

Globaalse valgustuse mudelid

Mark Fishel (fishel@ut.ee)

31. oktoober 2005



Globaalse valgustuse mudelid

- *Lokaalne valgustus* – tuleb otse valgusallikast
- *Globaalne valgustus* – peegeldatud teistest pinnadest
- Lokaalsetes mudelites globaalset valgust immitseeritakse keskkonna valgustuse (ambient light) komponendiga
- Muude efektide realiseerimiseks kasutatakse ugly hack'e



Täna

- Rekursiivne Ray Tracing
- Stohhastiline Ray Tracing
- Radiosity

- Lõikepunktide arvutamine
- Pindade klotsideks jagamine
- Form Factor'ite arvutamine



Ray Tracing



Ray Casting

Alternatiivne nähtavate pinnade determineerimise ja varjude simuleerimise algoritm:

- Iga ekraani piksli jaoks konstrueerime valguse kiire simuleerivat sirgjoont mis väljub vaataja asukohast ning läheb antud pikslist läbi
- Leiame sirgjoone lähima lõikepunkti mingi objektiga simuleeritud maailmast
 - Sedasi leiame mis objekti on näha läbi selle piksli



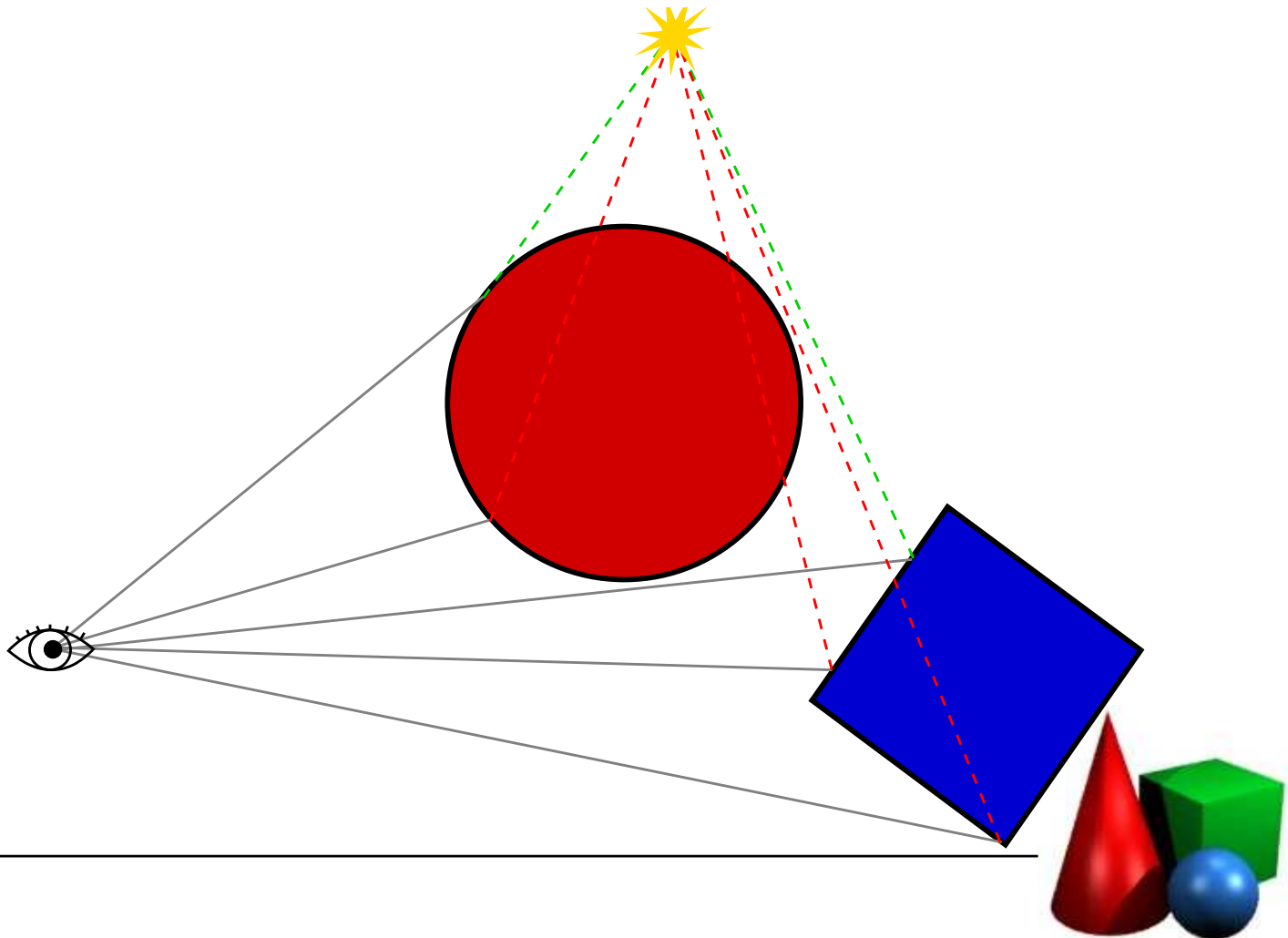
Ray Casting

- Konstrueerime veel ühte sirglõiku leitud punkti ja valgusallika asukoha vahel
 - Nimetame seda varju kombitsaks (shadow feeler)
 - Kui see lõikub mingi objektiga, punkt on varjus antud valgusallika suhtes
- Punkti valgustuse arvutamiseks kasutame lokaalse mudeli:

$$I = I_{loc}$$



Ray Casting



Peegeldamine

Realiseerime lisakiirega:

- Suunda määravad originaalkiire suunavektor ja objekti pinna normaal antud punktis
- Leiame lähima lõikepunkti
- Sealt laseme varju kombitsa ja uue peegeldatud kiirt – rekursiivselt
- Punkti valgustus:

$$I = I_{loc} + k_r I_r$$

- $k_r \in [0, 1]$ – peegeldamise koeffitsient



Läilaskine

Realiseerime lisakiirega:

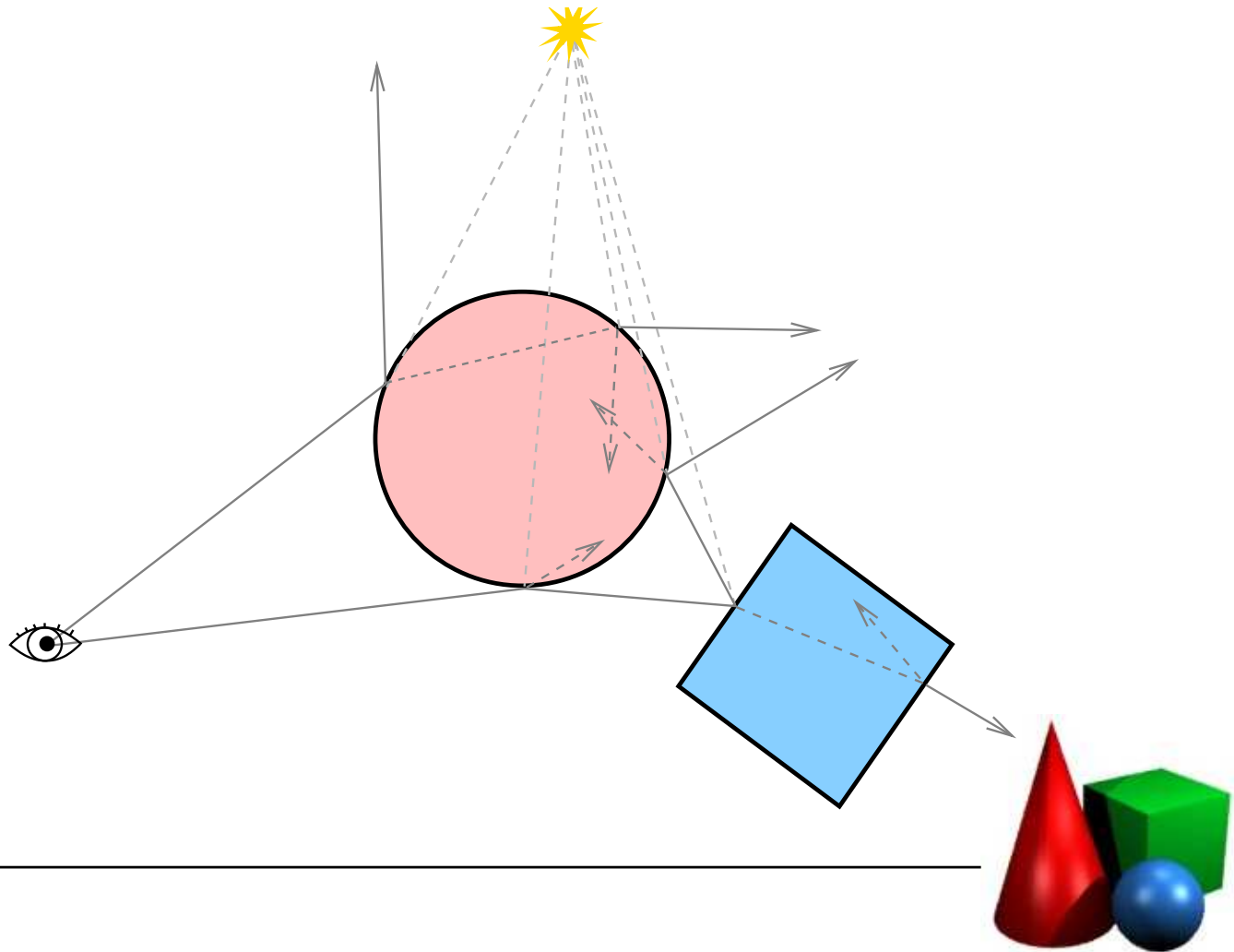
- Suunda määravad suunavektor, pinna normaal ja Snell'i seadus
- Lähim lõikepunkt on objekti endaga (seest)
- Laseme varju kombitsa, peegeldatud ja läbi lastud kiireid rekursiivselt
- Punkti valgustus:

$$I = I_{loc} + k_r I_r + k_t I_t$$

- $k_t \in [0, 1]$ – läbi laskmise koeffitsient



ekursiivne ay racing



Effektiivsuse tõstmine

- Objektide z-puhver
 - renderdame stseeni puhvrisse
 - valgustuse/sügavuse asemel salvestame mis objekt on antud pikslist nähtav
- Kohanev rekursiooni sügavuse kontroll
 - kaalutud rekursiivsed kiired, kui kaalude korrutis liiga väike – ei lase kiirt
- Valguspuhver
 - igal valgusallikal on oma objektide z-puhver
 - kontrollime kas varju kombitsale vastav puhvri väärtus on punkti omav objekt



Sakilisus

Põhiprobleem: aliasing, kuna üks kiir määrab piksli värvi. Lahendused:

- Cone Tracing
- Beam Tracing
- Pencil Tracing
- Stohhastiline Ray Tracing



Stochastic racing



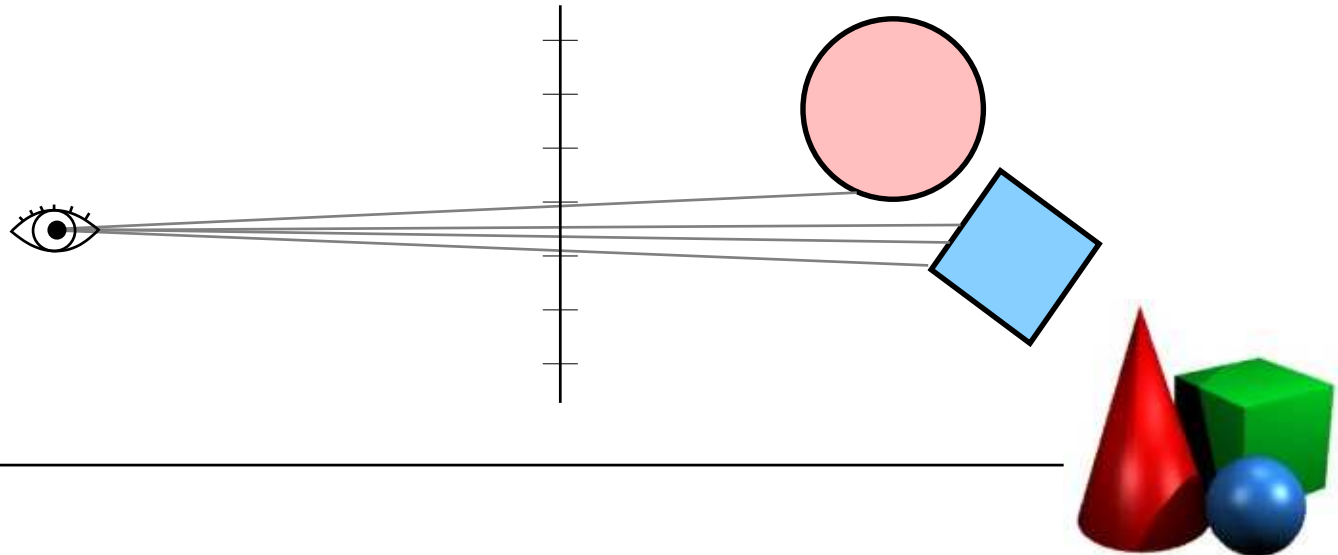
õhi õte

- Ühe kiire asemel laseme mitu kiirt iga piksli jaoks
- Iga kord muudame mingit parameetrit
- Muutus on stohhastiline
 - juhuslikuse element sees
 - iga kord genereeritakse uusi muutusi



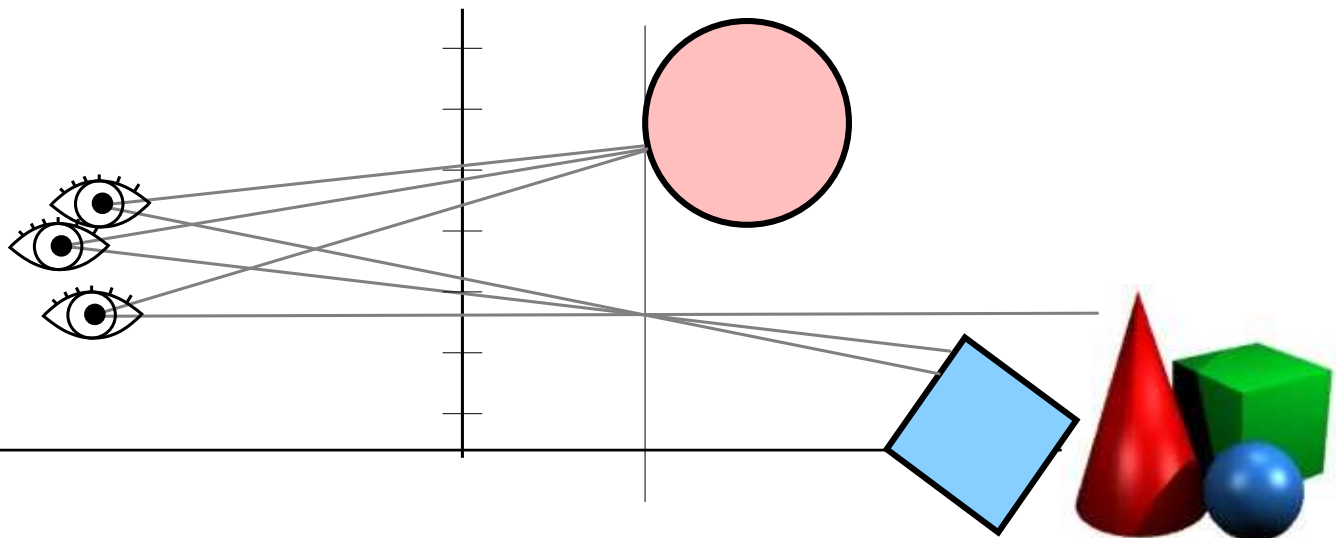
ntialiasing

- Laseme mitu kiirt läbi tervet piksli
- Punktid valitakse kasutades virvendamistehnikat (*jittering*)
- Piksli värviks on kiirte tulemuste aritmeetiline keskmine



Depth-of-Field

- Muudame vaatleja asukohta (originaalasukohta lähikäpses)
- Jätame fookus-tasandi paigale
- Tulemus – mida kaugem objekt on fookus-tasandist, seda udusem on selle kujutus



Motion blur

- Ühe kaadri sees muudame liikuva objekti asukohta
- Vastab virvendamisele ajas
- Vastab tegelikule efekti sisule, kui hästi mitu kiirt lasta



uu ilise valgusalli a

- Iga kiire jaoks muudame valgusallika asukohta (tegeliku ruumilise allika piires)
- Võrreldes lähenemisega “asendame mitme punktallikaga”
 - Pole probleemi läikimisega
 - Pole näha erinevaid punkallikaid stohhastilisuse pärast



efra tsioon

- Valgus erineva laine pikkusega – erinev refraktsiooni nurk
- Laseme mitu kiirt, iga vastab omaale laine pikkusele

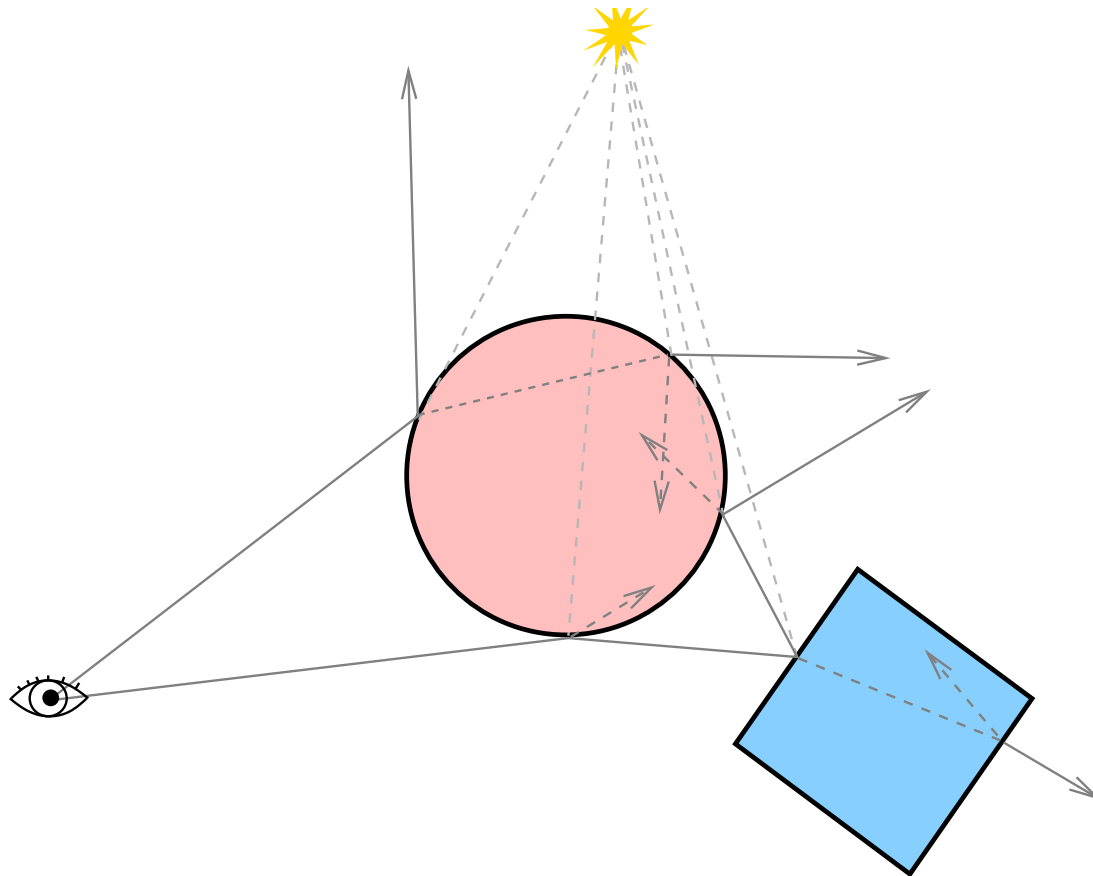


Kui as o ineeri a

- Näiteks antialiasingu 3×3 koos depth-of-field'iga 3×3
- Ei pea laskma 81 kiirt!
- Selle asemel valime juhuslikult kiired paaridesse, tulemuseks ikka 3×3

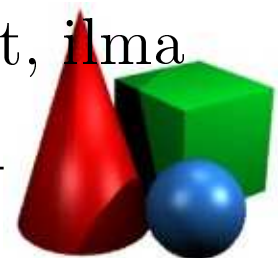


efra tsiooni pro lee



Ray racing tagurpi i

- Laseme kiired mitte vaataja asukohast vaid valgusallika asukohast
- Puhtalt rakendades pole efektiivne sest enamus kiiri ei satu meie silma
- Efektiivne kombineerides “edaspidise” ray tracing’uga:
 - Jagame pinnad klotsideks
 - Lastes kiired alates valgusallikatest arvutame ja salvestame iga klotsi valgustatust
 - Edasi rakendame tavalist ray tracingut, ilma varju kombitsateta



a iosity



Realism

- Mõeldud difuussete pinnade realistlikuks simuleerimiseks
- Pinnade varjustus sõltub
 - Valgusallikate asukohast
 - Stseeni geomeetrisest kirjeldusest
- Ei sõltu vaatleja asukohast
 - Eelarvutatav



lotsi

- Pinnad jagatakse klotsideks
- Iga klots mõjutab teisi peegeldades valgust diffusselt
- Sõltuvalt stseeni geomeetrisest kirjeldusest vastastikkune toime on erinev



Valgusalli a

- Valgusallikateks on samad klotsid
- Emitteerivad valgust
- Lihtsuse mõttes valgust emitteerivad kõik klotsid, mõned intensiivsusega 0
- Punktallikaid modelleerida raskem, saab kui väga väikest klotsi



Igorit ül iselt

- Igal klotsil arvutatakse palju valgust ta saab teistest, sellest saabki tema kiirgusintensiivsus (radiosity)



Algoritmi täpsed väärtused

- Klotsi poolt kiirgatava valguse intensiivsus võrdub

$$B_i^{exc} = E_i + R_i B_i^{in}$$

- E_i on emiteeriva valguse intensiivsus
- R_i on peegeldamise koeffitsient
- **Eesmärk:** leida väärtused B_i^{exc}



Algoritmi täpse ini

- Klotsi j mõju klotsile i väljendab Form Factor F_{ji}
- Klotsi sissetuleva valguse summaarne intensiivsus võrdub

$$B_i^{in} = \sum_{j=1}^n B_j^{exc} F_{ji} \frac{A_j}{A_i}$$

- A_i on i 'nda klotsi pindala
- Kuna $A_i F_{ij} = A_j F_{ji}$:

$$B_i^{in} = \sum_{j=1}^n B_j^{exc} F_{ij}$$



ööor aatri siga la en a ine

$$B_i^{exc} = E_i + R_i \sum_{j=1}^n B_j^{exc} F_{ij}$$

LVS n tundmatu väärtusega ja n valemiga.
Maatrikskujus:

$$\mathbf{B} = \mathbf{E} + \mathbf{M} \mathbf{B}$$

kus $M_{ij} = R_i F_{ji}$. Siit

$$\mathbf{B} = \mathbf{E} (\mathbf{I} - \mathbf{M})^{-1}$$



Iteratiivne la en a ine

- Igal iteratsioonil arvutame B_i^{in} ja B_i^{exc}
- Iga iteratsiooniga stseen läheneb oma lõppseisundile



ui jõua e

- Kuidas arvutada kiirte ja objektide lõikepunkte
- Kuidas jagada pinnad klotsideks
- Kuidas arvutada Form Factor'eid



si use ?

