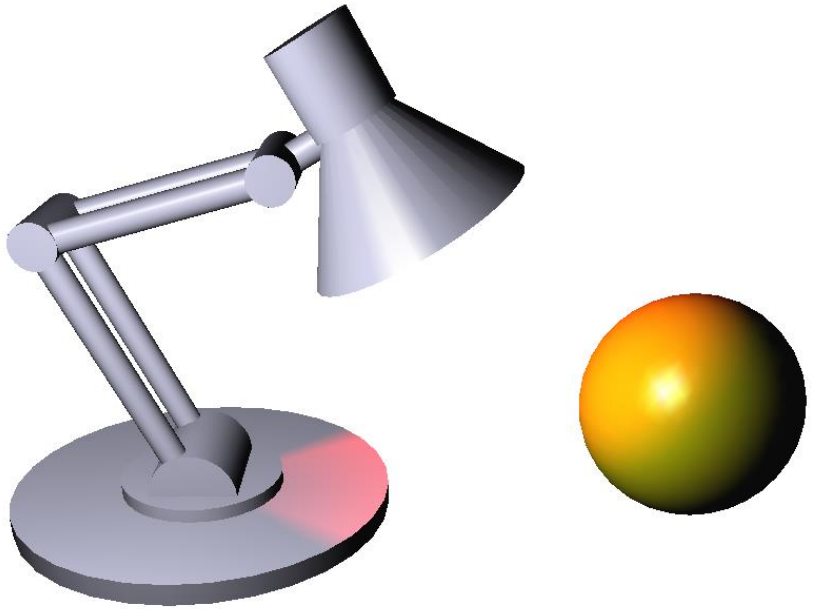
Osvjetljavanje



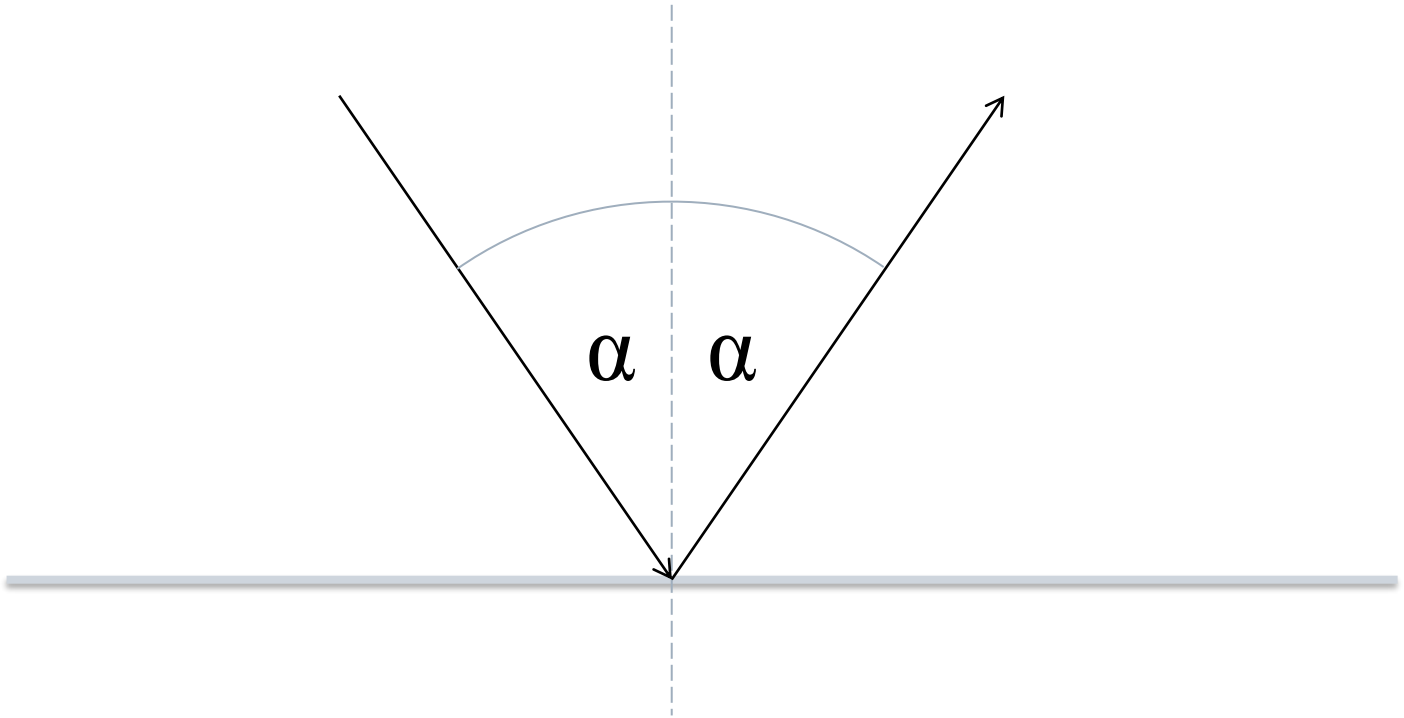
Osvjetljavanje
(engl. *lighting/illumination*)

- za dobivanje potpuno realističnih scena morali bi strogo poštivati fizikalne zakone – fizikalno utemeljeno iscrtavanje (engl. *physics based rendering*)
- najpoznatija je metoda praćenja zrake (engl. *ray tracing*), ali postoje i druge (*photon mapping, radiosity*)
- ove metode su vrlo zahtjevne što se tiče računalnih resursa, pa se za iscrtavanje u stvarnom vremenu uglavnom koriste aproksimacije, tj. pojednostavljeni modeli

Phongov model refleksije
(engl. Phong reflection model)

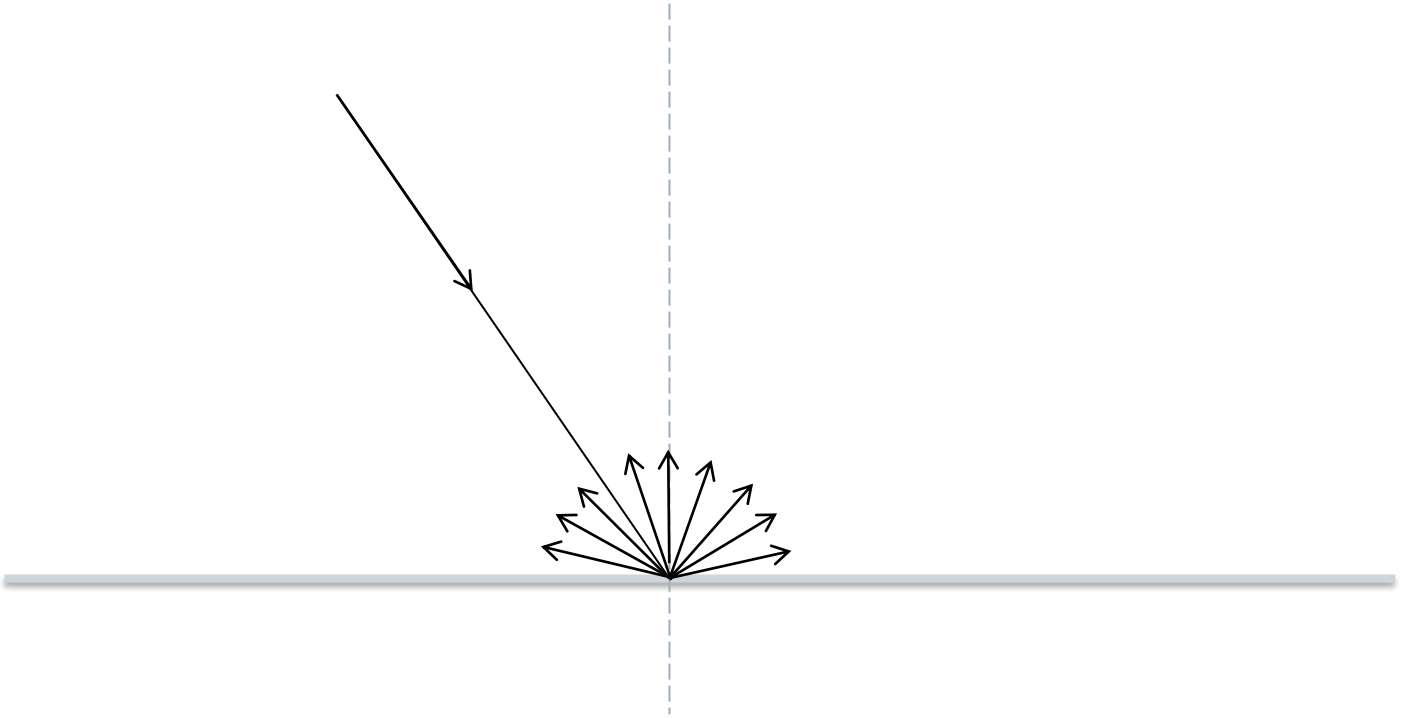
- bio je implementiran u „starom” OpenGL-u s fiksnim grafičkim protočnim sustavom
- refleksija (odbijanje) svjetlosti na površinama aproksimira se kombinacijom
  - difuzne (engl. *diffuse*) refleksije i
  - zrcalne (engl. *specular*) refleksije
- još se dodaje doprinos okolne svjetlosti (engl. *ambient light*) da bi se dobila konačna vrijednost intenziteta svake osnovne boje (crvene, zelene i plave), tj. konačna boja svakog piksela

Savršeno zrcalo



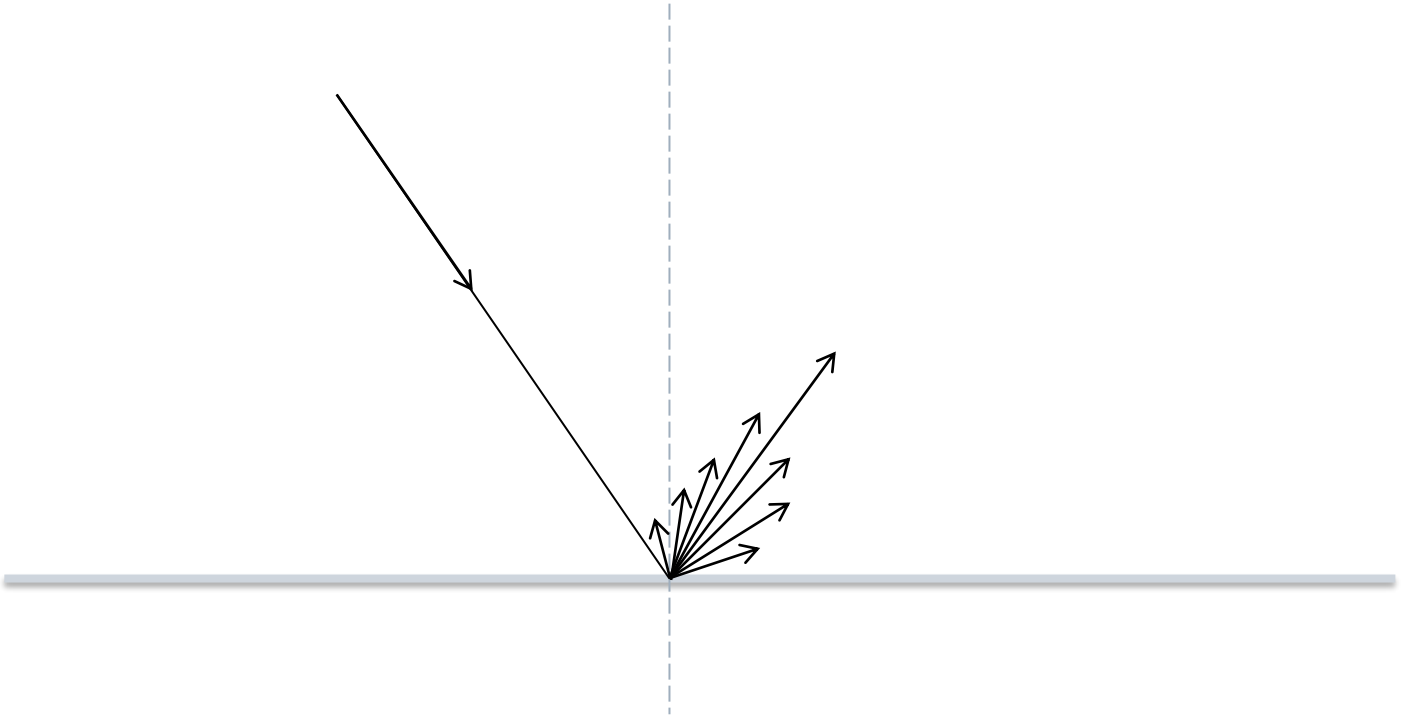
Zrcalna refleksija: kut refleksije jednak je kutu upada u odnosu na normalu

Potpuno difuzna površina



Svjetlost se raspršuje jednoliko na sve strane

Stvarna površina



Svjetlost se raspršuje približno pod upadnim kutem

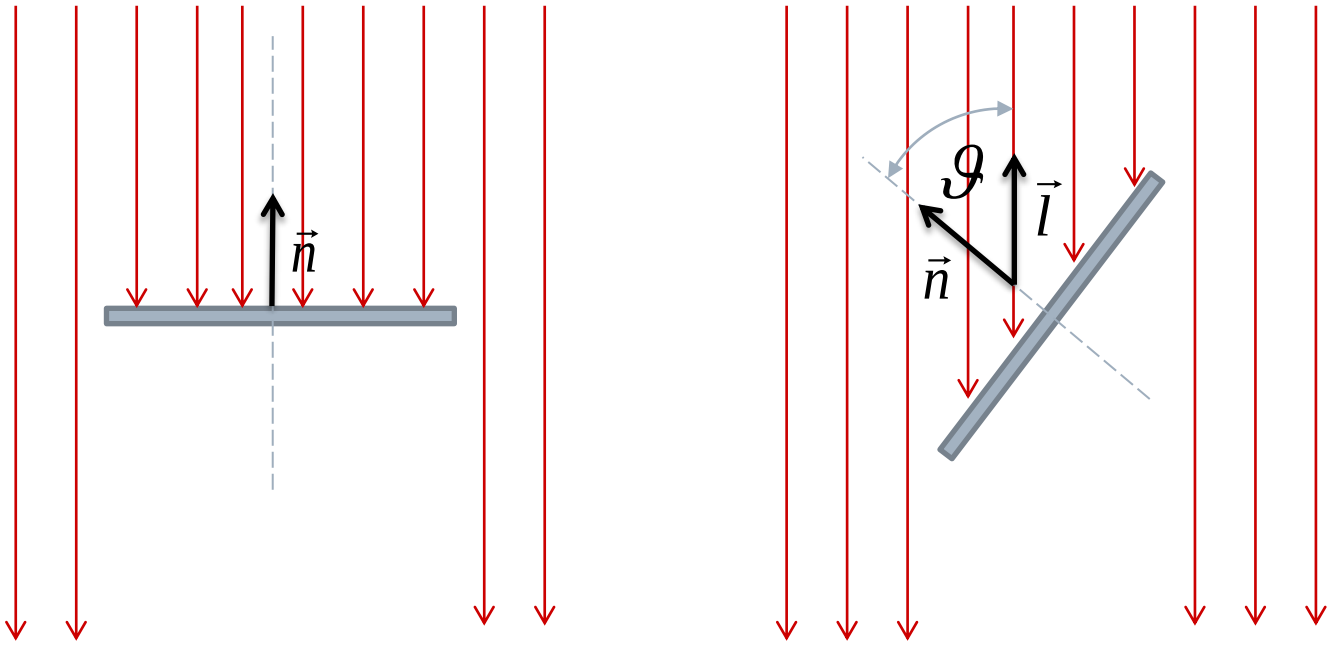
Difuzna refleksija

- ulazna svjetlost raspršuje se jednoliko u svim smjerovima
- međutim, važan je kut pod kojim svjetlost upada na površinu
- Lambertov zakon (zapravo slijedi iz zakona očuvanja energije):

> *Jakost rasvjete neke površine proporcionalna je kosinusu kuta što ga upadne zrake čine s okomicom na danu površinu.*
>
> (*Hrvatska enciklopedija, mrežno izdanje.* Leksikografski zavod Miroslav Krleža, 2020.)

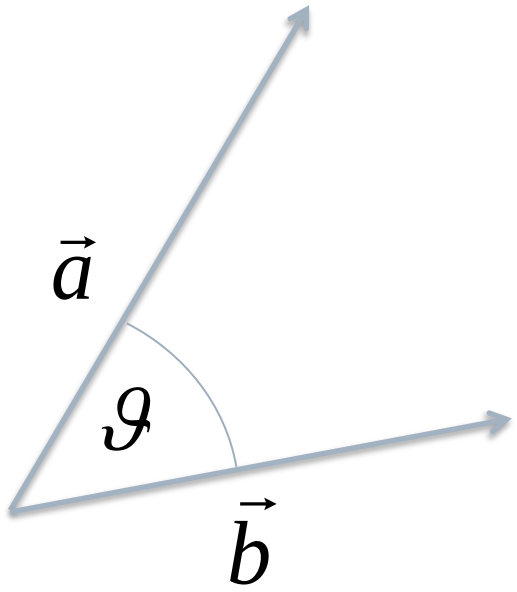
Lambertov (kosinusni) zakon

$\vec{n}$ - jedinični vektor normale $\vec{l}$ - jed. vektor usmjeren prema izvoru svjetlosti



$$I = I_0 \cos\vartheta = I_0 \, \vec{n} \cdot \vec{l} = I_0 |\vec{n}||\vec{l}| \cos\vartheta$$

Množenje vektora - skalarno



$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos \angle(\vec{a}, \vec{b})$$

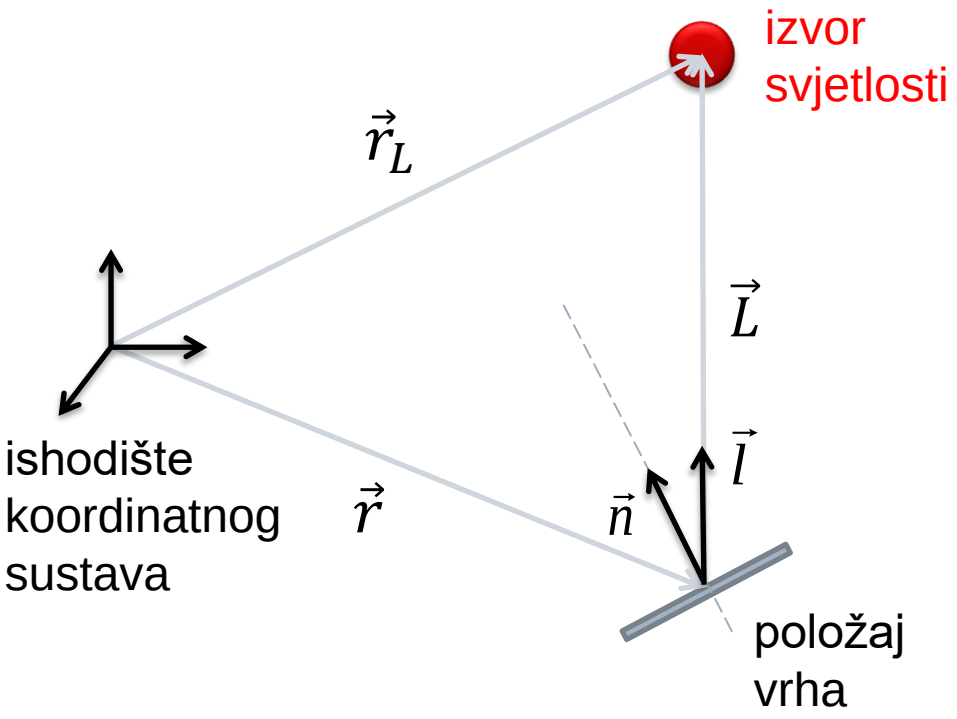
$$\vec{a} \cdot \vec{b} = a_x b_x + a_y b_y + a_z b_z$$

rezultat je broj (skalar)

Ako su vektori okomiti skalarni produkt je nula!

Jedinični vektor prema izvoru

$\vec{n}$ - jedinični vektor normale $\vec{l}$ - jed. vektor usmjeren prema izvoru svjetlosti



$$\vec{r} + \vec{L} = \vec{r}_L$$

$$\vec{L} = \vec{r}_L - \vec{r}$$

$$\vec{l} = \frac{\vec{L}}{|\vec{L}|}$$

Lambertov zakon - implementacija u programu za sjenčanje vrhova

```
#version 300 es
in vec4 a_vrhXYZ;
in vec3 a_normala;
uniform mat4 u_mTrans;
uniform vec3 u_izvorXYZ; // vektor položaja izvora
out float v_svjetlina; // prenosi se u fragment shader

void main() {
  vec4 vrh = u_mTrans * a_vrhXYZ; // transformacija vrha
  vec3 normala = mat3(u_mTrans) * a_normala; // transformacija normale

  // Lambertov zakon
  vec3 premaIzvoru = normalize(u_izvorXYZ - vec3(vrh));
  v_svjetlina = dot(premaIzvoru, normala);

  gl_Position = vrh;
}
```

Program za sjenčanje fragmenata

```
#version 300 es
precision highp float;
uniform vec3 u_boja;
in float v_svjetlina;
out vec4 bojaFragmenta;

void main() {
  // svjetlina se kombinira sa zadanom bojom
  bojaFragmenta = vec4(v_svjetlina * u_boja, 1.0);
}
```

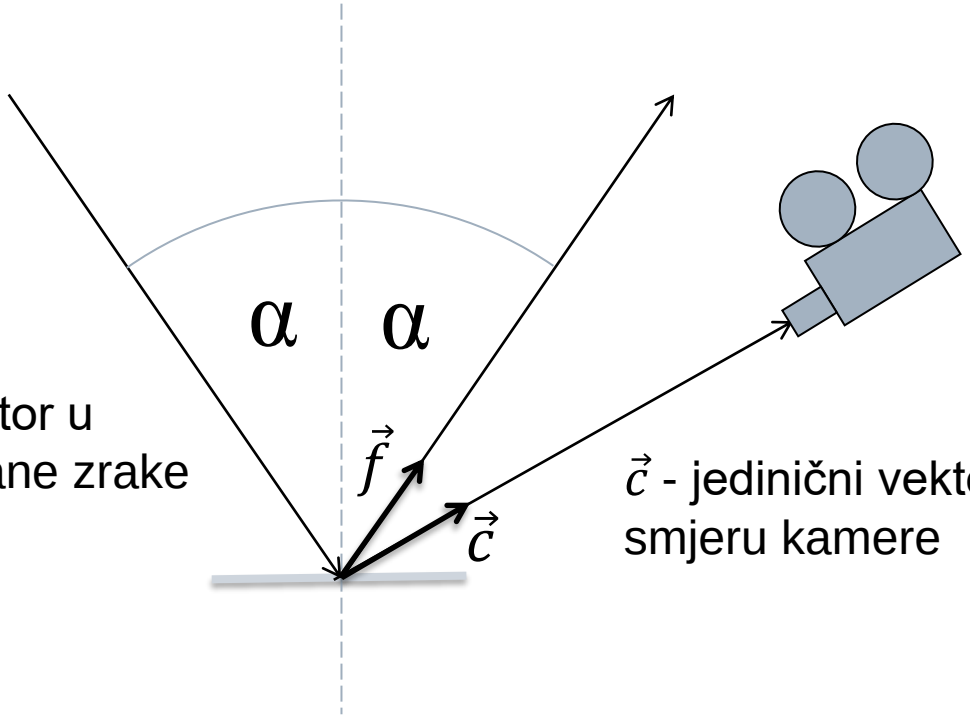
Gouraudovo vs. Phongovo sjenčanje

- sa stanovišta matematike normala je određena u odnosu na plohu pa nije logično normalu pridružiti vrhu – takvo pridruživanje nije jednoznačno određeno
- to je, međutim, praktično jer se svjetlina može interpolirati za svaku točku trokuta – Gouraudovo sjenčanje (1971.)
- bolje rezultate daje interpolacija normale (Phong, 1973.) – pogotovo za zrcalnu refleksiju

Zrcalna refleksija

- kod idealnog zrcala vrijedi zakon refleksije: kut refleksije jednak je kutu upadne zrake svjetlosti u odnosu na normalu
- mnogi materijali (posebno metali i glatke površine) imaju svojstvo zrcala, tj. najveći dio svjetlosti reflektira se pod kutom refleksije
- međutim, oni ipak nisu savršena zrcala, što više odstupaju od idealnog zrcala, to je veće rasipanje oko idealnog kuta refleksije

Zrcalna refleksija



$\vec{f}$ - jedinični vektor u smjeru reflektirane zrake

$\vec{c}$ - jedinični vektor u smjeru kamere

Intenzitet ovisi o kutu između reflektirane zrake i položaja kamere

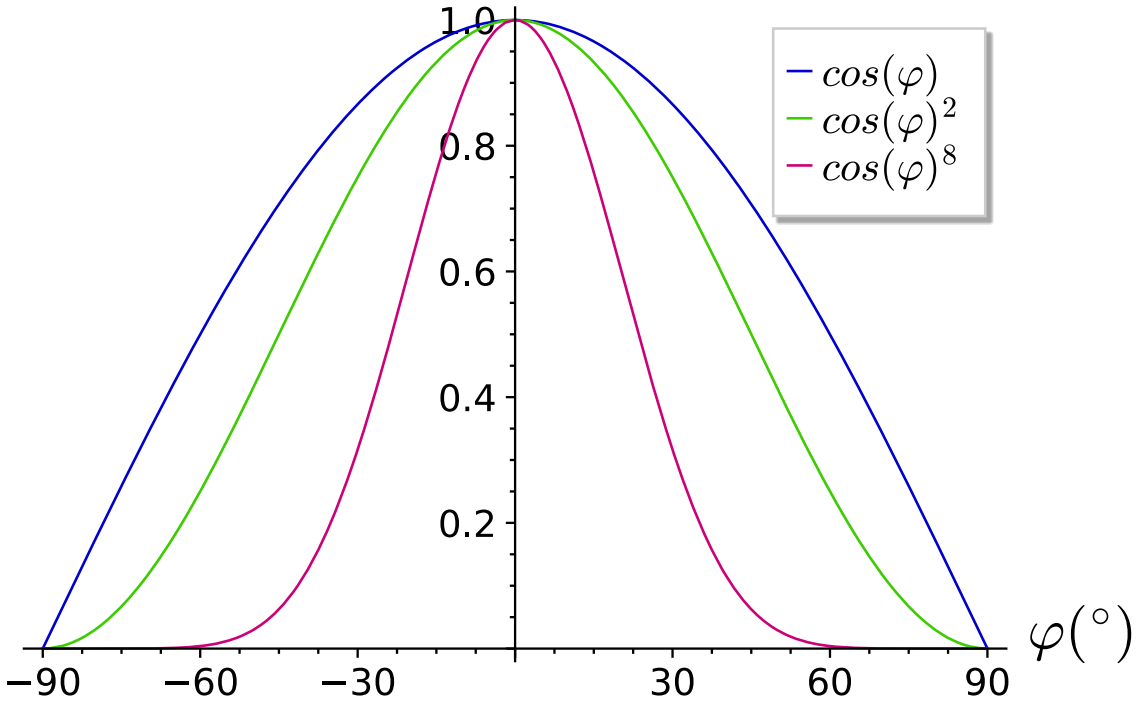
Zrcalna refleksija

- intenzitet reflektirane svjetlosti ovisi o kutu između reflektirane zrake i položaja kamere

$$I = I_0 \left(\vec{f} \cdot \vec{c} \right)^n$$

- da bi se smanjio kutni raspon pod kojim je zrcalna refleksija bliska jedinici koristi se eksponent n koji efektivno predstavlja sjajnost materijala

Potencije kosinusa



Implementacija u programu za sjenčanje vrhova

```
void main() {
  vec4 vrh = u_mTrans * a_vrhXYZ; // transformacija vrha
  vec3 normala = mat3(u_mTrans) * a_normala; // transformacija normale

  // Lambertov zakon
  vec3 premaIzvoru = normalize(u_izvorXYZ - vec3(vrh));
  v_svjetlina = dot(premaIzvoru, normala);

  // zakon refleksije
  vec3 premaKameri = normalize(u_kameraXYZ - vec3(vrh));
  vec3 reflektiranaZraka = reflect(-premaIzvoru, normala);
  float refleksija = max(dot(reflektiranaZraka, premaKameri), 0.0);
  refleksija = pow(refleksija, 8.0);

  v_svjetlina = v_svjetlina * 0.5 + refleksija * 0.5;
  gl_Position = vrh;
}
```

Okolna svjetlost (*ambient light*)

- dio svjetlosti ne dolazi direktno iz izvora svjetlosti, već se odbija od zidova, stropova, namještaja i ostalih predmeta (taj efekat je posebno izražen u zatvorenim prostorima)
- detaljan proračun od kuda je došao svaki foton je nemoguć, pa se okolna svjetlost aproksimira kao određeni dio (na primjer jedna petina) intenziteta izvora svjetlosti

Okolna svjetlost

- pošto dolazi sa svih strana, nema nikakve ovisnosti o kutu upada, pa scene osvijetljene samo okolnim (globalnim) osvjetljenjem ne ostavljaju uvjerljiv dojam trodimenzionalnosti
- iz tog razloga treba izbjegavati korištenje okolnog svjetla kao zamjenu za pravo osvjetljenje – međutim, okolno svjetlo je korisno, jer ipak daje barem obrise predmeta koji nisu direktno osvijetljeni

Definicija svojstava materijala

- moramo definirati što se događa sa svjetlošću koje pada na površinu – možemo odrediti svojstva za:
  - difuznu refleksiju
  - zrcalnu refleksiju
  - okolnu svjetlost
- svojstvo se definira za svaku od tri osnovne boje posebno (na primjer, crvenu površinu dobit ćemo ako definiramo da odbija samo crvenu komponentu boje)

Phongov model osvjetljavanja

(engl. *Phong lighting/illumination*)

- intenzitet svjetlosti u pojedinom vrhu (ili fragmentu) računa se kao suma intenziteta difuzne (I_d), zrcalne (I_s) i okolne svjetlosti (I_a) pomnoženih odgovarajućim koeficijentima (k_d, k_s i k_a) koji modeliraju svojstva materijala

$$I = I_d k_d \left(\vec{n} \cdot \vec{l}\right) + I_s k_s \left(\vec{f} \cdot \vec{c}\right)^n + I_a k_a$$

- intenzitet se računa posebno za crvenu, zelenu i plavu boju (koeficijenti mogu biti različiti)

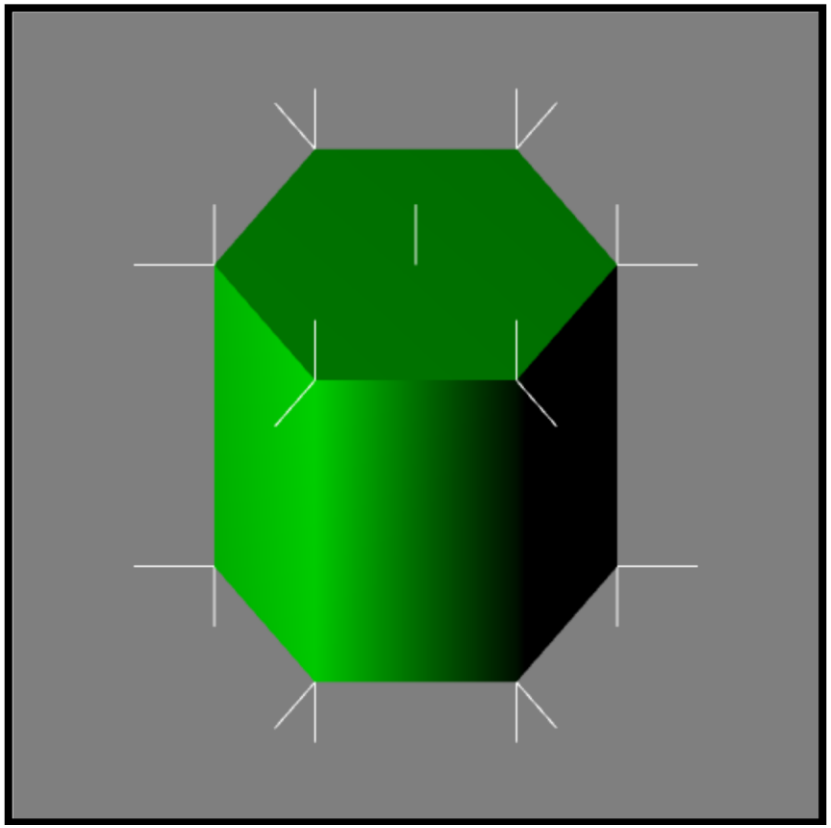
Vektori normale

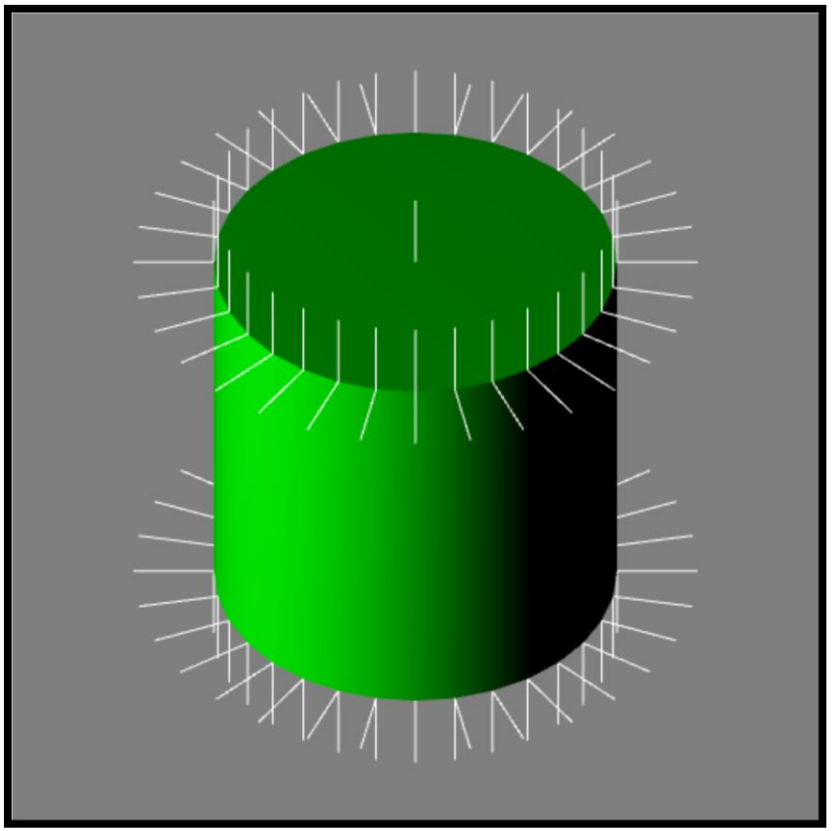
- proračuni difuznog i zrcalnog odbijanja svjetlosti zahtijevaju da se svakom vrhu pridruži vektor normale
- ispravno definirani vektori normale su nužni za funkcioniranje osvjetljavanja!
- vektore normale treba normirati i proslijediti u program za sjenčanje vrhova putem `attribute` varijable

Kako odrediti vektore normale?

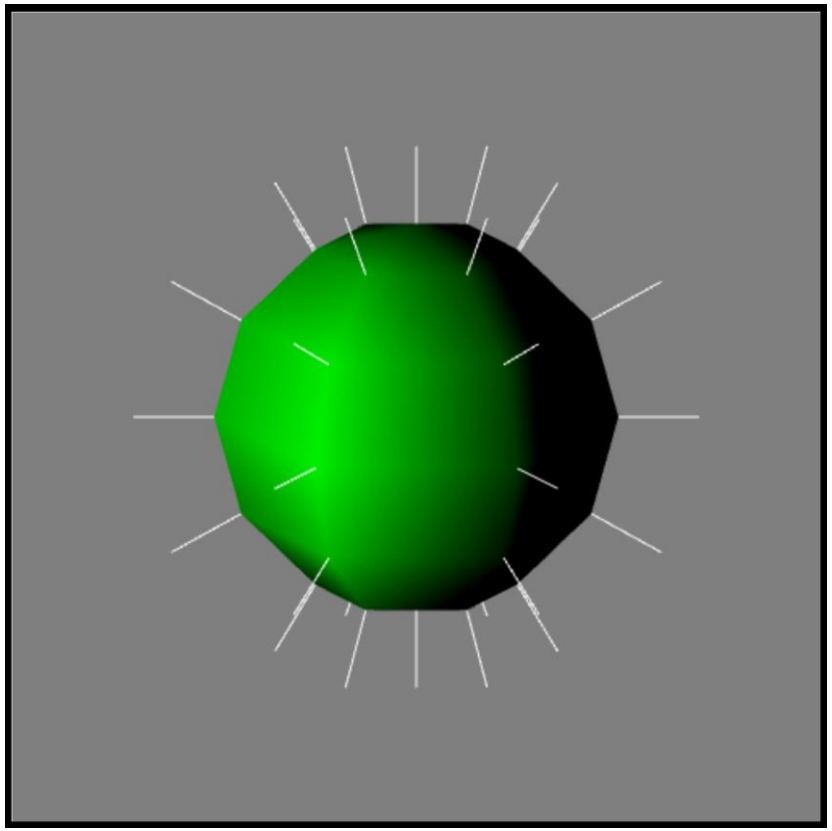
- ako postoji, treba iskoristiti simetriju!
- može pomoći i vektorski produkt: dva vektora koji nisu kolinearni definiraju ravninu, a njihov vektorski produkt je po definiciji okomit na tu ravninu
- odabir normala nije uvijek jednoznačan – primjer stošca

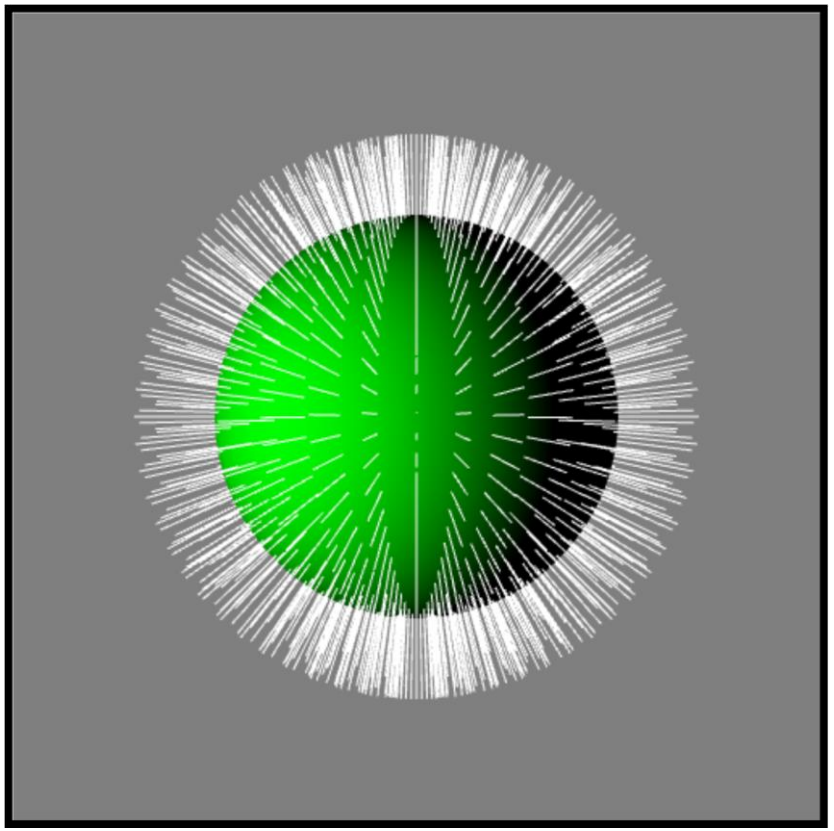
Primjer 1: Vektori normale za valjak



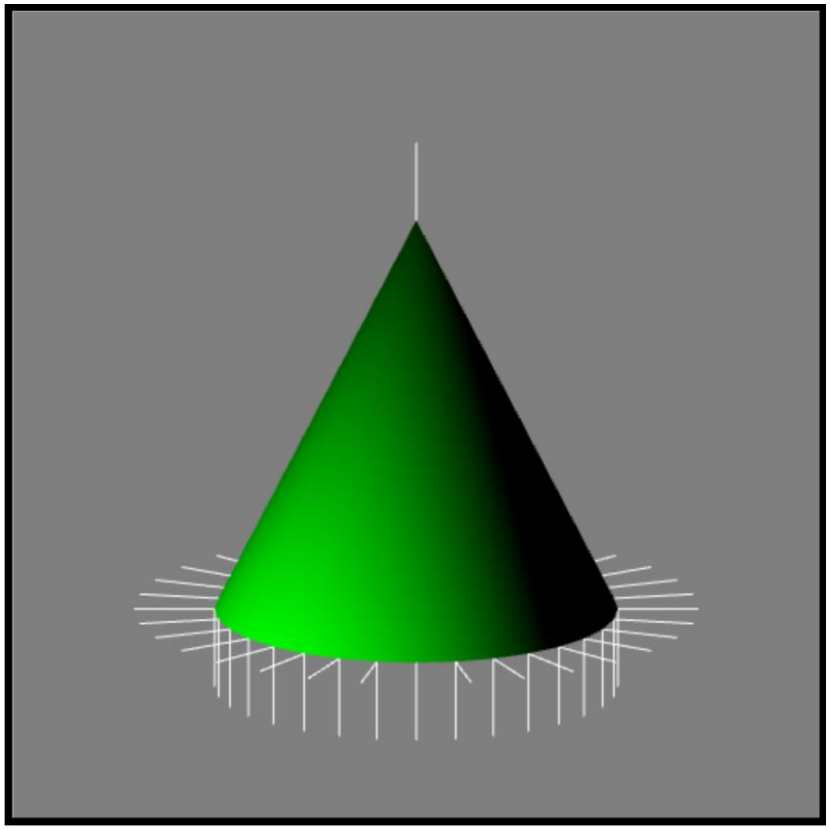


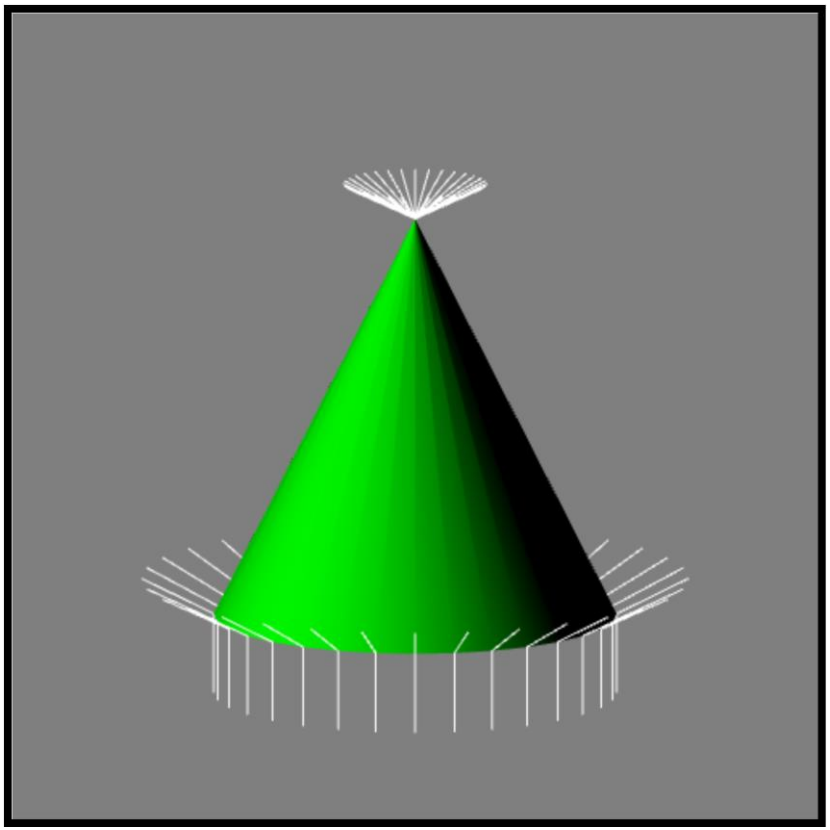
Primjer 2: Vektori normale za kuglu





Primjer 3: Vektori normale za stožac





Transformacije vektora normale

- tako dugo dok je matrica transformacije ortogonalna može se koristiti i za transformaciju normala
- ako matrica transformacije nije ortogonalna ona više ne čuva okomitost, tj. normale više nisu okomite pa više nisu ni normale
- problem su nejednaka skaliranja i smicanje
- u tom slučaju moramo izračunati posebnu matricu transformacije za normale invertiranjem i transponiranjem matrice transformacije za vrhove