

# Računarska grafika

Svetlost i senčenje

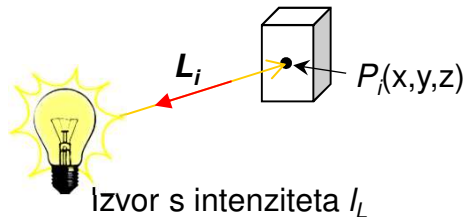


# Uvod

- Svetlost i senčenje izuzetno doprinose realističnosti slike
- Model izvora svetla – model upadnog svetla
  - izvori: ambijentalno svetlo, usmereno svetlo, tačkast izvor, reflektor
- Model osvetljenja (iluminacije) – model odbijanja (refleksije) svetla
  - lokalno (direktno) osvetljenje
  - globalno (indirektno) osvetljenje
- Modeli lokalnog osvetljenja (ambijentalno, difuzno, reflektivno)
  - relativno brzo izračunavanje
  - *JavaFX* podržava samo lokalno osvetljenje
- Modeli (algoritmi) senčenja (nijansiranja):
  - ravnomerno (*flat*) senčenje
  - Guroovo (*Gouraud*) senčenje
  - Fongovo (*Phong*) senčenje

# Izvori svetla

- Izvor se karakteriše energijom ( $E$ ) i talasnom dužinom ( $\lambda$ ) zračenog svetla
  - talasna dužina određuje boju svetla
- Intenzitet svetla izvora:  $I_L(E, \lambda)$ 
  - u računarskoj grafici se modelira komponentama modela boja (R,G,B)
- Model upadnog svetla koje potiče od datog izvora  $s$ , u tački scene  $P_i(x,y,z)$ :  
 $M_S(P_i, \mathbf{L}_i, I_L, \dots)$ 
  - $P_i$  je osvetljena tačka sa koordinatama  $(x,y,z)$  na nekoj površi (objekta)
  - $\mathbf{L}_i$  je jedinični vektor smera upadnog svetla u tački  $P_i$
  - $I_L$  je intenzitet svetla iz izvora  $s$  (modeliran R,G,B komponentama)



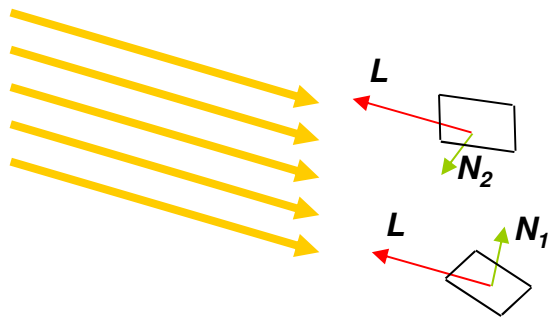
# Ambijentalno svetlo

- Objekat koji nije direktno osvetljen je i dalje vidljiv
- Svetlost koja ga obasjava potiče od ambijentalne refleksije
  - ambijentalna refleksija – odbijanje svetla od drugih objekata u sceni
- Modeluje se izvorom ambijentalnog svetla
  - ne računaju se odsjaji od pojedinih površi
  - samo se specificira konstantan intenzitet svetla za sve površi u sceni
  - definiše se pomoću RGB komponenti ambijentalnog svetla intenziteta  $I_L$
- Model upadnog svetla u tački  $P_i$  koje potiče od ambijentalnog izvora svetla intenziteta  $I_L$ :  
$$M_a(P_i) = f(I_L): \{I_U = I_L\}$$
- U svakoj tački  $P_i$  scene intenzitet upadnog svetla  $I_U$  je  $I_L$  (nema slabljenja)
- Takođe, u svakoj tački  $P_i$  scene intenzitet odbijenog svetla je  $I_O = I_U = I_L$

# Usmereno (*directional*) svetlo

- Izvor svetla dovoljno dalek da se svi zraci smatraju paralelnim
- Na primer, sunce kao izvor svetlosti
- Definiše se intenzitetom emitovanih RGB komponenti svetla  $I_L$  i:
  - vektorom smera svetla  $L$
- Smer svetla  $L$  je bitan za modelovanje odsjaja od površi
- Model upadnog svetla u tački  $P_i$  koje potiče od usmerenog svetla intenziteta  $I_L$  i smera  $L$ :

$$M_d(P_i) = f(\mathbf{L}, I_L): \{I_U = I_L, \mathbf{L}\}$$



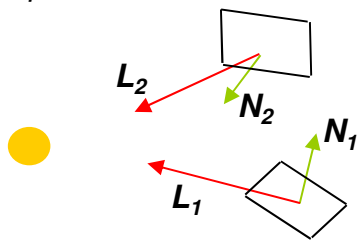
Intenzitet odbijenog svetla  $I_O$  u tački  $P_i$ :  
 $I_O \sim \mathbf{N}_i \cdot \mathbf{L}$

$\mathbf{N}_i$  različito za svaku površ  
 $L$  konstantno za sve površi

# Tačkasto (*point*) svetlo

- Relativno blizak izvor svetla koji radijalno širi zrake u svim pravcima
- Na primer, sijalica ili sveća
- Definiše se intenzitetom emitovanih RGB komponenti svetla  $I_L$  i:
  - tačkom u kojoj se nalazi izvor  $S(x,y,z)$
  - faktorima konstantnog  $k_c$ , linearnog  $k_l$  i kvadratnog  $k_q$  slabljenja sa rastojanjem  $d$  između izvora u tački  $S$  i tačke na površi objekta  $P_i$
- Model upadnog svetla u tački  $P_i$  koje potiče od tačkastog izvora na poziciji  $S$ , intenziteta  $I_L$  i faktorima slabljenja  $k_c, k_l, k_q$ :

$$M_p(P_i) = f(P_i, S, k_c, k_l, k_q, I_L): \{d(P_i, S), \mathbf{L}_i = \mathbf{v}(P_i, S), I_U = I_L / (1 + k_c + k_l \cdot d + k_q \cdot d^2)\}$$



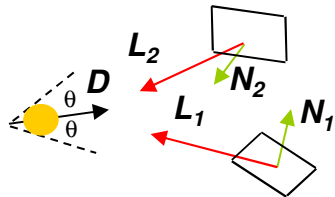
Intenzitet odbijenog svetla  $I_O$  u tački  $P_i$ :

$$I_O \sim \mathbf{N}_i \cdot \mathbf{L}_i$$

$\mathbf{N}_i$  i  $\mathbf{L}_i$  različito za svaku površ

# Reflektorsko (*spot*) svetlo

- Izvor svetla zrači tako da intenzitet opada sa povećanjem odklona zraka od vektora usmerenja reflektora  $\mathbf{D}$
- Na primer, reflektorska sijalica (reflektor), baterijska lampa
- Može da bude ograničeno zračenje na kupu sa osom na pravcu  $\mathbf{D}$
- Definiše se intenzitetom emitovanih RGB komponenti svetla  $I_L$  i
  - tačkom u kojoj se nalazi izvor  $S(x,y,z)$
  - vektorom usmerenja reflektora  $\mathbf{D}$
  - faktorima konstantnog  $k_c$ , linearnog  $k_l$  i kvadratnog  $k_q$  slabljenja sa rastojanjem  $d$  između izvora u tački  $S$  i tačke na površi  $P_i$
- Model upadnog svetla u tački  $P_i$  koje potiče od reflektorskog svetla intenziteta  $I_L$ , usmerenja  $\mathbf{D}$  i faktorima slabljenja  $k_c, k_l, k_q$ :
 
$$M_r(P_i) = f(P_i, S, \mathbf{D}, k_c, k_l, k_q, I_L) : \{d(P_i, S), \mathbf{L}_i = \mathbf{v}(P_i, S), I_U = I_L (-\mathbf{D} \cdot \mathbf{L}_i) / (1 + k_c + k_l \cdot d + k_q \cdot d^2)\}$$



Intenzitet odbijenog svetla  $I_O$  u tački  $P_i$ :

$$I_O \sim (-\mathbf{D} \cdot \mathbf{L}_i)(\mathbf{N}_i \cdot \mathbf{L}_i)$$

Opciono: ako  $\cos(\theta) > -\mathbf{D} \cdot \mathbf{L}_i \Rightarrow$  ignoriši svetlo

# Izvori svetla u *JavaFX*

- Svetla su objekti potomaka apstraktne klase `LightBase`
- Klasa `LightBase` je potomak klase `Node`
  - svetla se kao i drugi čvorovi (na primer, geometrijski oblici) dodaju sceni
- Svakom objektu svetla može da se
  - zada boja
  - upravlja njegovim stanjem: uključeno/isključeno
- Po stvaranju svetlo je podrazumevano bele boje i uključeno
- Metodi kojima može da se menja i dohvata boja svetla:
  - `setColor(Color)` i `getColor()`
- Metodi za uključivanje/isključivanje svetla i dohvatanje stanja svetla:
  - `setLightOn(boolean)` i `isLightOn()`



# Vrste izvora svetla u *JavaFX*

- *JavaFX* podržava četiri vrste svetala:
  - ambijentalno (klasa `AmbientLight`)
  - usmereno (klasa `DirectionalLight`)
  - tačkasto (klasa `PointLight`)
  - reflektorsko (klasa `SpotLight`)
- Sve klase su potomci klase `LightBase`
  - `AmbientLight`, `DirectionalLight` i `PointLight` su direkno izvedene
  - `SpotLight` je izvedena klasa iz `PointLight`
- U sceni može da bude proizvoljan broj svetala svake vrste
- U sceni se podrazumeva (ako se ne doda ni jedno svetlo) tačkasti izvor svetla na poziciji kamere (*headlight*)

# Osobine vrsta izvora svetla

- **Ambijentalno svetlo**
  - po prirodi difuzno (dolazi iz svih i odbija se u svim pravcima)
  - jednak mu je intenzitet na svim površima svih oblika u sceni
  - transformacije izvora nemaju smisla
- **Usmereno svetlo**
  - paralelni zraci
  - ima vektor usmerenja
  - nema slabljenja
- **Tačkasto svetlo**
  - izvor ima svoje mesto u 3D sceni
  - izvor radijalno širi zrake svetla
  - svetlo slabi sa rastojanjem od izvora (konstantno, linearno i kvadratno slabljenje)
- **Reflektorsko svetlo**
  - kao i tačkasto, ali ima i radijalno slabljenje i sa otklonom od usmerenja reflektora
  - dve kupe – unutar unutrašnje nema slabljenja, izvan spoljne nema svetla

Svetlo/Osobina	Mesto	Usmerenje
<b>Ambijentalno</b>	-	-
<b>Usmereno</b>	-	+
<b>Tačkasto</b>	+	-
<b>Reflektorsko</b>	+	+

# Liste objekata na koje utiče svetlo

- Svetlu mogu da se pridruže oblici na koje svetlo utiče
  - razdvaja se uticaj različitih svetala na različite oblike u sceni
- Objekat svetla održava listu pridruženih čvorova na koje se dato svetlo primenjuje
  - tip liste: `ObservableList<Node>`
- Metod `getScope()` objekta svetla vraća listu pridruženih čvorova
  - na tekuću listu se dodaju novi čvorovi scene na koje dato svetlo ima uticaj
  - `svetlo.getScope().addAll(čvor1, ..., čvorN);`
- Ukoliko je u datoj listi čvor tipa `Group`
  - dato svetlo se primenjuje i na sve čvorove članove grupe, rekurzivno
- Ukoliko je lista prazna, svetlo utiče na sve objekte u sceni

# Ambijentalno i usmereno svetlo

- Objekti ambijentalnog i usmerenog svetla se stvaraju
  - podrazumevanim konstruktorom ili
  - konstruktorom čiji je jedini parametar boja svetla:  
`AmbientLight(Color boja);`  
`DirectionalLight(Color boja);`
- Usmereno svetlo poseduje svojstvo `direction`
  - predstavlja njegovo usmerenje
  - postavlja se metodom `setDirection(Point3D)`
  - tačka tipa `Point3D` (iz paketa `javafx.geometry`)
    - predstavlja vrh vektora čija je napadna tačka u koordinatnom početku
    - konstruktor: `Point3D(double x, double y, double z)`
  - vektor ne mora da bude normalizovan (intenzitet ne mora da bude 1)
  - podrazumevano usmerenje je `Point3D(0, 0, 1)`

# Tačkasto svetlo

- Objekti tačkastog izvora svetla se stvaraju
  - podrazumevanim konstruktorom
  - konstruktorom čiji je parametar boja svetla: `PointLight (Color boja)`
- Nema usmerenje, ali ima podužno slabljenje i doimet
- Svojstva tačkastog svetla (podrazumevana vrednost – nema slabljenja)
  - `constantAttenuation` – koeficijent konstantnog slabljenja (1.0)
  - `linearAttenuation` – koeficijent linearnog slabljenja (0.0)
  - `quadraticAttenuation` – koeficijent kvadratnog slabljenja (0.0)
  - `maxRange` – **domet svetla** (`Double.POSITIVE_INFINITY`)
- Koeficijent podužnog slabljenja se računa po formuli:

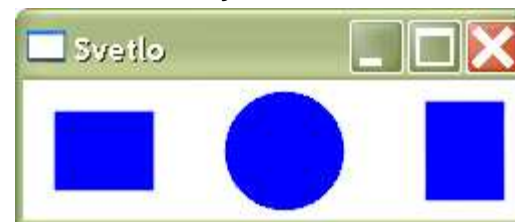
$$k_s(d) = k_c + k_l \cdot d + k_q \cdot d^2$$

# Primer

```
import javafx.scene.PointLight;
import javafx.scene.AmbientLight;
...
    AmbientLight a = new AmbientLight();
    a.setColor(Color.BLUE);
    PointLight s1 = new PointLight(); s1.setColor(Color.CYAN);
    s1.setTranslateX(0); s1.setTranslateY(35); s1.setTranslateZ(-20);
    PointLight s2 = new PointLight(); s2.setColor(Color.RED);
    s2.setTranslateX(250); s2.setTranslateY(70); s2.setTranslateZ(10);
    Group koren = new Group();
    koren.getChildren().addAll(kvadar, lopta, valjak, a, s1, s2);
```



Samo ambijentalno svetlo:

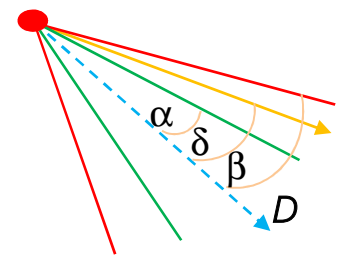


# Reflektorsko svetlo

- Objekti reflektorskog svetla se stvaraju
  - podrazumevanim konstruktorom
  - konstruktorom čiji je parametar boja svetla: `SpotLight (Color boja)`
- Osim podužnog, postoji radijalno slabljenje određeno svojstvima:
  - `innerAngle` – ugao unutrašnje kupe u stepenima ( $\alpha$ , podrazumevano  $0^\circ$ )
  - `outerAngle` – ugao spoljašnje kupe u stepenima ( $\beta$ , podrazumevano  $30^\circ$ )
  - `falloff` – stepen slabljenja od unutrašnje do spoljašnje kupe ( $n$ , podraz. 1)
- Valjani opsezi:  $0 \leq \text{innerAngle} \leq \text{outerAngle} \leq 180$  ; `falloff`  $\geq 0$
- Koeficijent radijalnog slabljenja se računa po formuli:

$$k_r(\delta) = \begin{cases} 1, & \delta < \alpha \\ \left( \frac{\cos \delta - \cos \beta}{\cos \alpha - \cos \beta} \right)^n, & \alpha \leq \delta \leq \beta \\ 0, & \delta > \beta \end{cases}$$

Svetlost i senčenje



11.04.2024.

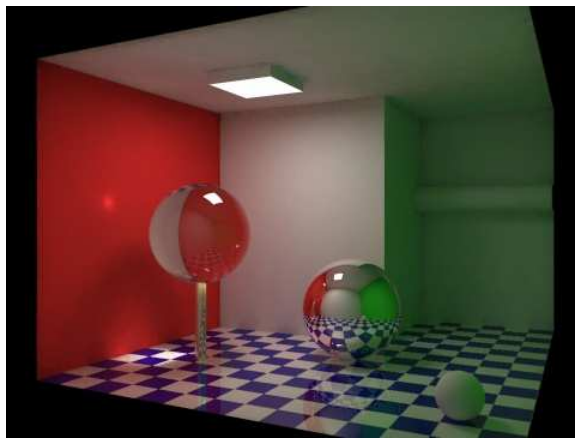
# Modeli osvetljenja

- Modeluje se interakcija svetla sa određenim tačkama osvetljene površi objekta da se odredi njihova nijansa boje
- U opštem slučaju, nijansa u nekoj tački površi objekta je određena:
  - ambijentalnim svetlom
  - difuznim svetlom (na koje utiču izvori usmerenog, tačkastog i reflektorskog svetla)
  - odsjajem izvora svetla (*specular reflection*) koji zavisi od reflektivnosti materijala
  - drugim osobinama materijala: boja, prozirnost, tekstura, reljefnost, samoosvetljenje
  - senkama drugih objekata
  - ogledanjima drugih objekata (*reflections*)
  - prelamanjem svetla (*refractions*)
- Model osvetljenja razmatra
  - attribute izvora svetala (vrstu, intenzitet, poziciju, smer svetla, radijalno slabljenje)
  - attribute sredine prostiranja (konstantno, linearno, kvadratno slabljenje)
  - attribute površi objekta (boju, reflektivnost, transparentnost)
  - interakciju između svetala i objekata (orijentaciju površi objekta)
  - interakciju između objekata i oka posmatrača (smer pogleda)



# Podela modela osvetljenja

- Modeli osvetljenja se dele na lokalne i globalne modele
  - Lokalni modeli (direktno svetlo) razmatraju samo
    - izvor svetla, poziciju posmatrača i svojstva materijala objekta
  - Globalni modeli (indirektno svetlo) uzimaju u obzir
    - interakciju svetla sa svim površima u sceni (odbijanje, prelamanje,...)
    - primeri globalnog osvetljenja:



Preuzeto sa Wikipedia

# Crtanje (*rendering*)

- Crtanje u rasterskoj grafici može da bude zasnovano na tehnikama:
  - sken-konverzija (*scan-conversion*), adekvatno je lokalno osvetljenje
  - praćenje zraka (*ray-tracing*), adekvatno je globalno osvetljenje
- Sken-konverzija – za svaki piksel duž linije skeniranja poligona:
  - određuje se nijansa datog piksela
- Praćenje zraka – za svaki piksel slike:
  - kreira se zrak od oka posmatrača, kroz piksel
  - određuju se preseki zraka sa objektima u sceni
  - čuva se presek koji je bliži od prethodno utvrđenih
  - određuje se osvetljaj direktnom svetlošću izvora, ali i od odbijanja i prelamanja
  - boji se piksel nijansom određenom superpozicijom boja direktne svetlosti i svetlosti odbijene ili prelomljene od površi u sceni
- *JavaFX* podržava samo tehniku sken-konverzije, pa tako i samo modele lokalnog (direktnog) osvetljenja

# Modeli lokalnog osvetljenja

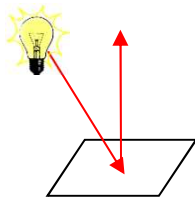
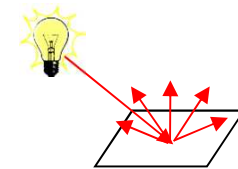
- Modeli lokalnog osvetljenja:
  - Ambijentalno osvetljenje
    - ne uzima se u obzir normala na površ ni smer svetla
  - Difuzno osvetljenje (Lamberov model)
    - uzima se u obzir vektor normale na površ i vektor smera svetla
  - Reflektivno (spekularno) osvetljenje (*Fongov* model)
    - uzima se u obzir vektor normale, vektor smera svetla i vektor pogleda
- Modeli se kombinuju, efekti se superponiraju
- *JavaFX* podržava kombinaciju sva tri modela
- *OpenGL* takođe podržava kombinaciju sva tri modela

# Model ambijentalnog osvetljenja

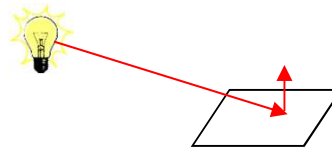
- Ambijentno ili pozadinsko (*background*) svetlo:
  - svetlo koje rasejava okruženje, reflektovano svetlo od objekata u sceni
- Predstavlja veoma uprošćeno globalno osvetljenje (aproksimacija GO)
  - model GO *Radiosity* za precizno izračunavanje ambijentalnog svetla
- Nezavisno je od:
  - položaja izvora svetla, orijentacije objekta, položaja i orijentacije posmatrača
- Nema usmerenje
- Uzima se da je intenzitet upadnog ambijentalnog svetla na svakoj površi  $I_a$
- Različiti materijali mogu da reflektuju različite iznose ambijentalnog svetla
  - koeficijent  $k_a$  ( $0 \leq k_a \leq 1$ ) određuje odnos reflektovanog ambijentalnog svetla i upadnog
- Ambijentalna komponenta reflektovanog svetla na objektu iznosi:
  - $Ambient = k_a * I_a$

# Model difuznog osvetljenja (1)

- Difuzno svetlo:
  - osvetljaj koji površ prima od svetlosnog izvora i reflektuje jednako u svim pravcima
- Nebitno gde se nalazi posmatrač, bitan smer svetla i orijentacija površi
- Potrebno je odrediti iznos energije (svetla) koje objekat prima iz izvora svetla
- Zasniva se na Lamberovom zakonu:



prima više svetla

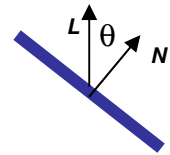


prima manje svetla

# Model difuznog osvetljenja (2)

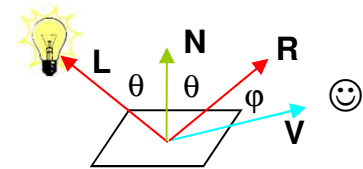
- Lamberov zakon:
  - energija zračenja  $E_i$  koju prima mali deo površine iz izvora svetla je:  
 $E_i = I_d \cos(\theta)$
  - $I_d$  intenzitet difuzne komponente upadnog svetla,
  - $\theta$  ugao između vektora smera svetla ( $\mathbf{L}$ ) i normale na površ ( $\mathbf{N}$ )
- Različiti materijali mogu da reflektuju različite iznose difuznog svetla
  - koeficijent refleksije difuznog svetla:  
 $k_d$ , ( $0 \leq k_d \leq 1$ )
- Difuzna komponenta reflektovanog svetla na objektu iznosi:

$$\text{Diffuse} = k_d I_d \cos(\theta) = k_d I_d \mathbf{L} \cdot \mathbf{N}$$



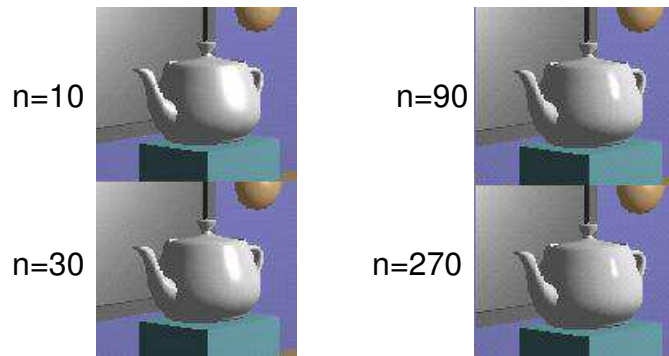
# Fongov model osvetljenja (1)

- Empirijski model lokalnog osvetljenja tačaka na površi objekta
- *Phong reflection model*, *Phong illumination*, *Phong lighting*
- Ne treba da se meša sa algoritmom Fongovog senčenja (*Phong shading*)
  - algoritam interpolacije
- Predložen 1973.u PhD disertaciji na *University of Utah*
- Fong je zapazio da sjajne površi imaju male istaknute odsjaje, dok mat površi imaju veće površi odsjaja sa blažim prelazima
- Ako bi površ bila idealno reflektivna, odsjaj izvora bi se video samo na pravcu vektora ( $\mathbf{R}$ ) koji zaklapa isti ugao  $\theta$  sa normalom ( $\mathbf{N}$ ), kao i vektor svetla ( $\mathbf{L}$ ), odnosno za  $\varphi=0$ , ili  $\mathbf{V}=\mathbf{R}$
- Za neidealnu površ intenzitet odsjaja opada sa povećanjem  $\varphi$



## Fongov model osvetljenja (2)

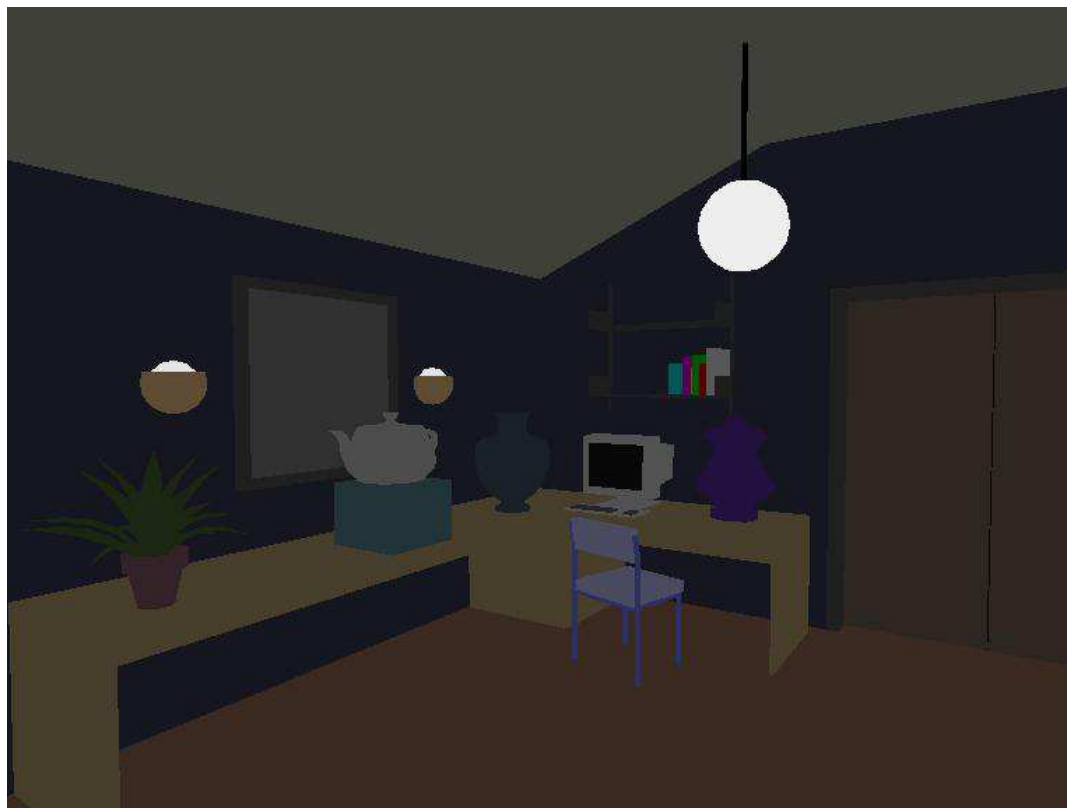
- Fongov model reflektivnog (spekularnog) osvetljenja:  
$$Specular = k_s * I_s * \cos^n(\varphi) = k_s * I_s * (\mathbf{R} \cdot \mathbf{V})^n$$
  - $k_s$  koeficijent spekularne refleksije
  - $I_s$  intenzitet spekularne komponente upadnog svetla
  - $\varphi$  ugao između vektora pogleda ( $\mathbf{V}$ ) i vektora idealne refleksije ( $\mathbf{R}$ )
  - $n$  koeficijent odsjaja (refleksivnosti materijala)
- Što je veće  $n$  (visoka refleksivnost) to je manja vrednost  $\cos^n(\varphi)$ ,
  - odsjaj je bliži idealnom, rasipanje svetla je manje



Preuzeto iz A.Watt  
3D Computer Graphics



# Ambijentalno osvetljenje - primer



Preuzeto iz A.Watt  
3D Computer Graphics

## + difuzno osvetljenje - primer



Preuzeto iz A.Watt  
3D Computer Graphics

# + spekularno osvetljenje - primer



Preuzeto iz A.Watt  
3D Computer Graphics

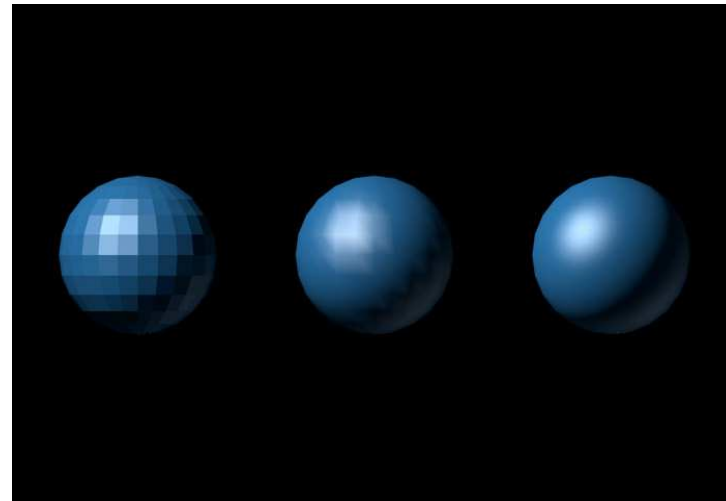
# Teksture i globalno osvetljenje



Preuzeto iz A.Watt  
3D Computer Graphics

# Modeli senčenja

- Prilikom rasterizacije poligona vrši se određivanje nijanse piksela na (horizontalnoj) liniji skeniranja (sken-konverzija)
- U određenim tačkama poligona koji se iscrtava primenjuje se odgovarajući model osvetljenja
- Tri algoritma senčenja:
  - ravnomerno,
  - Guroovo (Gauraud)
  - Fongovo (Phong)



# Ravnomerno (*flat*) senčenje

- Svetlo se izračunava samo u jednoj tački poligona
  - uzima se u obzir normala i osobina materijala u prvom temenu
- Ceo planarni poligon (svi njegovi pikseli) se boji istom nijansom
- Dobra strana:
  - brzina, izračunava se samo jedna nijansa za ceo poligon
- Loše:
  - dobijaju se oštre ivice poligona, realističnost je slaba
  - izražen “Mahov efekat traka” (sledeći slajd)
- Kada može da se koristi:
  - kada su poligoni dovoljno mali (fina mreža)
  - kada je izvor svetla dovoljno udaljen
  - kada je posmatrač dovoljno udaljen

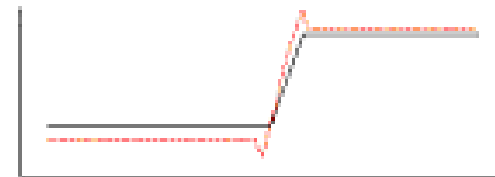
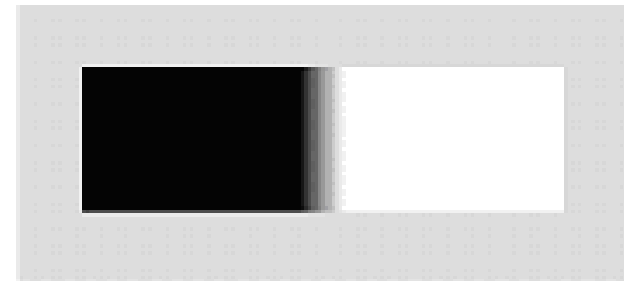


# Mahov efekat traka

- Ljudsko čulo vida pojačava efekat naglih prelaza u nijansi



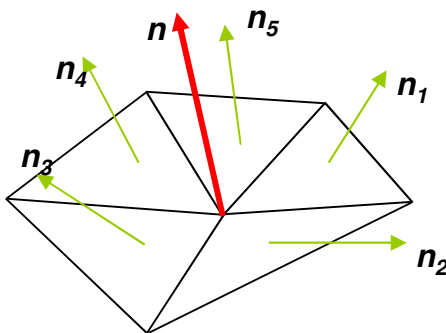
- Smanjivanje efekta traka
  - ukloniti diskontinuitete vrednosti nijanse
  - izračunati svetlo u više tačaka površi, pa raditi interpolaciju
  - dobijaju se blagi prelazi, umanjuje se Mahov efekat traka
  - crvena linija na grafikonu – percepcija nijanse



- Dva algoritma (modela) senčenja koja rešavaju gornji problem
  - Guroovo senčenje
  - Fongovo senčenje

# Guroovo senčenje

- Nijansa se izračunava u temenima poligona, pa se u pikselima poligona uzima interpolirana vrednost boje
- Interpoliranje se radi prvo duž ivica poligona, a zatim duž svake linije skeniranja pri popunjavanju poligona
- Dobija se kontinualna promena nijanse unutar poligona
- Problem – ogledanje izvora je tačno samo u temenima
- Mogućnost da se u svakom temenu poligona normala izračuna usrednjavanjem normala susednih poligona



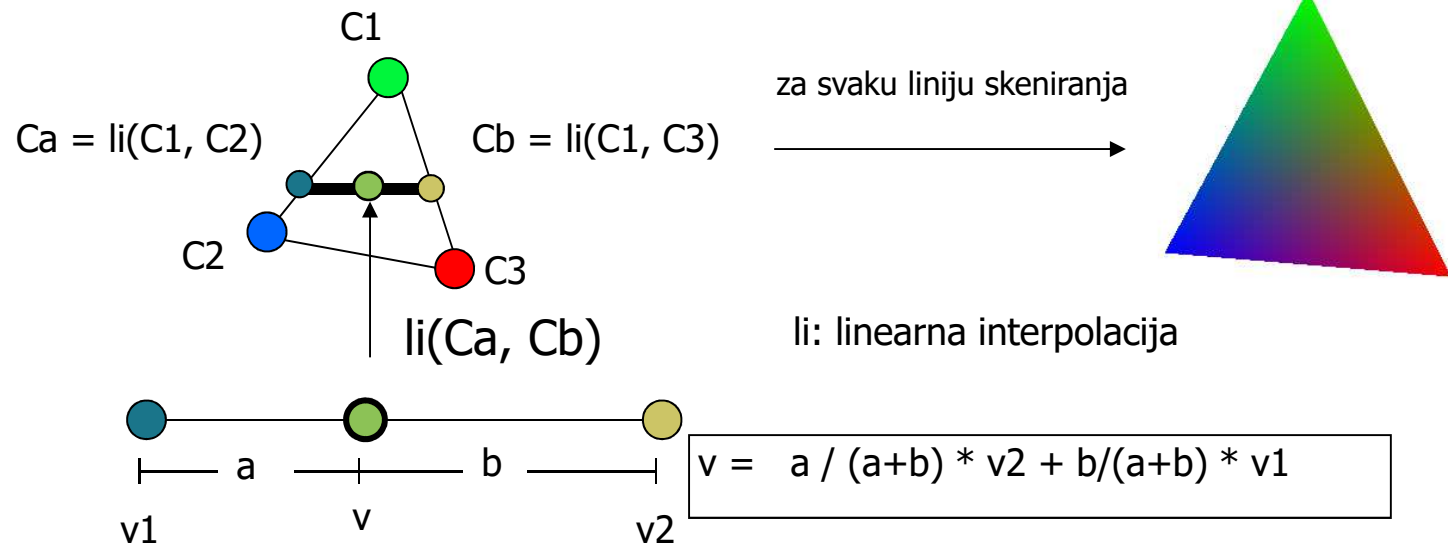
$$n = (n_1 + n_2 + n_3 + n_4 + n_5) / 5$$

Svetlost i senčenje

11.04.2024.



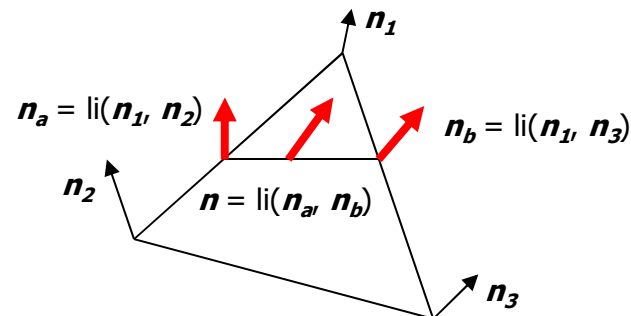
# Interpolacija



- Za izračunavanje krajnjih tačaka linije skeniranja može da se koristi interpolacija po y

# Fongovo senčenje

- Izračunava se osvetljaj svakog piksela unutar poligona
  - Guroovo senčenje – boja po pikselu
  - Fongovo senčenje – svetlo po pikselu
- Umesto interpolacije boje vrši se interpolacija normale
- Potrebne su normale u svakom pikselu
  - ne obezbeđuje ih korisnik, automatski se računaju
  - normale u temenima se određuju usrednjavanjem normala
  - algoritam interpolira normale i računa svetlo u toku rasterizacije poligona
  - račun se sprovodi u koordinatnom sistemu realnog sveta



Svetlost i senčenje

# Modeli globalnog osvetljenja

- Ray casting
- Recursive ray tracing
- Radiosity
- Radiance
- Photon mapping
- ...