



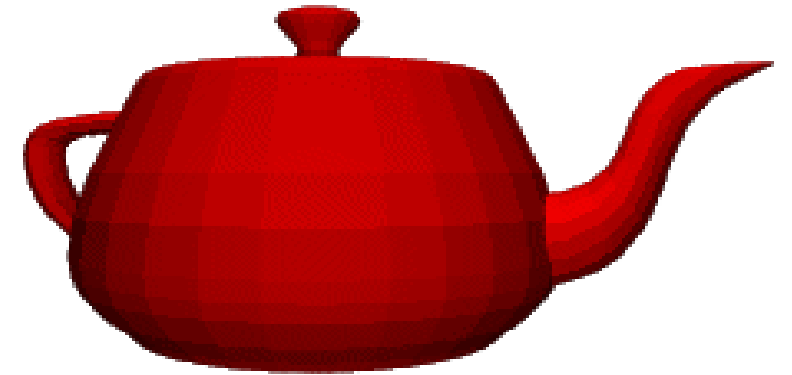
OSVETLJEVANJE



- Računamo interakcijo svetlobe s površinami
- Bolj ali manj kompleksna simulacija fizike



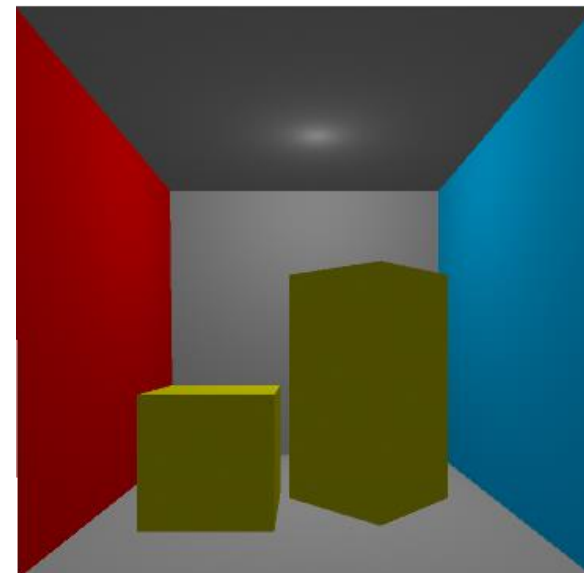
Osvetljevanje





Lokalno in globalno osvetljevanje

- Glede simulacijo širjenja svetlobe po prostoru lahko ločimo
- **Lokalno** osvetljevanje
 - poenostavljeno
 - le en odboj med izvorom in gledalcem
 - računanje osvetlitve ene ploskve je neodvisno od ostalih
 - ni senc
 - enostavna implementacija v cevovodu
- **Globalno** osvetljevanje:
 - upoštevamo več odbojev svetlobe od predmetov
 - algoritmi temeljijo na sledenju žarkov
 - računsko zahtevno





Osvetlitveni model



Osvetlitveni model – reflection model

- Za dobro upodobitev predmetov moramo znati modelirati kako se svetloba odbija od predmetov
 - odboj, odsev - *reflection*
 - odboj je odvisen od materiala
- Kako se svetloba odbija od npr.
 - ogledala
 - papirja
 - lesa
 - kovine





- Osvetlitveni model določa kako svetloba interaktira s predmetom
- Enostaven model: odbita svetloba je vsota treh komponent
 - **razpršeni** odboj (*diffuse*)
 - **zrcalni** odboj (*specular*)
 - **ambientna** osvetlitev (*ambient*)
- S tovrstno predstavitevijo lahko zajamemo velik nabor realnih materialov

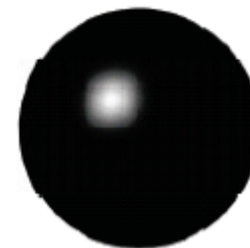


Enostaven osvetlitveni model

razpršeni odboj



zrcalni odboj



ambientna svetloba



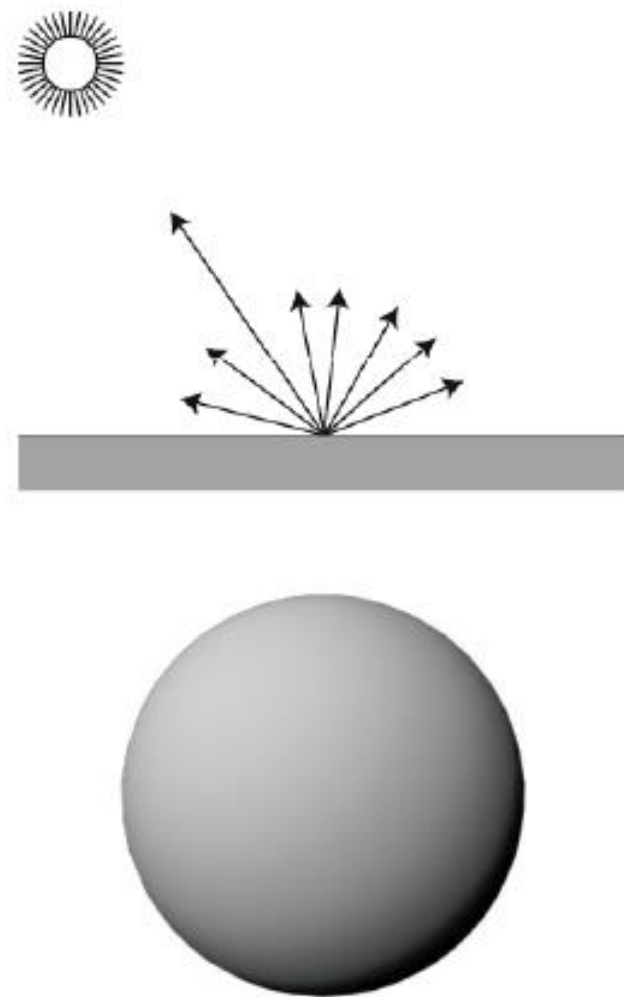
vsota





Razpršeni odboj – diffuse reflection

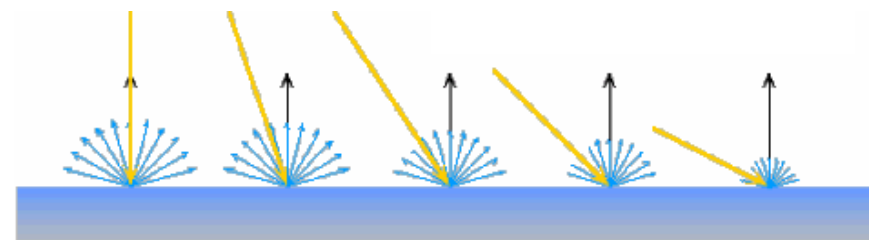
- Material, ki odbija svetlobo enakomerno na vse strani
- Ni važno s katere strani oz. pod kakim kotom ga gledamo, izgleda približno enako
- Motni, nesvetleči materiali, ki so drobno hrapavi na površini
 - papir
 - neobdelan les
 - neobdelan kamen





- Odbita svetloba:
 - proporcionalna s kosinusom kota med vpadno svetlobo in normalo na površino
 - večji kot, manj je površina svetla
 - pravilo imenujemo **Lambertov kosinusni zakon** (Johann Heinrich Lambert, 1760)
 - površinam oz. materialom zato rečemo tudi Lambertovi

Razpršeni odboj

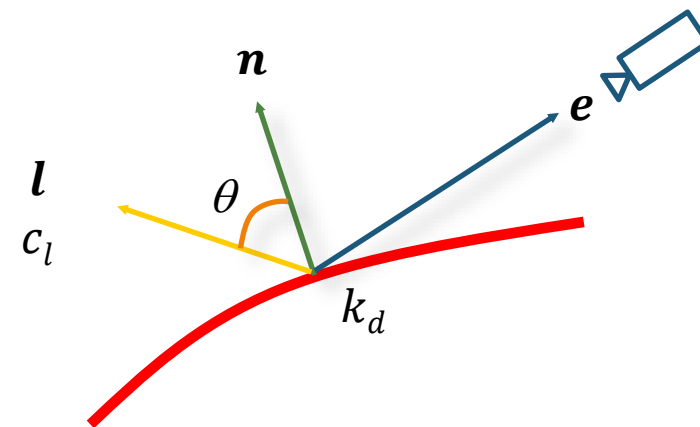




- Komponente:
 - \mathbf{n} – normala na površino (normirana)
 - \mathbf{l} – smer svetlobe (normirana)
 - k_d – razpršena odbojnost – RGB vektor, ki določa koliko svetlobe se odbije – barva materiala
 - c_l – intenziteta vpadne svetlobe – RGB vektor
 - c – intenziteta odbite svetlobe – RGB vektor
- V enačbi **ni smeri pogleda** \mathbf{e} – od povsod predmet izgleda enako



Razpršeni odboj

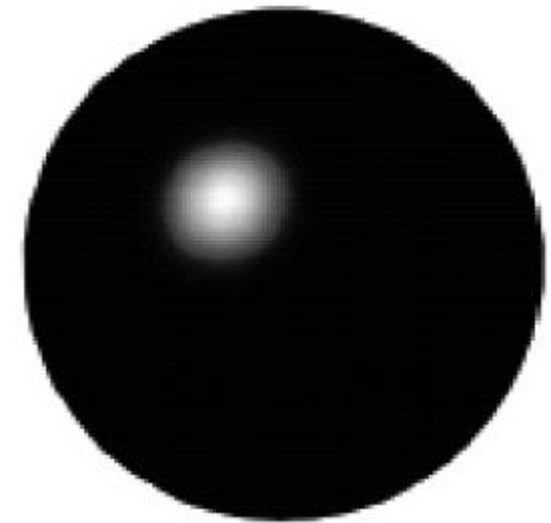


$$c = c_l k_d (\mathbf{n} \cdot \mathbf{l}) = c_l k_d \cos \theta$$



Zrcalni odboj – specular reflection

- Odblesk/sijaj (*specular highlight*)
 - zabrisan odsev vira svetlobe
 - položaj je odvisen od smeri gledanja
- Svetleče površine
 - polirana kovina
 - lakirani materiali, npr. avto
 - plastika

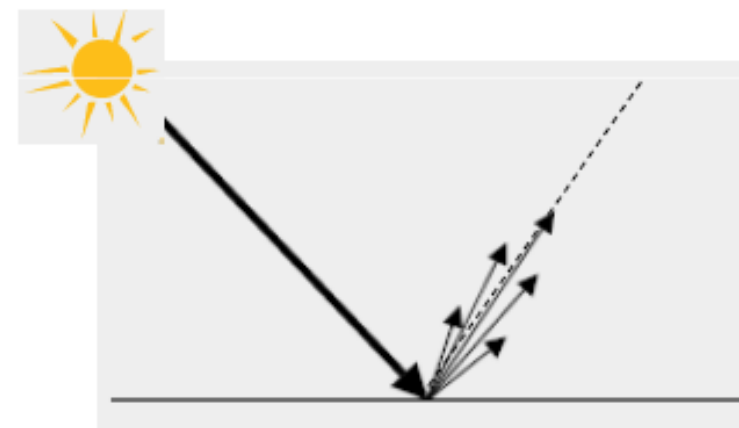
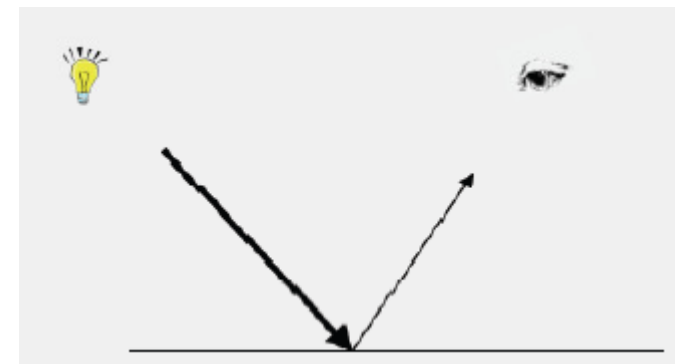




- Idealen odboj v ogledalu
 - popolnoma gladka površina
 - kot vpada = kot odboja
- Tipični materiali niso popolnoma odbojni
 - svetleči materiali, so mikro hrapavi



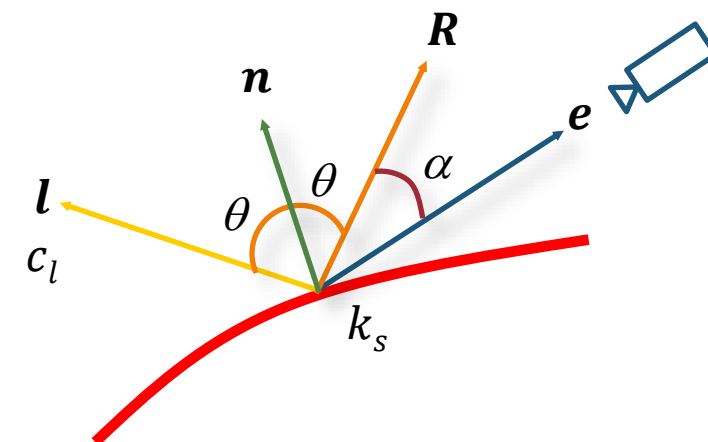
Zrcalni odboj





- Svetloba se odbija približno v smeri idealnega odboja
- **Phongov model** (Bui-Tuong Phong, 1975)
 - kot α med idealnim odbojem \mathbf{R} in smerjo pogleda \mathbf{e} določa količino odbite svetlobe
 - parameter zrcalnega odboja p določa velikost razpršitve – večji p , bližje smo idealnemu odboju

Zrcalni odboj

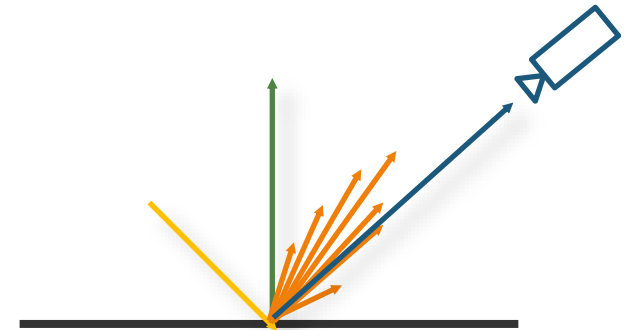
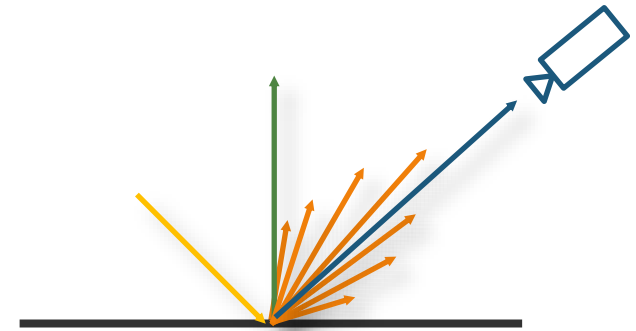
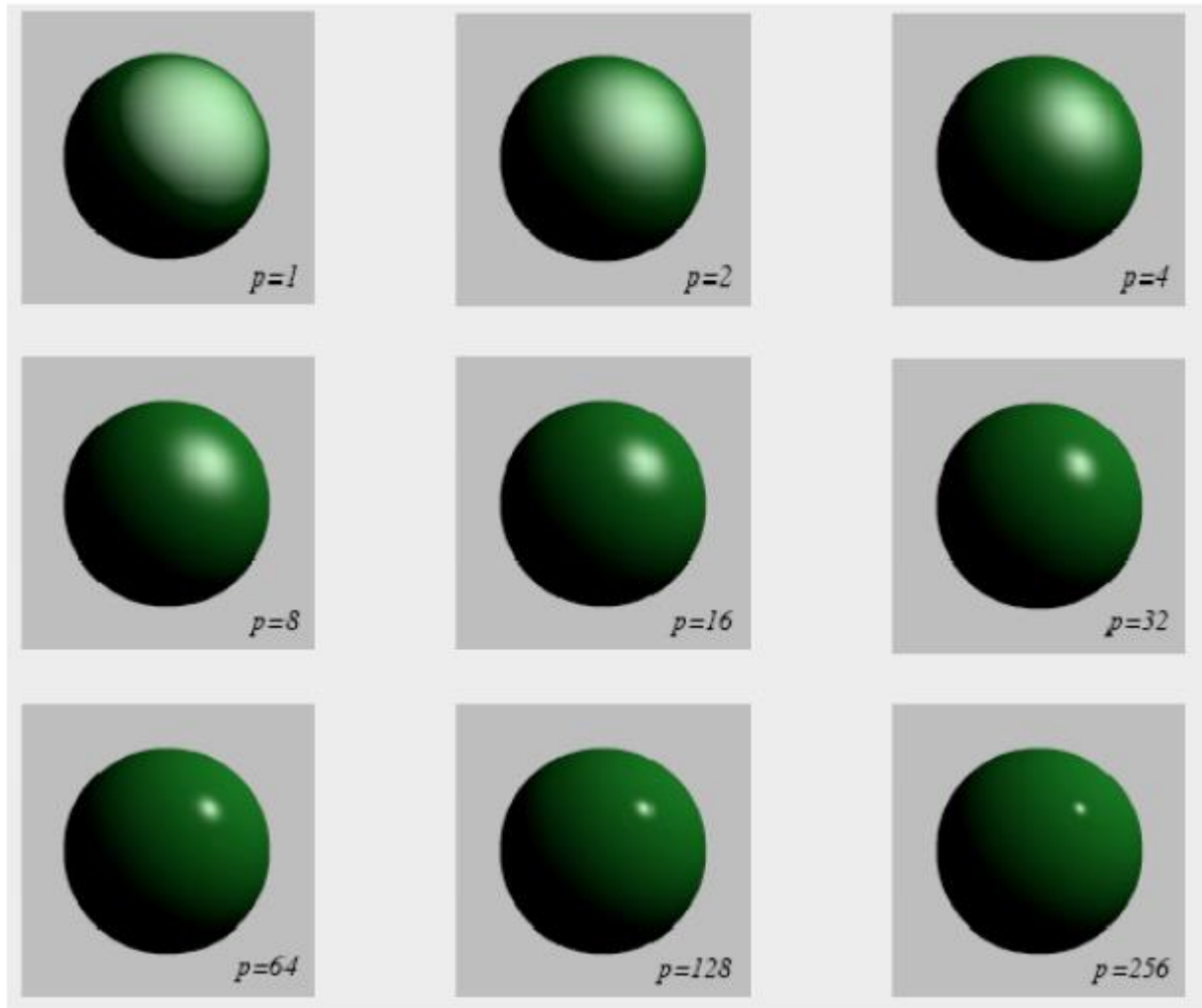


$$c = c_l k_s (\mathbf{R} \cdot \mathbf{e})^p$$

$$\mathbf{R} = 2(\mathbf{l} \cdot \mathbf{n})\mathbf{n} - \mathbf{l}$$



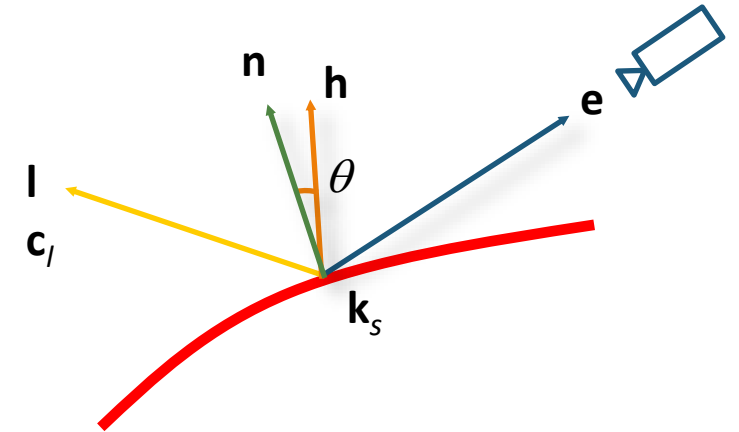
Phongov model





Blinnov model (Jim Blinn, 1977)

- Podoben Phongu
 - ne potrebujemo izračuna odboja
 - kompromisni vektor \mathbf{h} , ki je na sredini med \mathbf{l} in \mathbf{e}
 - če je blizu normale, smo blizu popolnega odboja
 - parameter svetlosti p
- Če sta luč in gledalec daleč od površine, lahko predpostavimo, da je \mathbf{h} konstanten
 - hitrejši izračun



$$c = c_l k_s (\mathbf{h} \cdot \mathbf{n})^p$$

$$\mathbf{h} = \frac{\mathbf{l} + \mathbf{e}}{|\mathbf{l} + \mathbf{e}|}$$



- V realnosti je del svetlobe povsod, saj se odbija od sten in ostalih predmetov v sceni
- Pri lokalni osvetlitvi jo aproksimiramo z **ambientno svetlobo**
 - predstavlja približen prispevek svetlobe k splošni sceni ne glede na položaj luči in predmetov
- Povsod dodamo konstantno osvetlitev
- Nimamo več popolnoma temnih delov

Ambientna svetloba



$$c = c_a k_a$$



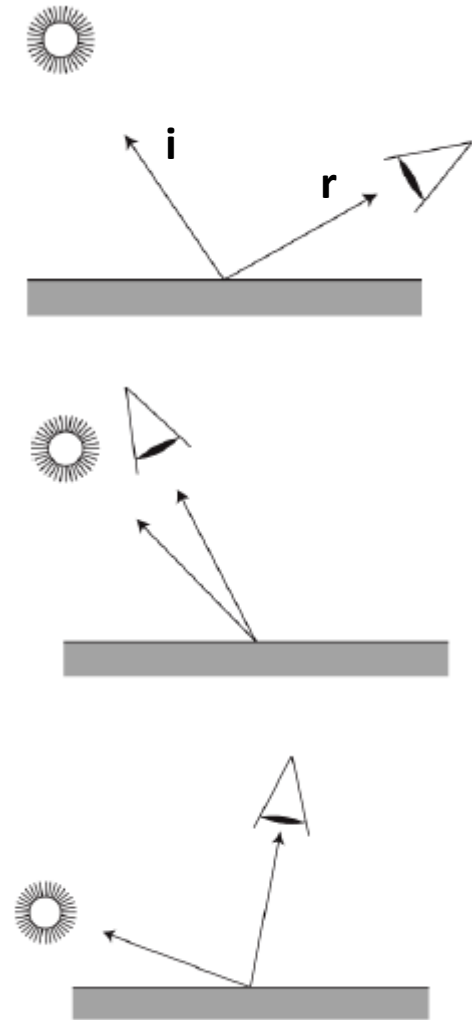
Izračun osvetlitve

- Imamo lahko več virov svetlobe, seštejemo prispevke vseh
- c_a in c_i določajo RGB jakost in barvo svetlobnih virov
- k_a , k_d in k_s določajo RGB odbite količine svetlobe snovi – barvo predmeta
- Izračun s **Phongovim modelom** (za več luči):
 - $c = c_a k_a + \sum_i c_i (k_d (\mathbf{l}_i \cdot \mathbf{n}) + k_s (\mathbf{R}_i \cdot \mathbf{e})^p)$

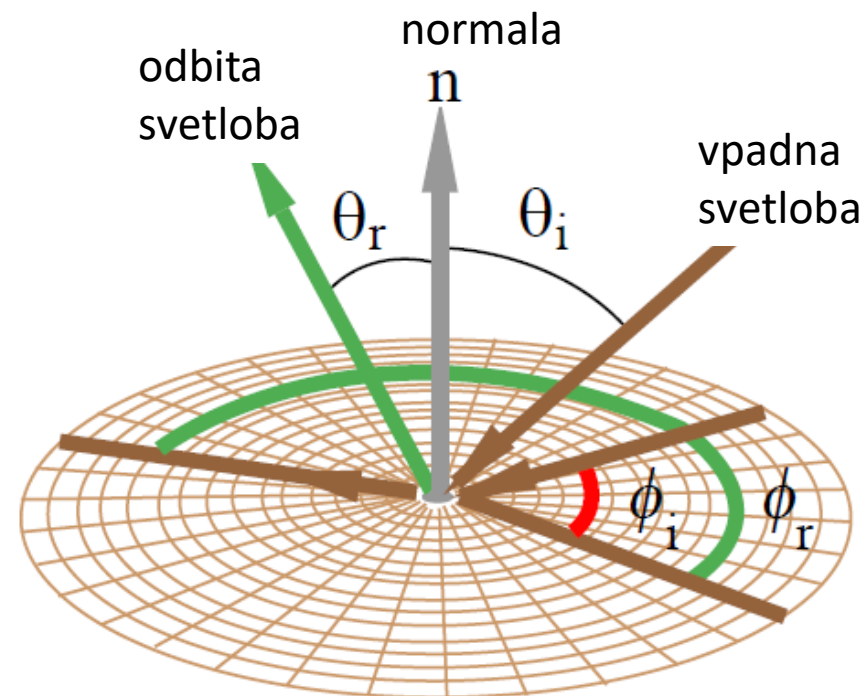


BRDF: *bidirectional reflection distribution function*

- Lambert, Phong, Blinn so enostavni modeli materialov
 - poznamo tudi bolj kompleksne, fizikalno osnovane npr. Oren-Nayar, Cook-Torrence, Ward itn.
- Posplošeno lahko odboj zapišemo s funkcijo: **BRDF**
 - za vsak par: smer vpadne svetlobe (luči) \mathbf{i} , in smeri odboja (gledalca) \mathbf{r} , določi koliko svetlobe se odbije do gledalca
- Prej omenjeni modeli so posebni (analitični) primeri BRDF



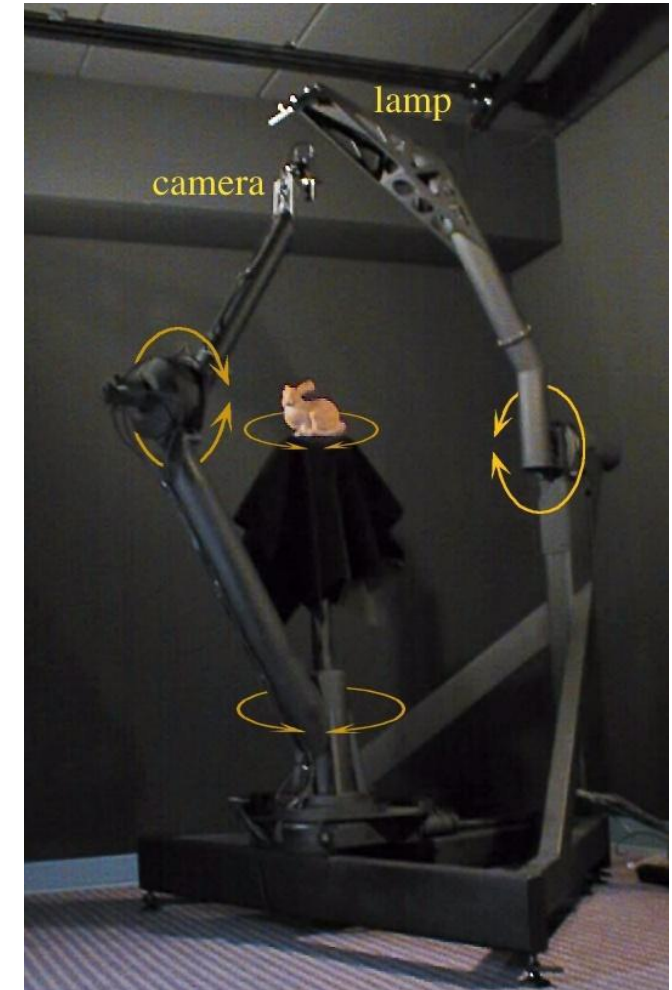
- BRDF je štiridimenzionalna funkcija kotov vpadne in odbite svetlobe
 - $BRDF(\theta_i, \phi_i, \theta_r, \phi_r)$
 - koti predstavljajo vektorja vpadne (θ_i, ϕ_i) in odbite (θ_r, ϕ_r) svetlobe v sferičnih koordinatah
 - fizikalno predstavlja delež sevalnosti, ki se odbije v smeri \mathbf{r} , glede na skupno obsevanost v točki odboja
- Odboj od luči c_i do opazovalca lahko zapišemo kot:
 - $c = c_i BRDF(\theta_i, \phi_i, \theta_r, \phi_r) \cos \theta_i$





BRDF

- Lahko jih izmerimo!
 - 4D tabela vrednosti
 - lahko se spreminjajo tudi v času – 5D
 - Knjižnice vrednosti
 - vsaka snov ima nekaj milijonov npr. $90 \times 90 \times 180 \times 5 = 4.374.000$ meritev
- Izmerjene vrednosti lahko parametriziramo, dobimo analitičen model
- Lahko jih narišemo!
- Lahko se jih učimo



Demo: Time varying BRDF, pokaže gonioreflectometer



- Izmerjene vrednosti lahko neposredno uporabimo pri upodabljanju



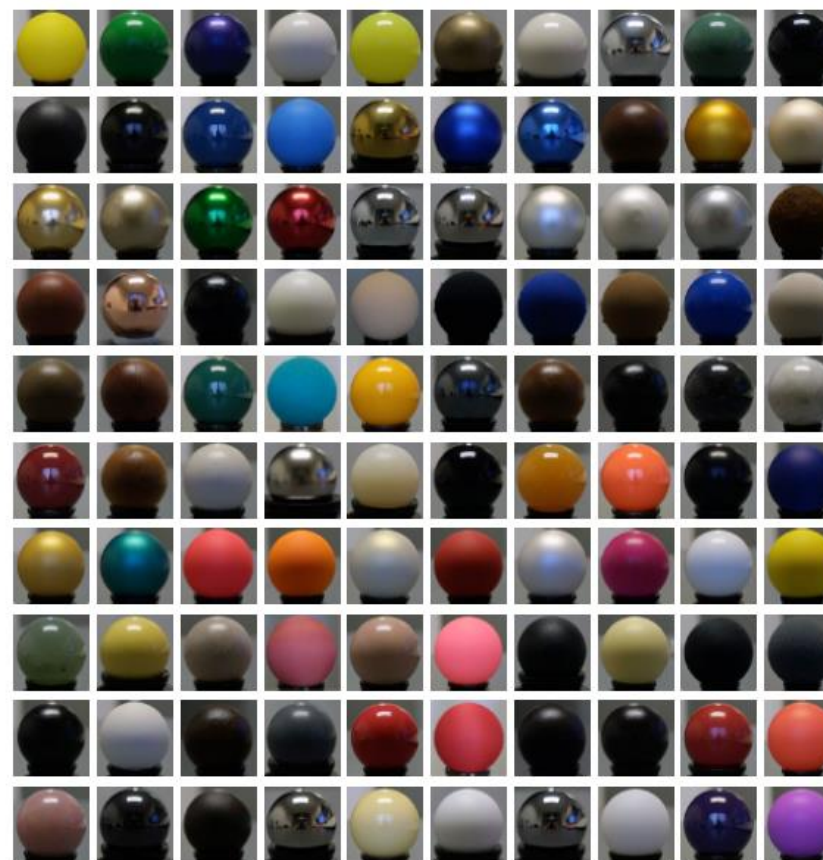
Nickel

Hematite



Gold Paint

Pink Felt





Luči



Luči

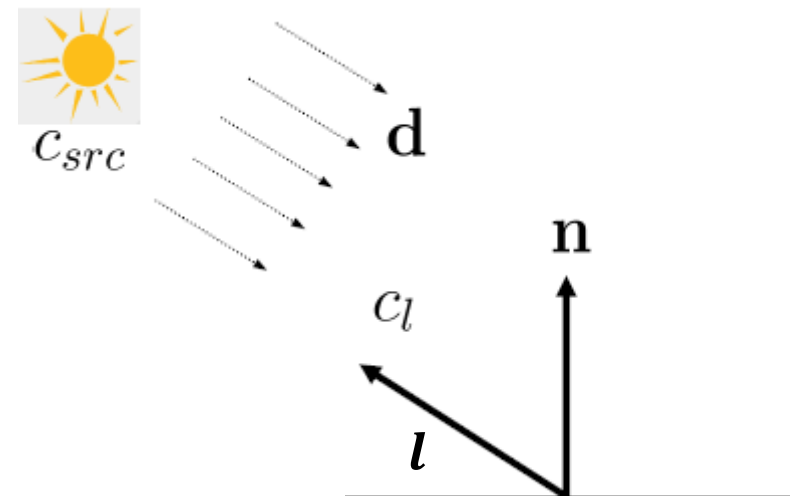
- Luči imajo lahko različne parametre
 - barvo
 - površino s katere sevajo svetlobo
 - usmerjenost
 - tudi površine, ki odbijajo svetlobo, so viri svetlobe
- Večina interaktivnih (npr. OpenGL) in bolj kompleksnih okolij podpira:
 - usmerjene luči (sonce)
 - točkaste luči (žarnica)
 - reflektorje
- Za implementacijo potrebujemo izračun vektorja proti luči \mathbf{l} in jakosti/barve s katero je točka osvetljena c_l
 - npr. Phong:
$$c = c_a k_a + c_l (k_d (\mathbf{l} \cdot \mathbf{n}) + k_s (\mathbf{R} \cdot \mathbf{e})^p)$$





- Izvor je zelo daleč
 - približek sonca
- Določata jo barva c_{src} in smer d
 - žarki svetlobe so si vzporedni - smer d je enaka
 - jakost/barva c_{src} je povsod enaka
- Implementacija
 - $l = -d$
 - $c_l = c_{src}$

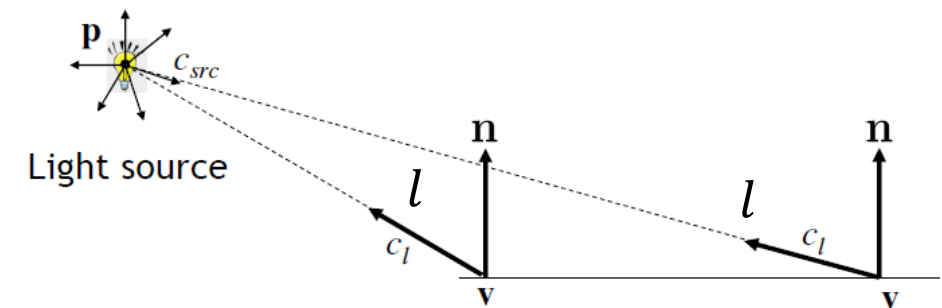
Usmerjena luč





- Enostaven model žarnice
- Svetlobo seva v vse smeri enako
- Kot vpadne svetlobe je odvisen od položaja \mathbf{p}
- Jakost pada s kvadratom razdalje
- Implementacija
 - $\mathbf{l} = \frac{\mathbf{p} - \mathbf{v}}{|\mathbf{p} - \mathbf{v}|}$
 - $c_l = \frac{c_{src}}{f(|\mathbf{p} - \mathbf{v}|^2)}$

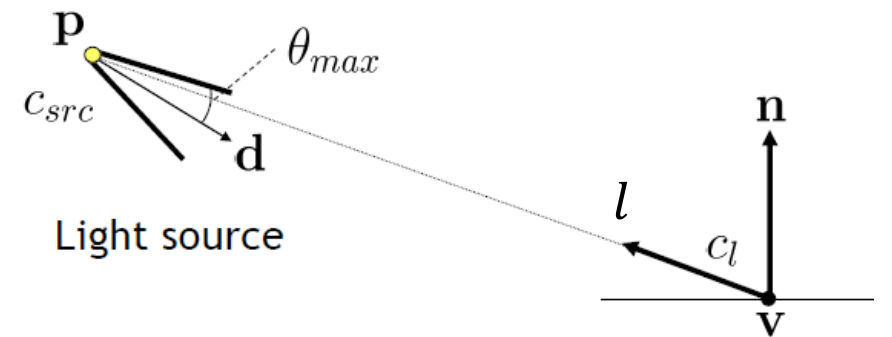
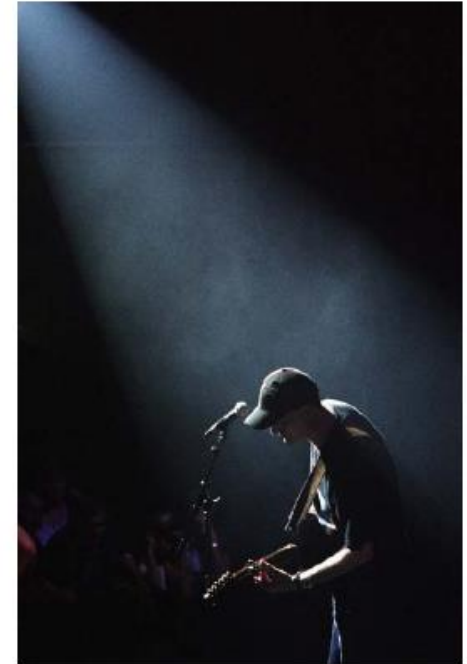
Točkasta luč





- Seva v neko smer \mathbf{d}
 - poleg položaja \mathbf{p} ima torej tudi usmeritev
- Jakost:
 - odvisna od širine **stožca** θ_{max}
 - pada proti robu stožca s potenco f
- Implementacija
 - $\mathbf{l} = \frac{\mathbf{p} - \mathbf{v}}{|\mathbf{p} - \mathbf{v}|}$
 - $c_l = \begin{cases} 0, & \text{če } -\mathbf{l} \cdot \mathbf{d} \leq \cos(\theta_{max}) \\ c_{src}(-\mathbf{l} \cdot \mathbf{d})^f, & \text{sicer} \end{cases}$

Reflektor





Osvetljevanje v grafičnem cevovodu



Osvetljevanje - implementacija

- Izberemo model materiala, npr. Phongov
 - $c = c_a k_a + c_l (k_d (\mathbf{l} \cdot \mathbf{n}) + k_s (\mathbf{R} \cdot \mathbf{e})^p)$
- Glede na izbrani model lahko definiramo lastnosti materiala kot uniforme, lastnosti oglišč ali jih podamo v ločeni teksturi
 - npr. jakost ambientnega (k_a), razpršenega (k_d) in zrcalnega odboja (k_s), koeficient zrcalnega odboja (p)
- Za izračun potrebujemo tudi pravilne **normale** v ogliščih
- Kdaj računamo osvetlitev?
 - v ogliščih (bolj učinkovito)
 - za vsak fragment/piksel poligona (lepše, več računanja)

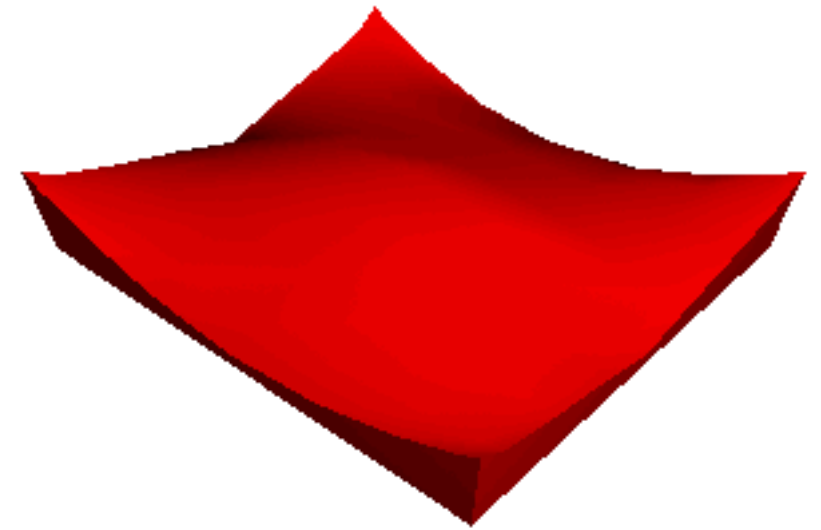




- Tudi
 - Gouraudovo senčenje (Henri Gouraud, 1971)
 - *per-vertex shading*
- Osvetlitev izračunamo v vsakem **oglišču** poligona
 - torej v senčilniku oglišč
- Barvo-svetlost prenesemo v senčilnik fragmentov preko interpolirank
 - se torej bilinearno interpolira v notranjosti trikotnika



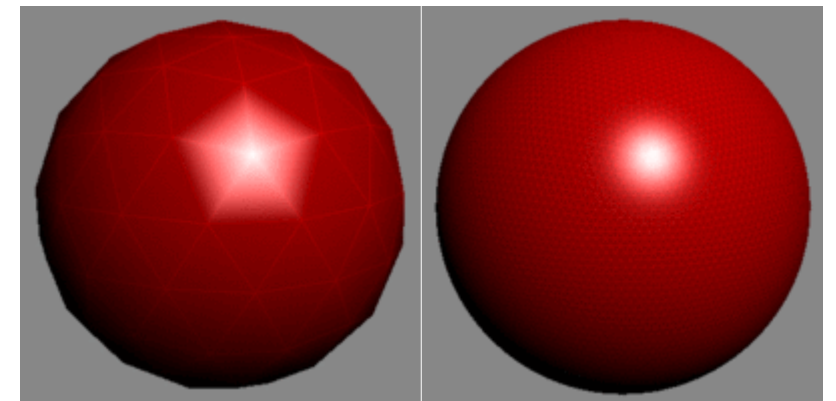
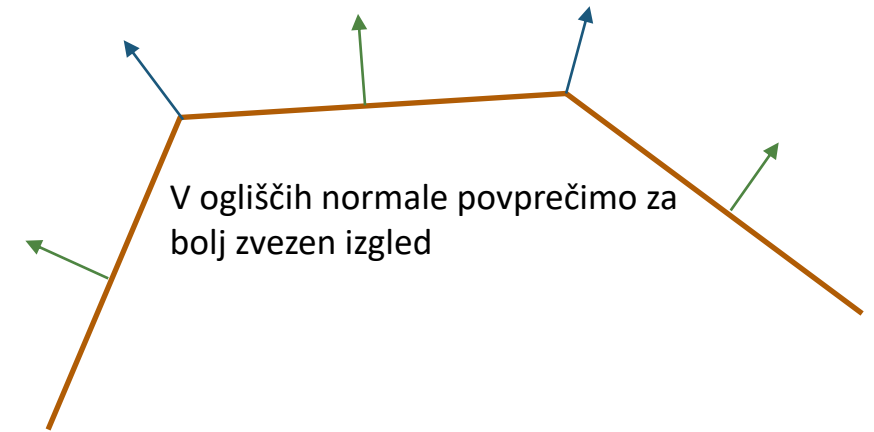
Osvetljevanje v ogliščih





- Za izračun osvetlitve potrebujemo **normale** v ogliščih
 - za mehke prehode med ploskvami normale postavimo na povprečje normal ploskev, ki se stikajo v oglišču
 - 3D modelirniki to navadno avtomatsko počnejo, lahko preklapljamo med flat in average normalami
- Gouraudovo senčenje je
 - hitro (osvetlitev računamo le v ogliščih)
 - kvaliteta je problematična predvsem, ko je število poligonov majhno
 - boljše rezultate dosežemo z večjim številom poligonov

Osvetljevanje v ogliščih



malo poligonov

veliko poligonov



- *Flat shading*
 - nezvezen izgled, morda to želimo
- Cel poligon je enako osvetljen
 - računamo na nivoju oglišč
 - v senčilniku oglišč ali geometrije
- Implementacija
 - normale oglišč morajo biti enake (pravokotne na trikotnik)
 - barvo iz senčilnika prenesemo preko interpoliranke s tipom interpolacije *flat*, da se ne interpolira med oglišči



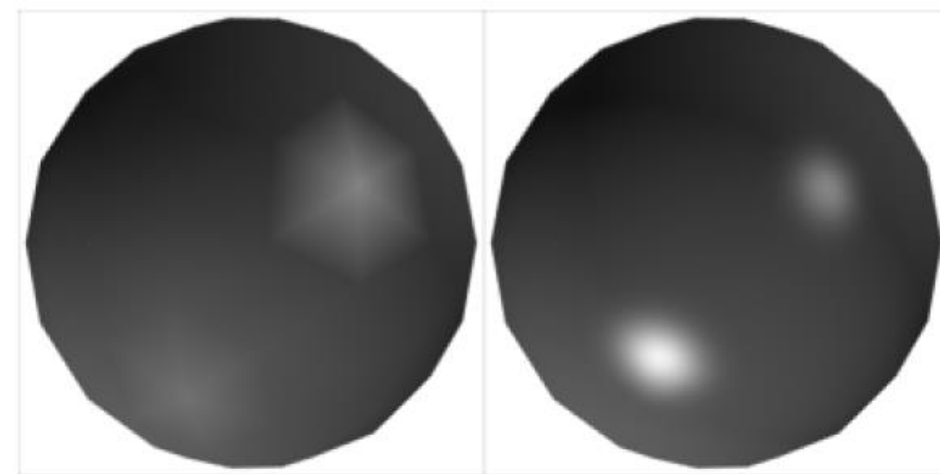
Plosko (konstantno) senčenje





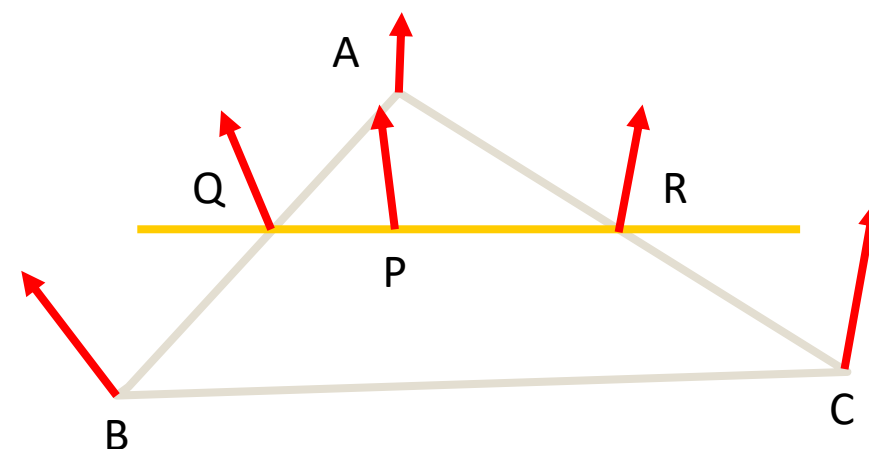
- Tudi
 - Phongovo (mehko) senčenje
 - *Per-pixel shading*
- Osvetlitev izračunamo v vsakem pikslu/fragmentu poligona
 - torej v senčilniku fragmentov
- Za izračun osvetlitve v točki, rabimo **normalo v vsakem fragmentu**
 - v senčilnik fragmentov normale prenesemo kot interpoliranke
- Boljša kvaliteta, precej mehkejši rezultat kot Gouraud
 - počasnejše, sploh če je v sceni več luči

Osvetljevanje v fragmentih



Gouraud

Phong





REFERENCE

- N. Guid: Računalniška grafika, FERI Maribor
 - J.D. Foley, A. Van Dam et al.: Computer Graphics: Principles and Practice in C, Addison Wesley
- 