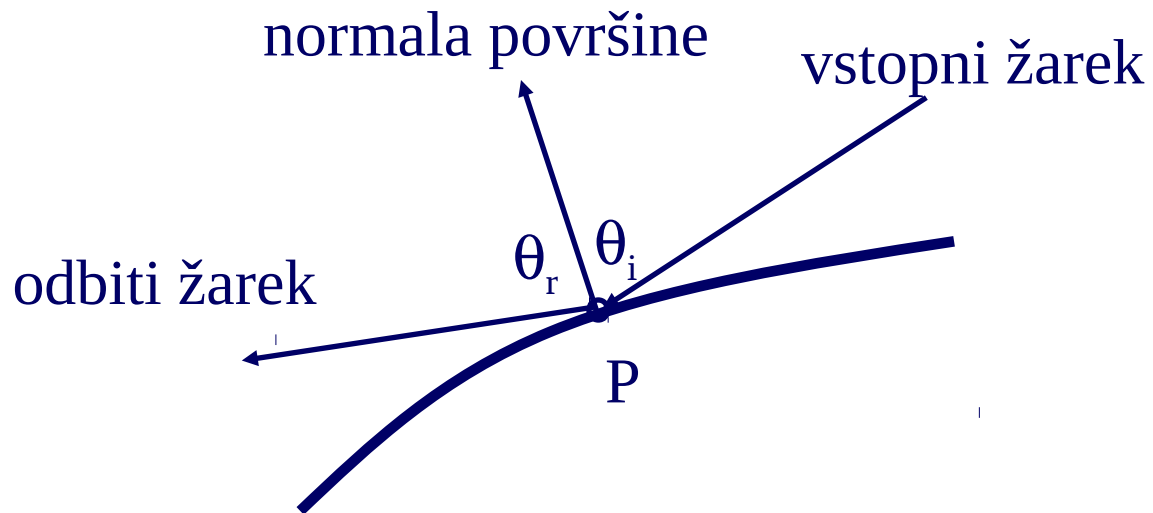


Točka očesa

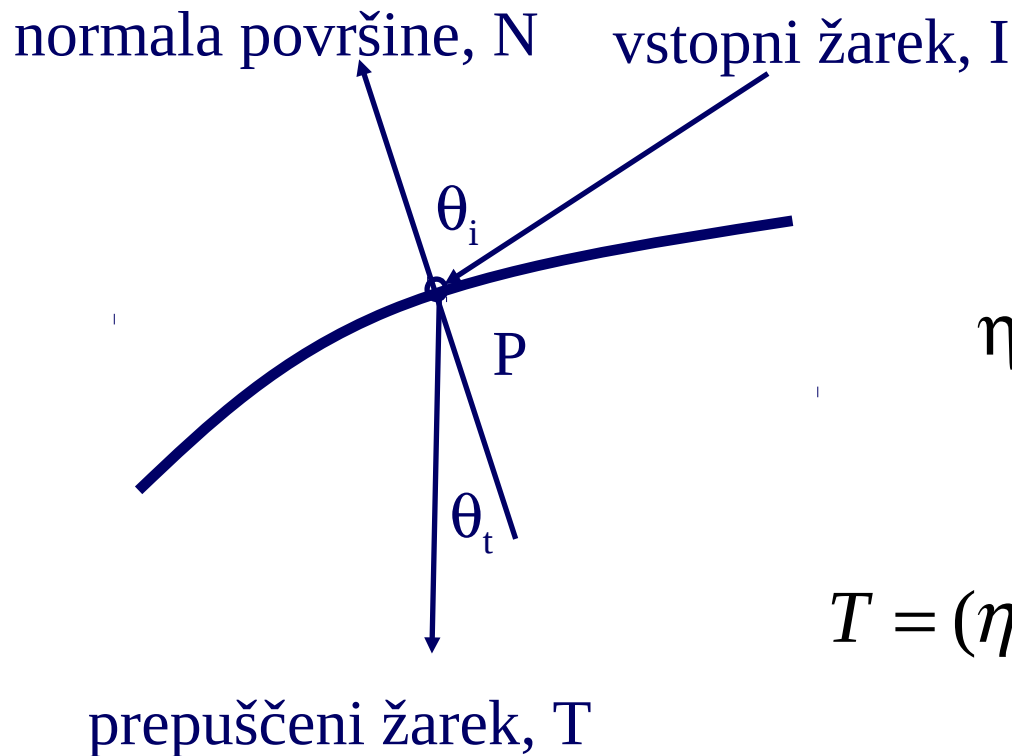
Točka očesa



Računanje odbitih žarkov



Računanje prepuščenih žarkov



Snellov zakon

$$\sin \theta_t = \sin \theta_i \left(\frac{\eta_{i\lambda}}{\eta_{t\lambda}} \right)$$

$\eta_{i\lambda}, \eta_{t\lambda}$: lomni količnik materialov

se izpelje v

$$T = (\eta_{r\lambda} \cos \theta_i - \cos \theta_t) N - \eta_{r\lambda} I$$

kjer je $\eta_{r\lambda} = \frac{\sin \theta_t}{\sin \theta_i}$

Sledenje žarka - prednosti/slabosti

Prednosti

- Prosojnost
- Odbojnost
- Sence
- Kompleksni primitivi
(matematične enačbe)

Slabosti

- Težko pohitriti
- Ni popolna globalna osvetlitev
- Zelo počasno izračunavanje posameznega piksla

Tehnike pohitritve

Hitrejša sekanja

- Hitrejša sekanja žarek-objekt
- Manj sekanj žarek-objekt

Manj žarkov

- Nadzorovana globina drevesa
- Prilagodljivo glajenje robov (anti-aliasing)

Posplošitev žarkov

- sledenje snopu
- sledenje stožcu
- sledenje žarkom, ki gredo skozi skupno točko

Razdelitev prostora

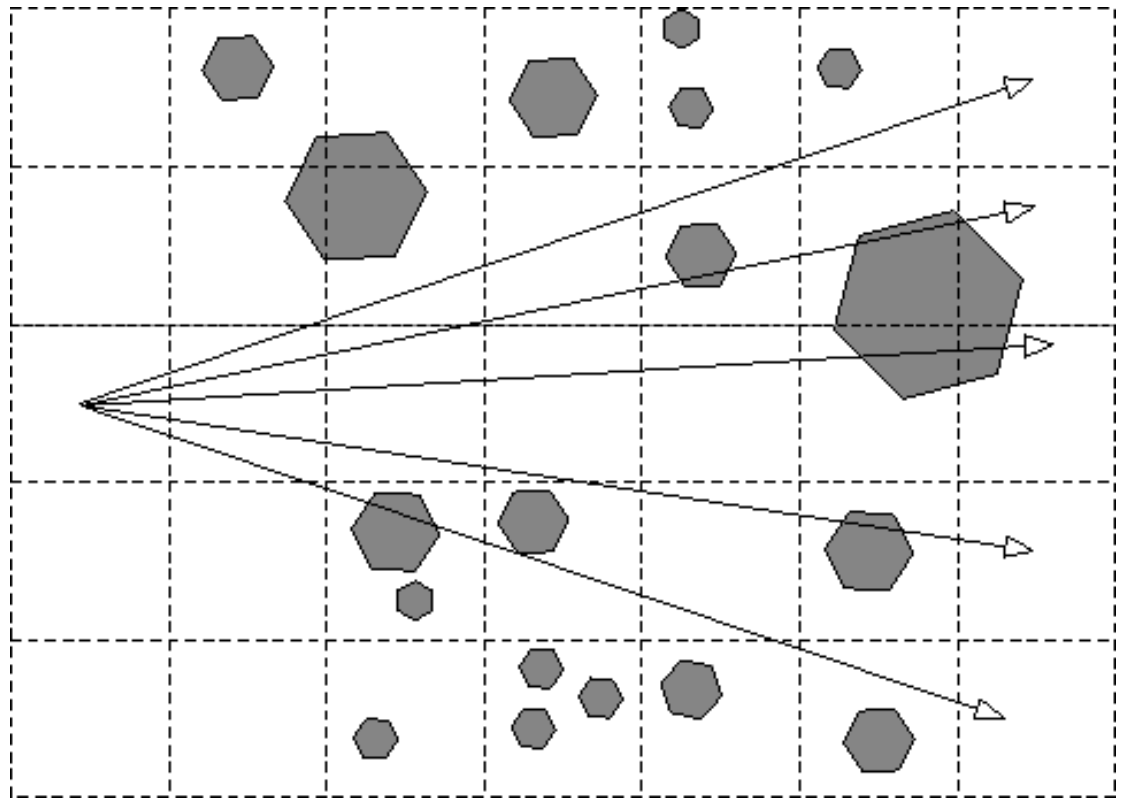
Razdeli prostor na regije

Postavi objekte v različne regije

Izračunaj katere regije lahko “vidiš” iz svoje trenutne pozicije

Testiraj samo te objekte

Sestavi hierarhijo testiranja



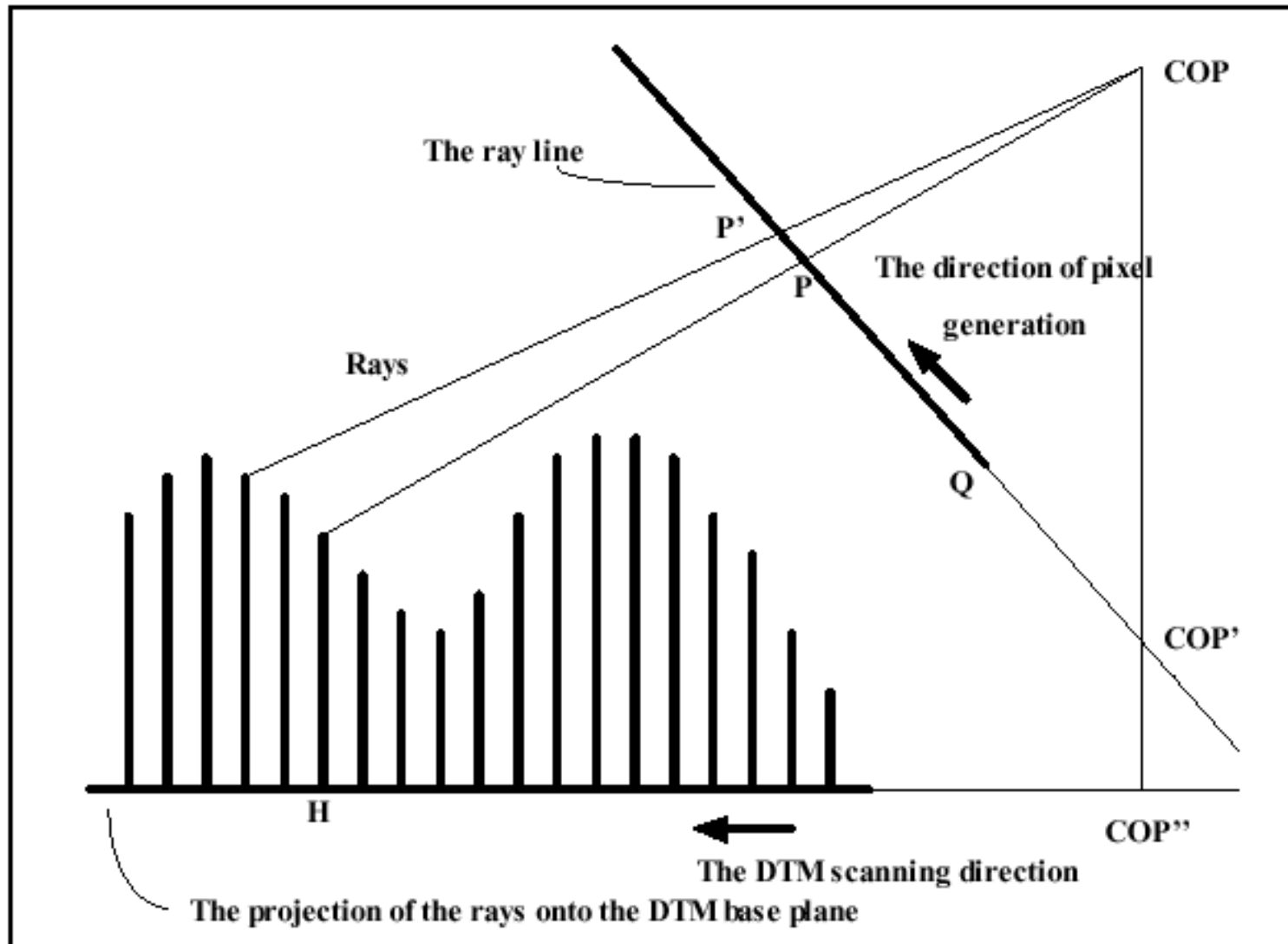
Bounding Volumes

A bounding volume is a volume which contains the entire object and has a simpler ray intersection test

Only if the ray intersects the bounding volume do we need to test against the object itself

In typical environments, a ray misses most objects

Upodabljanje terena



Zakaj sledenje žarka izgleda lažno

Jagged edges

Hard shadows

Everything in focus

Objects completely still

Surfaces perfectly shiny

Glass perfectly clear



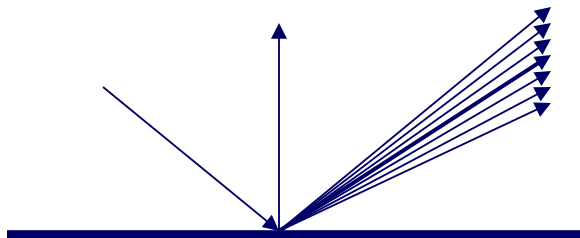
Blesteče površine

Surface microfacets perturb reflection ray directions

Nearby objects reflect more clearly because distribution still narrow

Farther objects reflect more blurry because distribution has spread

Reconstruction filter weighted toward reflection direction



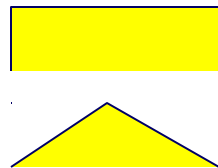
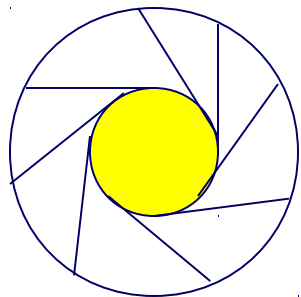
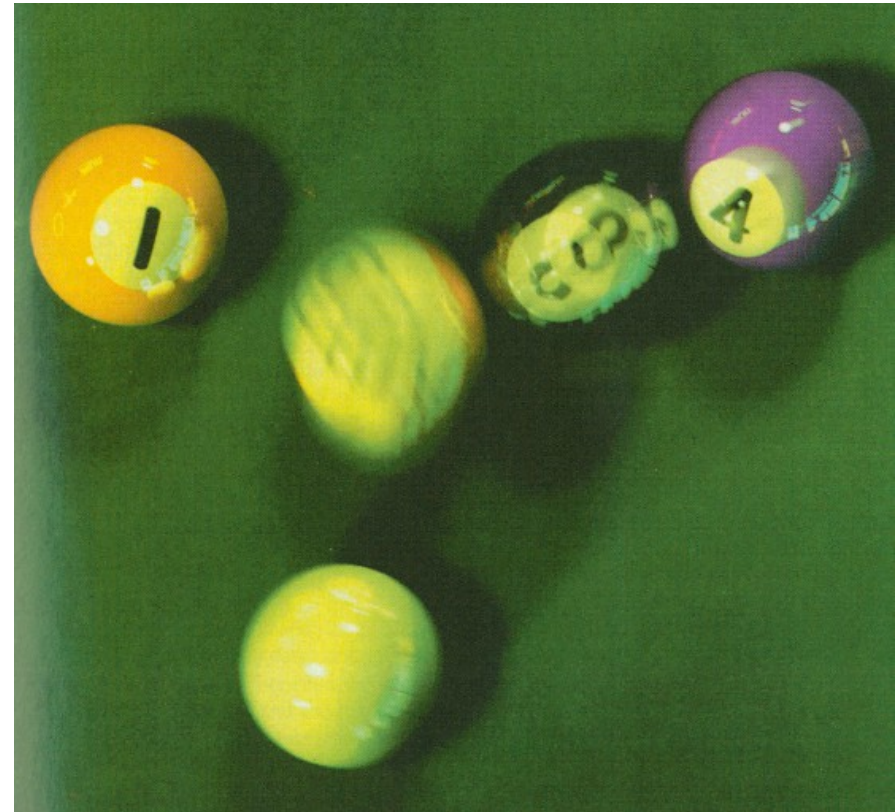
Zameglitev zaradi gibanja (Motion Blur)

Cast multiple rays from eye through **same point** in each pixel

Each of these rays intersects the scene at a **different time**

Reconstruction filter controls shutter speed, length

- Box filter – fast shutter
- Triangle filter – slow shutter



$$I(x, y, t) = r(t) * f(x, y, t)s(t)$$

shutter
function

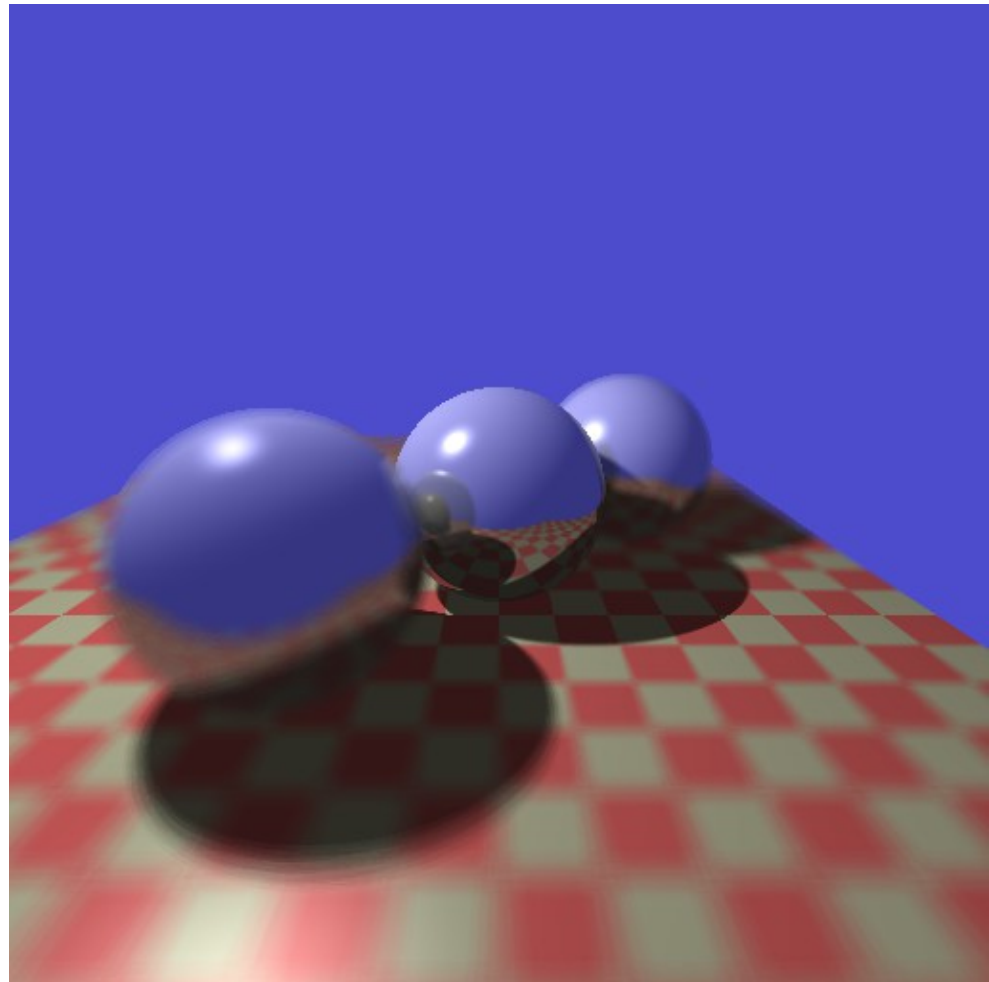
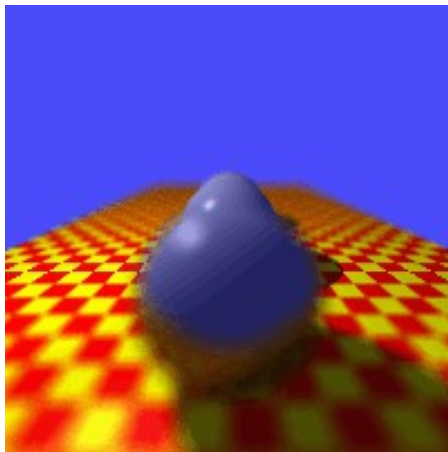
animated
continuous
image function

temporal
samples

Globina polja

Better simulation of
camera model

- f-stop
- focus



Demo blurr



Mehke sence

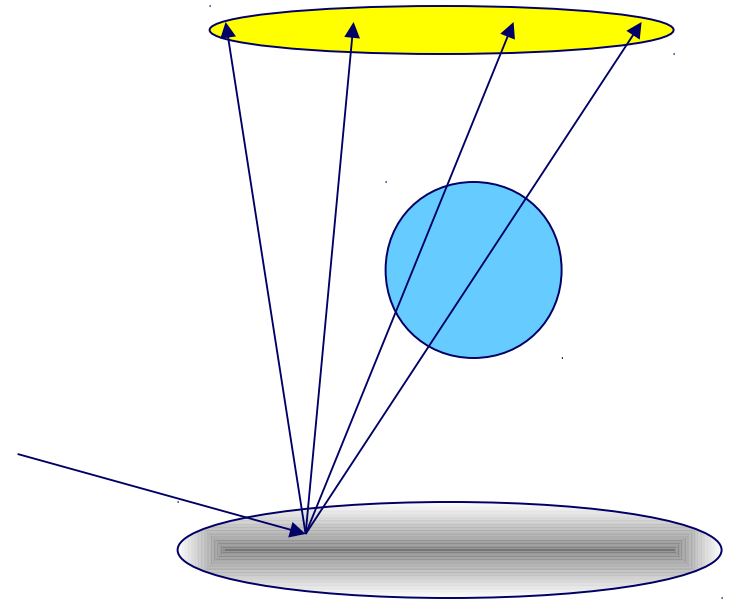
Point light sources unrealistic

Use area light source

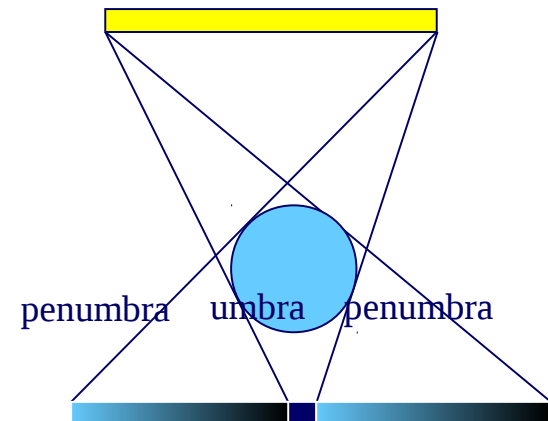
Cast shadow rays from surface to different locations on light

Hits/rays = % illuminated

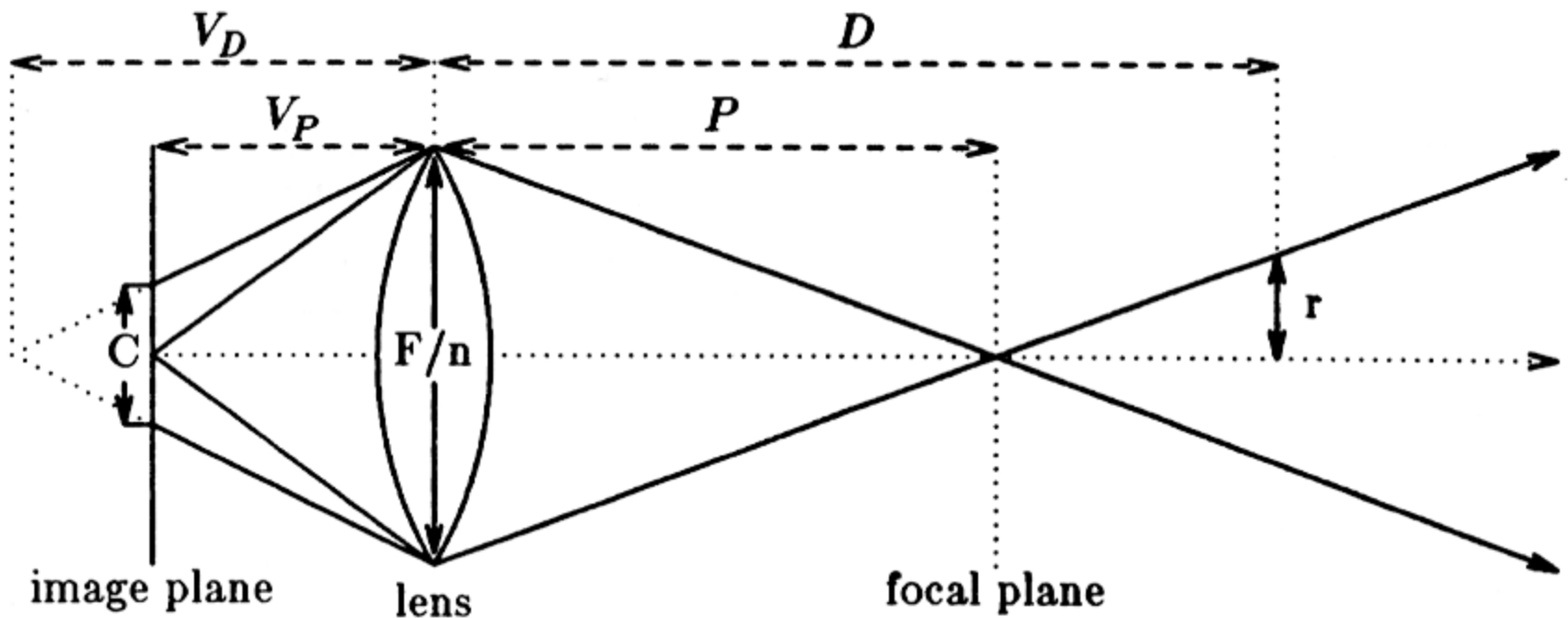
Jitter locations to remove aliasing artifacts



50% illuminated



Model kamere



F – focal length

n – aperture number

C – circle of confusion

$$V_P = FP/(P-F)$$

$$V_D = FD/(D-F)$$

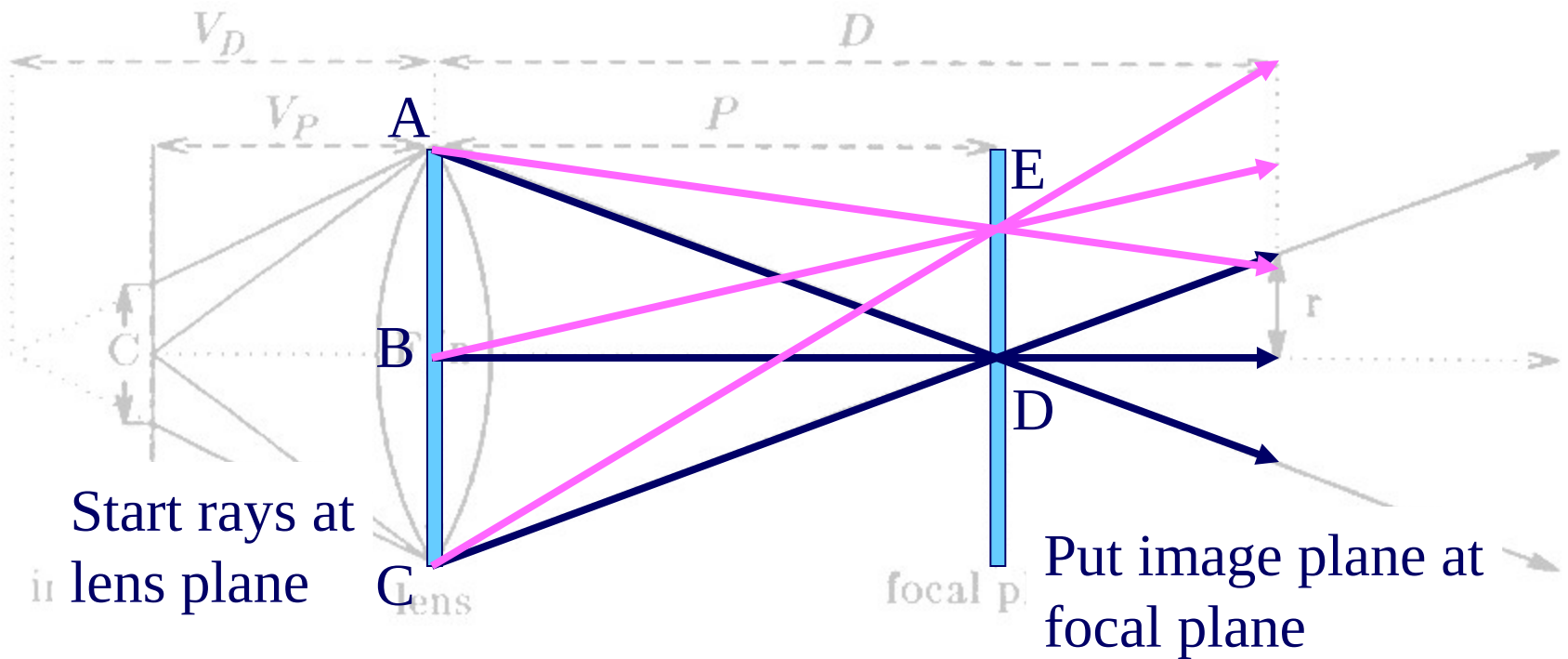
$$C = (|V_D - V_P|/V_D) (F/n)$$

$$r = \frac{1}{2} (F/n) (D-P)/P$$

$$R = (-V_P/D) r$$

$$R = \frac{1}{2} C$$

Implementacija



Standard ray tracing:

Pixel D uses ray BD

Pixel E uses ray BE

All rays emanate from B

Distributed ray tracing:

Pixel D uses rays AD, BD, CD

Pixel E uses rays AE, BE, CE

Rays emanate from lens plane

Porazdeljeno sledenje žarka (Distributed Ray Tracing)

Replace single ray with distribution of rays

Not just fat ray through pixel, but fat rays everywhere

Multiple eye rays

- supersampling
- motion blur
- depth of field

Multiple shadow rays

- soft shadows

Multiple reflection rays

- glossy surfaces

Multiple refraction rays

- diffuse glass

Supersampling

Cast multiple rays from eye through different parts of same pixel

Image function $f(x,y)$

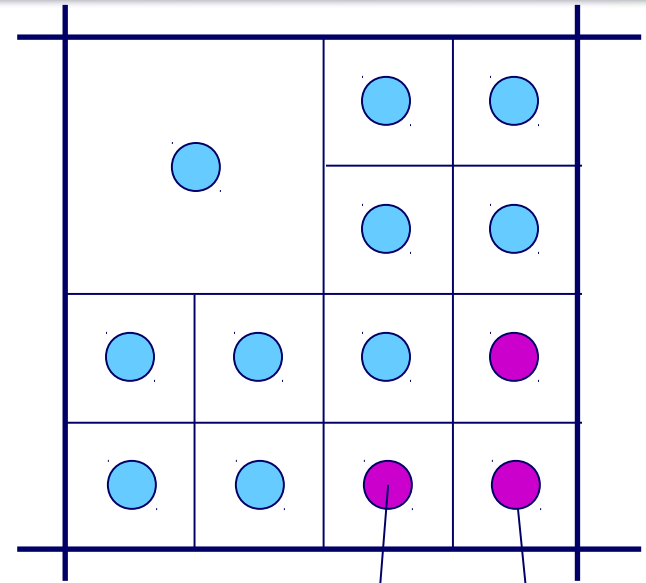
- Color that ray through (x,y) returns

Sampling function $s(x,y)$

- Controls where the samples occur within the pixel
- One if sample at (x,y) , else zero

Reconstruction filter $r(x,y)$

- Computes the weighted average of resulting colors into a single color
- Also weights according to area



$$\underbrace{I(x, y)}_{\text{reconstructed image function}} = \underbrace{r(x, y)}_{\text{reconstruction filter}} * \underbrace{f(x, y)}_{\text{continuous image function}} \underbrace{s(x, y)}_{\text{sampling function}}$$

$$I_{\text{pixel}} = \iint I(x, y) dx dy$$

$$I_{\text{pixel}} = I\left(\frac{1}{2}, \frac{1}{2}\right)$$

Porazdeljeno vzorčenje žarkov (Distributed Ray Sampling)

Supersampled antialiasing

- Jittering removed jaggies, Moires
- Ordered jitter, Poisson balanced distribution

Decorrelation inhibits aliasing

Need to decorrelate samples

- across all parameter axes
- between all parameter axes

