



KATEDRA  
INFORMATIKY  
UNIVERZITA PALACKÉHO V OLMOUCI

# Světlo a stíny

## KMI/3DG

Mgr. Markéta Trnečková, Ph.D.

# Světlo

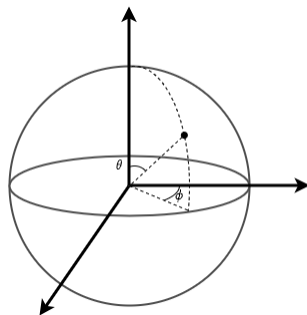
## Základní pojmy

- **Viditelné spektrum** – úzké pásmo EM spektra, každá frekvence uvnitř tohoto intervalu odpovídá určité barvě
- Dopadne-li světlo na objekt, některé frekvence jsou pohlceny a jiné odraženy, nebo mohou projít skrz
- **Světelný paprsek** – polopřímka (má počátek a směr)
- Zatím se omezíme jen na optické vlastnosti – barevnou reprezentaci

# Světlo

## Prostorové úhly

- Výpočet osvětlení bodu – světlo procházející skrz polokouli, která tento bod obklopuje
- K popisu používáme vektory na povrchu koule
- Jednotkový vektor  $\vec{\omega}$  může být popsán jako bod na povrchu jednotkové koule
- Popsat ho můžeme pomocí dvou úhlů  $(\theta, \phi)$
- $\theta$  – určuje výšku – **elevator**
- $\phi$  – azimut – **azimuth**
- **Sférické souřadnice** –  $[r, \theta, \phi]$  jednoznačná poloha bodu na kouli o poloměru  $r$
- Prostorové úhly se orientují tak, aby směřovaly směrem od povrchu



# Světlo

## Prostorové úhly

### ■ Převod sférických souřadnic do kartézských:

$$x = r \cdot \cos \phi \cdot \sin \theta$$

$$y = r \cdot \sin \phi \cdot \sin \theta$$

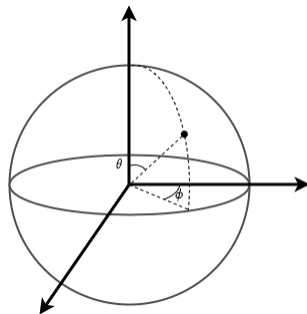
$$z = r \cdot \cos \theta$$

### ■ Převod kartézských souřadnic do sférických:

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\tan \phi = \frac{x}{y}$$

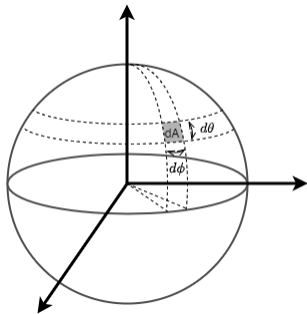
$$\tan \theta = \frac{\sqrt{x^2 + y^2}}{z}$$



# Světlo

## Diferenciální plocha

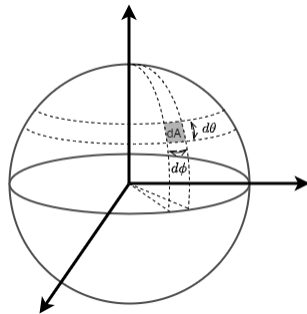
- Směr můžeme popsat i pomocí **diferenciální plochy** ( $dA$ ) na povrchu koule o poloměru  $r$
- Diferenciální plocha je vymezena 2 úhly  $d\theta$  a  $d\phi$  a poloměrem  $r$   
$$dA = r^2 \sin\theta d\theta d\phi$$
- Velikost plošky se zmenšuje úměrně tomu, jak se přibližuje k pólu koule, proto se ve vzorci objevuje korekční člen  $\sin\theta$
- Úhel na kružnici je definován v radiánech, prostorový úhel ve steradiánech
- Obvod kružnice o poloměru  $r - o = 2\pi r$
- Plný úhel je tedy  $2\pi$
- Celkový povrch koule –  $4\pi r^2$
- Plný prostorový úhel je  $4\pi$



# Světlo

## Diferenciální plocha

- **Diferenciální prostorový úhel**  $d\vec{\omega}$  je vymezen ploškou  $dA$   
$$d\vec{\omega} = \sin \theta d\theta d\phi$$
- Jeden prostorový úhel vyjadřuje na povrchu koule různě velké oblasti
- Volně se zaměňují pojmy diferenciální prostorový úhel, diferenciální ploška na jednotkové kouli a vektor – všechny popisují směr
- Směr  $d\vec{\omega}$  je určen bodem na jednotkové kouli a velikost odpovídá velikosti prostorového úhlu v daném směru
- Vektor  $\vec{\omega} = (\theta, \phi)$  určuje jednoznačně nějaký směr, stejně jako vektor  $(v_x, v_y, v_z)$



# Světlo

## Radiometrické pojmy

- **Foton** – základní světelná částice
- Nejmenší kvantum záření, které může být vyzářeno
- Energie fotonu –  $e_\lambda = \frac{h \cdot c}{\lambda}$   
 $h \approx 6.63 \cdot 10^{-34}$  Js ... Planckova konstanta  
 $c = 299792$  m/s ... rychlost světla ve vakuu  
 $\lambda$  ... vlnová délka fotonu (m)
- **Radiantní energie**  $Q$  (J) je energie fotonů určité vlnové délky  $\lambda$  s energií  $e_\lambda$   
 $Q = \int_0^\infty n_\lambda e_\lambda d\lambda$   
 $n_\lambda$  ... počet fotonů vlnové délky  $\lambda$
- Dále uvedené veličiny nebudeme vztahovat ke konkrétní vlnové délce. Tyto veličiny získáme jako integrál na vlnové délce závislých příspěvků přes celé viditelné spektrum.

# Světlo

## Radiometrické pojmy

- Světlo dopadající na povrch objektu či vyzářené do prostoru v určitém čase – **zářivý výkon** (radiant power) nebo také **zářivý tok** (flux)
- Označení  $\Phi$ , jednotka watt W
- Množství energie vyzářené (přijaté) za jednotku času
$$\Phi = \frac{dQ}{dt}$$
- V případě zářivého výkonu se nerozlišuje, zda se jedná o výkon vyzářený nebo dopadající, my budeme rozlišovat:
  - $\Phi_i$  – dopadající (incident)
  - $\Phi_r$  – odražený (reflected)



# Světlo

## Radiometrické pojmy

- Světelný tok dopadající na jednotku plochy, též hustota světelného výkonu na ploše – **iradiance**  $E$  ( $W/m^2$ )

$$E = \frac{d\Phi_i}{dA}$$

$A$  ke ozářené plocha v  $m^2$

- Světelný tok vyzářený jednotkou plochy (může být vlastní záření, nebo odražené) – **radiozita**  $B$  nebo  $M$  ( $W/m^2$ )

$$M = B = \frac{d\Phi_i}{dA}$$

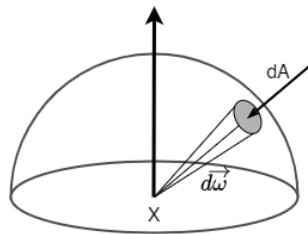
$A$  ke ozářené plocha v  $m^2$

# Světlo

## Radiometrické pojmy

- **Zářivá intenzita** (zářivost)  $I$  (W/sr) – hustota výkonu b prostorovém úhlu – světelný tok  $\Phi$  proudící prostorovým úhlem  $d\vec{\omega}$

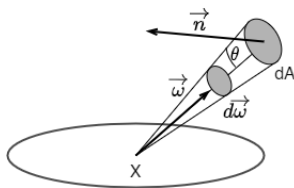
$$I = \frac{d\Phi}{d\vec{\omega}}$$



# Světlo

## Radiometrické pojmy

- **Radiance** – přijímaný (či vyzářený) výkon na jednotkovém prostorovém úhlu na jednotku kolmo promítnuté plochy
- Veličina udává počet fotonů přicházejících (nebo vyzářených) v určitém směru za jednotku času a procházející průmětem diferenciální plošky  $dA$ , který je kolmý na tento směr
- Radiance je to, co obvykle označujeme jako barvu
- Závidí na poloze bodu  $X$  (bod ve kterém radianci počítáme) a směru  $\vec{\omega}$
- $L(x, \vec{\omega})$  ( $W/sr m^2$ )  
$$L(x, \vec{\omega}) = \frac{d^2\Phi}{\cos\theta dA d\vec{\omega}}$$



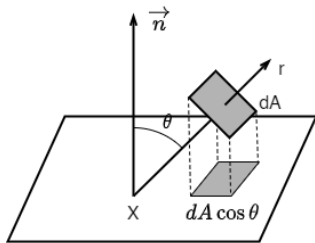
# Světlo

## Radiometrické pojmy

- $L(x, \vec{\omega})$  ( $W/srm^2$ )

$$L(x, \vec{\omega}) = \frac{d^2\Phi}{\cos\theta dA d\vec{\omega}}$$

$\cos\theta$  ve jmenovateli je kvůli tomu, že velikost plochy závidí na kosinu úhlu, který svírá kolmice k promítané ploše a kolmice k ploše promítnuté.



# Světlo

## Radiometrické pojmy

- Všechny dříve zavedené pojmy se dají zavést pomocí radiance
- **Radiantní energie** – integrál přes čas ( $T$ ), polokouli ( $\Omega$ ) a body plochy  
$$Q = \int_T \int_{\Omega} \int_{x \in A} L(x, d\vec{\omega}) \cos \theta dA d\vec{\omega} dt$$
- **Zářivý tok** – integrál body plochy a polokouli  
$$\Phi = \int_{\Omega} \int_{x \in A} L(x, d\vec{\omega}) \cos \theta dA d\vec{\omega}$$
- **Irradiance** – analogicky i **radiozita**  
$$E = \int_{\Omega} L(x, d\vec{\omega}) \cos \theta d\vec{\omega}$$
- **Intenzita** – analogicky i **radiozita**  
$$I = \int_{x \in A} L(x, d\vec{\omega}) \cos \theta dA$$

# Světlo

## Radiometrické pojmy

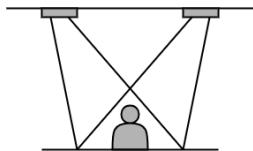
- Ve vakuu je radiance konstantní na celé své dráze
- Lidské oko je citlivé na radianci. Odezva je úměrná přijaté radianci, barva či kontrast se nemění se vzdáleností, ze které se těleso pozoruje
- Obrazovka počítače radianci interpretuje jako barvu

# Světlo

- **Světlo** – forma energie, skládá se z paprsků různých vlnových délek
- Různé objekty reagují se světlem různě – některé ho více odrážejí (například kovy), jiné ho pohlcují, nebo světlo prochází skrz ně (např. sklo)
- Kromě materiálu také tvar a povrch (zda je hladký, či členitý) ovlivňují to, jak se světlo od objektu odráží
- Od objektu se mohou odrážet jen některé vlnové délky – barva (všechny vlnové délky tvoří bílé světlo)
- Barvám a základům, jak lidské oko vnímá barvy jsme se věnovali v POGR.

# Hloubka

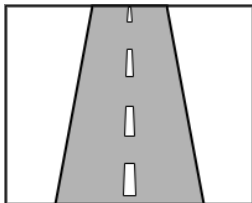
- **Hloubka** – vzdálenost
- Člověk pozoruje svět 2 očima, každé oko se na objekt dívá z jiné pozice
- Díky rozdílu těchto dvou obrazů dokážeme určit relativní vzdálenost objektů – **stereoskopické vidění**
- Zavřete oko a srovnejte pohled tak, abyste se dívali na hranu stolu, ale neviděli jí z boku. Otevřete druhé oko a to první zavřete. Hrana najednou vypadá úplně jinak.
- Rozdíl mezi tím, co je vidět jednotlivými očima závisí na vzdálenosti objektu. Čím bude dál, tím bude větší rozdíl
- Zakryjte papírem část prostoru a podívejte se pravým okem, co vše je skryto za papírem. Pak pravé oko zavřete a podívejte se levým.





# Hloubka

- Hloubku také určíme také tím, že objekty, co jsou blíže k nám, ovlivňují ty, které jsou za nimi
- Neprůhledné objekty skrývají objekty za nimi a poloprůhledné mění jejich vzhled
- Další jev, který napomáhá s vnímáním hloubky je to, že se objekty, co se od nás vzdalují, zmenšují



# Osvětlení

## ■ Rozdělení:

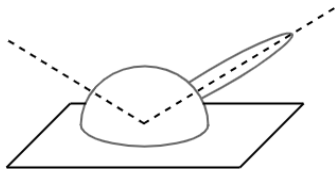
- **Lokální** – jak je každý objekt jednotlivě osvětlen
- **Globální** – osvětlení celkově, nejen světlem ze zdroje, ale i odraženým od jiných objektů

## ■ V lokálním osvětlení předpokládáme:

- Světlo, které se odrazí od povrchu, se odrazí okamžitě – díky tomu nemůžeme simulovat **fosforescenci**
- Foton o vlnové délce  $\lambda$  bude vyzářen se stejnou vlnovou délkou – nemůžeme vyjádřit **fluorescenci**
- Světlo přicházející k povrchu z nějakého vstupního směru  $\vec{\omega}_i$  se střetne s konkrétním bodem  $x$  a odtud se odrazí (neodrazí se z jiného bodu – necestuje pod povrchem)

# BRDF

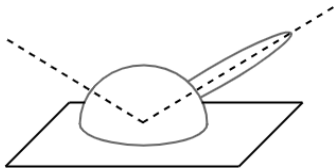
- Popis toho, jak se světlo odráží od objektů – **Bidirectional reflectance distribution function** (BRDF)
- Závisí na relativní pozici a orientaci zdroje světla, objektu a také pozorovatele
- Bidirectional – pokud prohodíme pozici zdroje světla ( $\omega_i$ ) a pozorovatele ( $\omega_r$ ), výsledek se nezmění = **Helmholtzův princip reciprocity**
- Odražené světlo se distribuuje do různých směrů



- Čím bude objekt lesklejší, tím více se paprsek odrazí ve směru podél čárkované čáry a do okolí se odrazí málo až téměř nic

# BRDF

- Čím bude objekt lesklejší, tím více se paprsek odrazí ve směru podél čárkované čáry a do okolí se odrazí málo až téměř nic
- Pokud by byl povrch drsný, polokoule bude velká a ve směru čárkované čáry se odrazí přibližně stejně světla



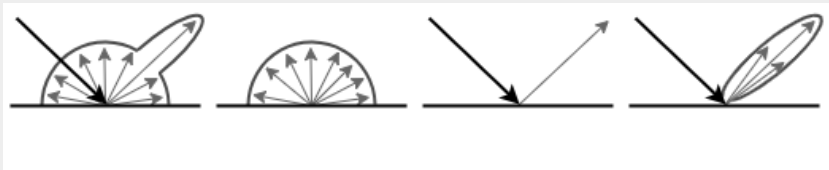
# BRDF

- BRDF můžeme popsat pomocí 3 složek:

- **Zrcadlová složka** – dopadající paprsek je odražen jen v jednom směru
- **Difúzní složka** – perfektní = odráží paprsek všemi směry stejně
- **Speculární složka** (glossy) – paprsek se odráží do více směrů, ale kopíruje směr paprsku, který se odrazí od povrchu pod stejným úhlem, jako na něj dopadl

## Příklad

Přiřaďte obrázky k jednotlivým složkám BRDF.



# BRDF

- Některé objekty mají stejnou BRDF bez ohledu na to, odkud na ně světlo dopadá – **izotropie**
- **Anizotropie** – BRDF se liší dle směru, v jakém na objekt dopadá (např. saténová látka)
- Tvar také může záviset na vlnové délce světla – různé pro různé vlnové délky
- BRDF je vždy pozitivní – zákon o zachování energie (energie se nemůže vytvořit ani zaniknout, jen přeměnit)
- Měříme ji pomocí **goniometru** – zdroj světla a detektor, které jsou namířeny na stejné místo (kde je materiál)
- Oba se mohou kolem materiálu pohybovat
- Aby bylo měření přesné – nesmí být ovlivněno okolním světlem
- Existuje několik modelů, které aproximují BRDF – **osvětlovací model** (shading model)

# Jednoduchý osvětlovací model

## ■ Lokální osvětlení 3 složky:

- **Specular (zrcadlová)** – produkuje světlý bod (část) na objektu, je přímo založená na přímém odrazení světla a bude vidět, pokud paprsek odraženého světla půjde směrem k pozorovateli (nemusí úplně přesně). Projevuje se u lesklých hladkých objektů (u drsných se světlo šíří všemi směry rovnoměrně)
- **Diffuse (šíření)** – obecné zvýraznění objektu, závisí pouze na směru světla a normálovém vektoru na povrchu objektu. V místech, kde je normála a směr světla stejný (podobný) – větší difúzní složka. Čím je úhel mezi nimi větší, tím více se redukuje. Vůbec nezáleží na poloze pozorovatele.
- **Ambient (okolí)**



Bez osvětlení, ambientní složka, difúzní složka, spekulární složka, vše

## Jednoduchý osvětlovací model

- Spekulární a difúzní složka jsou závislé na přímém osvětlení ze zdroje světla
- V místnosti s jedním zdrojem světla difúzní a spekulární složky osvětlí jen ty objekty, které nejsou zastíněny jinými
- Objekty, které nejsou přímo ve světle jsou viditelné, což znamená, že se k nim světlo nějakým způsobem dostane
- Později se budeme zabývat globálním osvětlením, které se tímto problémem zabývá.
- V lokálním osvětlení toto modelujeme pomocí ambientní složky, což je konstanta, kterou přidáme každému objektu (aproximuje nepřímé osvětlení)



## Jednoduchý osvětlovací model

- Koule jsou renderovány s každou složkou zvlášť a pak dohromady
- 2. koule nevypadá prostorově, všechny body mají stejnou hodnotu
- 3. koule vypadá prostorově (má nějaký tvar), vypadá jako by byla vytvořená z matného materiálu (nemá žádný odlesk)
- 4. koule má odlesk, uprostřed je hodnota nejvyšší, čím jdeme dále od středu, je odlesk menší

- **Jednoduchý osvětlovací model:**

$$I = k_a I_a + k_d I_d + k_s I_s$$

$k_*$  jsou konstanty, které se volí různě pro různé materiály

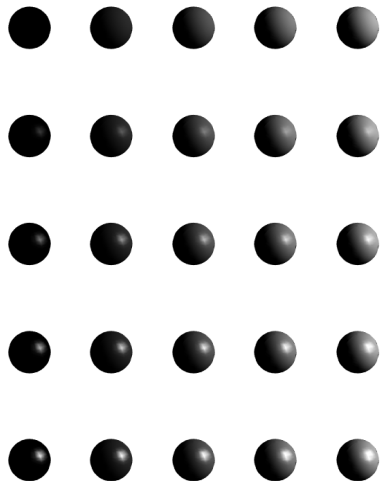
- Difúzní složka – materiál s hrubým povrchem bude mít větší konstantu  $k_d$  než naleštěné
- Maximálně difúzní materiál – křída



Bez osvětlení, ambientní složka, difúzní složka, spekulární složka, vše

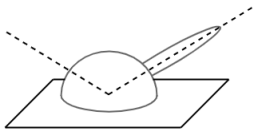
# Jednoduchý osvětlovací model

- → difúzní složka  
0, 0.25, 0.5, 0.75, 1
- ↓ spekulární složka  
0, 0.25, 0.5, 0.75, 1



# Jednoduchý osvětlovací model

- Difúzní složka – se zvětšující se konstantou se zvětšuje polokoule
- To, že v centru je větší intenzita není dáno touto konstantou
- Světlejší je proto, že se z okolních míst odráží difúzní složky
- Spekulární složka – materiály s hrubým povrchem budou mít tuto konstantu menší než ty lesklé hladké
- Čím je větší, tím je větší výběžek
- Rovnici počítáme pro každou vlnovou délku (barevnou složku) zvlášť
- Konstanty se mohou lišit pro jednotlivé barvy



## Okolní světlo – ambientní

- **Okolní světlo** (ambient) – je modelován konstantou, kterou přidáváme k objektům jako nepřímé osvětlení, které přichází z okolí (ne přímo ze zdroje světla)
- Světlo bez jakéhokoliv směru
- Udělá objekty viditelné, ale nevytvoří na nich žádné odlesky, nebo zvýraznění

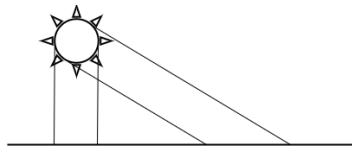
## Rozptýlené světlo – difúzní

- **Rozptýlené světlo** (diffuse)
- Pro perfektně difúzní materiál se světlo rovnoměrně odrazí do všech směrů
- $L_r(x, \vec{\omega}_r) = \int_{\Omega} f_d(x, \vec{\omega}_r, \vec{\omega}_i) L_i(x, d\vec{\omega}_i) \cos \theta d\vec{\omega}_i$   
 $f_d$  je BRDF, což je konstanta. Pro všechny úhly je odražená hodnota rovna iradianci vynásobené BRDF.
- Světlý bod uprostřed se zobrazí bez ohledu na to, kde je pozorovatel. závisí pouze na umístění objektu vůči zdroji světla



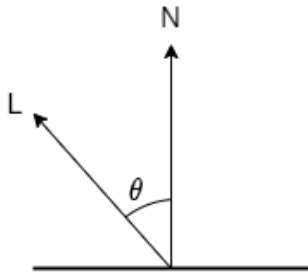
## Rozptýlené světlo – difúzní

- Difúzní odraz nemá na objekt ostré hrany
  - Hrubé materiály mají větší difúzní složku než lesklé
  - Vzhled souvisí s tvarem objektu, kterým směrem k němu ze zdroje dopadá a jak vypadá okolí
  - Difúzní odraz bude největší v místech, kde je zdroj světla přímo nad ním a bude klesat čím je dál
- 
- Představme si slunce a jak osvětluje zem
  - Pokud jde energie přímo dolů, dopadá na menší plochu, než stejné množství energie dopadající pod nějakým úhlem



## Rozptýlené světlo – difúzní

- Na obrázku je povrch
- $N$  ... normála k povrchu
- $L$  ... směr, kterým je světelný zdroj
- $\theta$  ... úhel pod kterým dopadá paprsek
- Čím je úhel větší, tím se  $\cos \theta$  zmenšuje a tedy se zmenšuje i difúzní odraz
- $I_d = I_L \cos \theta = I_L \cdot (L \cdot N)$   
 $I_L$  je intenzita zdroje  
Pokud je úhel  $N$  a  $L$  větší než  $90^\circ$ , pak je skalární součin vektorů záporný a tudíž se nic neodrazí
- Nejsvětější bod je v místě, kde je  $\cos$  největší (to je v případě, kdy jsou vektory  $L$  a  $N$  shodné – úhel je 0)



## Zrcadlové světlo – spekulární

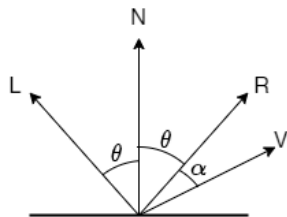
- **Zrcadlové světlo** (specular)
- Velké jasné odlesky na povrchu tělesa
- Jeho velikost závisí na materiálu
- Čím je objekt lesklejší, tím je světlá část menší, u hrubých povrchů se více rozptýlí do stran
- Záleží i na poloze pozorovatele a světla





## Zrcadlové světlo – spekulární

- **Zrcadlo** – odráží světlo pod zrcadlově souměrným úhlem ke směru dopadu
- $L$  ... směr, kterým je světelný zdroj
- Paprsek se odrazí pod úhlem  $R$   
$$R = 2 \cdot (N \cdot L) \cdot N - L$$
- $V$  ... směr odkud objekt pozorujeme
- Pokud se  $V$  a  $R$  rovnají, uvidíme nejjasnější odraz
- Čím jsou od sebe  $R$  a  $V$  vzdálenější, pozorovatel vidí méně odrazu



## Zrcadlové světlo – spekulární

- Osvětlení

$$I_S = I_L \cdot \cos^n \alpha = I_L \cdot (R \cdot V)^n$$

$n$  určuje jak moc je široký odraz (simuluje hrubost povrchu)

větší  $n$  = hladší povrch a užší výstupek

- Na obrázku jsou různě velká  $n$  (5, 10, 15, 20, 25, 30)



## Atmosférický útlum

- Atmosféra obsahuje prachové částice, smog, vodní a jiné páry
- Množství těchto částic určuje, jak daleko vidíme
- Také ovlivňuje ostrost vzdálených objektů
- Při generování obrazu můžeme atmosférické podmínky simulovat
- Barva oparu –  $(H_R, H_G, H_B)$
- Hloubka, kde opar začíná –  $z_{start}$
- Hloubka, odkud opar zakrývá celý objekt –  $z_{end}$
- Mezi  $z_{start}$  a  $z_{end}$  interpolujeme barvu objektu  $(O_R, O_G, O_B)$  a barvu oparu
- Atmosféra ovlivňuje i vlastnosti světelného zdroje – částičky v atmosféře způsobují rozptýlení světla
- Čím je více částiček v atmosféře (větší opar), tím dříve se světlo utlumí
- Toto modelujeme snížením intenzity zdroje se vzdáleností objektu od tohoto zdroje

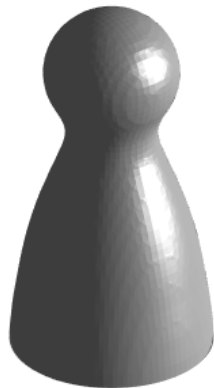
## Osvětlovací modely – shading

- **Stínování** (shading nikoliv shadow)
- Ukážeme si, jak dříve popsané zkombinovat a díky tomu spočítat hodnoty pro vykreslení objektu
- Každý objekt se skládá z nekonečného počtu bodů, pro které bychom museli počítat osvětlení
- Bylo vyvinuto několik metod, jak nepočítat hodnoty pro všechny body, ale jen pro některé a zbytek z nich odvodit
- Dále předpokládáme, že je ve scéně jen jeden zdroj světla
- Pokud by bylo ve scéně zdrojů více, pak bychom výslednou hodnotu získali, jako součet ambient složky, všech difúzních složek (pro každý světelný zdroj) a všech spekulárních složek
- Samozřejmě oříznuté tak, aby byly hodnoty v rozsahu 0 a 1

# Osvětlovací modely – shading

## Flat shading

- **Flat shading** (konstantní stínování)
- Osvětlení se počítá pro každou plochu povrchu a celá plocha je vykreslena stejnou barvou
- Jelikož mají celé plochy stejnou barvu, jsou na povrchu objektu viditelné přechody mezi nimi



# Osvětlovací modely – shading

## Flat shading

### ■ Osvětlovací model je definován 8 parametry:

- Ambientní komponenty –  $k_{ar}$ ,  $k_{ag}$ ,  $k_{ab}$
- Difúzní komponenty –  $k_{dr}$ ,  $k_{dg}$ ,  $k_{db}$
- Spekulární komponenta –  $k_s$
- Spekulární exponent –  $n$

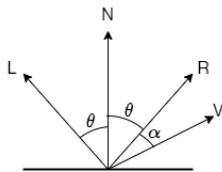
### ■ $I_L$ intenzita zdroje světla, $L$ směr zdroje světla, $N$ normála plochy, $R$ směr odrazu od plochy, $V$ směr pozorovatele

### ■ Výsledná barva:

$$C_r = k_{ar} + I_{LR} * [k_{dr} * L \cdot N + k_s * (R \cdot V)^n]$$

$$C_g = k_{ag} + I_{LG} * [k_{dg} * L \cdot N + k_s * (R \cdot V)^n]$$

$$C_b = k_{ab} + I_{LB} * [k_{db} * L \cdot N + k_s * (R \cdot V)^n]$$



# Osvětlovací modely – shading

## Flat shading

- Pro trojúhelníkovou plochu (ta vždy leží v rovině) získáme normálu vektorovým součinem vektorů, které získáme ze dvou stran trojúhelníku (vektory začínají ve stejném vrcholu).

### Příklad

Pro trojúhelník  $ABC$  spočítejte normálu.  $A = [0, 0, 0]$ ,  $B = [10, 0, 5]$ ,  $C = [3, 2, 5]$ .

- Pokud máme normály jednotlivých vrcholů, můžeme normálu plošky spočítat, jako průměr normál vrcholů.

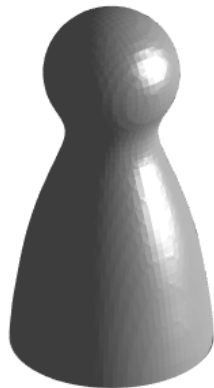
### Příklad

Pro trojúhelník  $ABC$ , kde známe normály vrcholů, spočítejte normálu plochy.  
 $\vec{n}_A = (1, 0, 0)$ ,  $\vec{n}_B = (0.5, 1, 0)$ ,  $\vec{n}_C = (0, 0.5, 0)$ .

# Osvětlovací modely – shading

## Flat shading

- Vykreslování je velmi rychlé (barva se počítá pro každou plošku jednou)
- Vzhled není příliš realistický
- Používá se v náhledech





# Osvětlovací modely – shading

## Gouraud shading

- **Gouraud shading**
- Vylepšení flat shading
- Interpolační metoda, která hladce stínuje mnohoúhelníkové plochy
- Eliminuje patrné zlomy mezi jednotlivými ploškami
- Počítá hodnotu osvětlení (barvy) všech vrcholů stejným způsobem jako flat shading

$$c_r = k_{ar} + I_{LR} * [k_{dr} * L \cdot N + k_s * (R \cdot V)^n]$$

$$c_g = k_{ag} + I_{LG} * [k_{dg} * L \cdot N + k_s * (R \cdot V)^n]$$

$$c_b = k_{ab} + I_{LB} * [k_{db} * L \cdot N + k_s * (R \cdot V)^n]$$



# Osvětlovací modely – shading

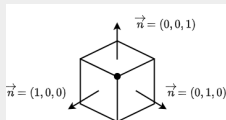
## Gouraud shading

- Jakmile máme osvětlení každého vrcholu, podél hran spočítáme osvětlení bodů pomocí lineární interpolace hodnot vrcholů, lineární interpolace mezi hranami pak určuje osvětlení bodů v ploše
- Pokud nemáme informaci o normálovém vektoru vrcholu, můžeme jí spočítat (odhadnout) zprůměrováním normálových vektorů přilehlých ploch



## Příklad

Určete normálu vrcholu.



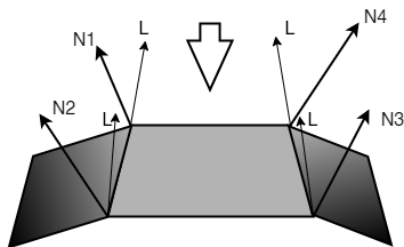
# Osvětlovací modely – shading

## Gouraud shading

- Hrany mezi plochami nejsou viditelné, protože body na hraně mají stejnou hodnotu pro obě plochy (sdílející tuto hranu)
- Obě plochy pak díky interpolaci mění svojí barvu (jak se posouváme od hrany dál) postupně
- Povrch vypadá hladce a nejsou v něm patrné přechody
- Ale! je zde problém se zrcadlovým (specular) zvýrazněním
- Pokud dopadne na hranu, bude viditelné, ale v případě, že by mělo dopadnout do středu plochy, zmizí

# Osvětlovací modely – shading

## Gouraud shading



- Plocha je tvořena 4 vrchol, u všech vrcholů je zaznačen normálový vektor
- Všechny ukazují směrem od středu plochu, uprostřed by měla být plocha zaoblená
- Mají-li všechny stejný úhel s  $L$  (směr, kterým leží zdroj světla), pak pokud se díváme přímo nad středem, mají i stejný úhel se směrem pozorovatele
- Spočítané barvy pro vrcholy budou stejné, tedy i všechny body plochy budou mít stejnou barvu (získáme je lineární interpolací stejných hodnot)
- Plocha vypadá rovně, i když očekáváme, že by měl být střed světlejší (kvůli zaoblení)

# Osvětlovací modely – shading

## Gouraud shading

- Rozdělíme-li plochu na 2, bude výsledek vypadat věrohodněji

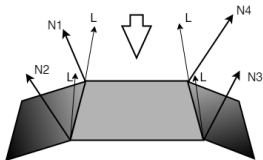


- Je obtížné odhadnout místa, kde bude mít toto osvětlení problém
- Pokud pootočíme objektem trochu do strany, zvýraznění se objeví
- Stejně tak, když plochu rozdělíme
- realističtější výsledky, pokud je povrch tvořen malými plochami

# Osvětlovací modely – shading

## Phong shading

- Phong shading
- Vylepšení předchozího osvětlení
- Předpokládejme situaci z předchozí části



- Co musíme udělat s normálovými vektory  $N1$  a  $N2$ , aby neukazovaly doleva a s  $N3$  a  $N4$ , aby neukazovaly doprava?
- Směrem od  $N1$  k  $N4$  by se měl směr normály postupně měnit od  $N1$  k  $N4$  (ve spojitě hladkém světě by to tak bylo)
- Díky tomu by odpadl problém s odleskem uprostřed



# Osvětlovací modely – shading

## Phong shading

- Phong upravil Gourard shading tak, aby rozpoznával směr normály normál podél plošek
- Místo interpolace intenzit podél plošky, interpolujeme normály
- Výpočetní náročnost je mnohem větší
- Rovnice osvětlení počítáme pro každou interpolovanou normálu

$$c_r = k_{ar} + I_{LR} * [k_{dr} * L \cdot N + k_s * (R \cdot V)^n]$$

$$c_g = k_{ag} + I_{LG} * [k_{dg} * L \cdot N + k_s * (R \cdot V)^n]$$

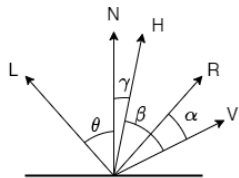
$$c_b = k_{ab} + I_{LB} * [k_{db} * L \cdot N + k_s * (R \cdot V)^n]$$



# Osvětlovací modely – shading

## Phong shading

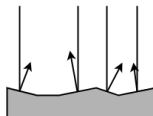
- Zrychlení tohoto procesu představil Blinn (**Blinn-Phong shading**)
- Používá aproximaci nazvanou **halfway vector** ( $H$ ), který je uprostřed mezi  $L$  a  $V$
- $H$  je jednotkový vektor
$$H = \frac{L+V}{\|L+V\|}$$
- Pokud  $R$  a  $V$  ukazují stejným směrem,  $H$  bude mít shodný směr s normálou
- Čím více se  $V$  a  $R$  liší, tím více se bude  $H$  lišit od normály
- $R \cdot V$  můžeme aproximovat pomocí  $N \cdot H$





## Realističtější stínování

- Limitace dříve představeného modelu je, že s objekty pracuje, jako by měly perfektně hladký povrch
- Některé materiály jsou přirozeně drsné
- Jedním možným řešením je, že se použije pro modelování tohoto textura, ale není to úplně ideální řešení
- Některé materiály mají nepatrné nerovnosti na povrchu – většinou jsou nepravidelné (to se pravidelnými texturami špatně modeluje)
- Navíc tyto nerovnosti nemění barvu (texturu) objektu, ale jen to, jakým stylem se od povrchu odráží světlo
- Hrubý povrch zapříčiní, že se světlo odráží v různých směrech, odchylka je ale malá
- Povrch můžeme chápat jako by se skládal z mikroplošek (**microfacets**) a ty jsou pak zodpovědné za odraz světla



## Realističtější stínování

- Při zobrazování pak hraje roli jejich velikost a jejich natočení
- Tyto mikroplošky jsou základem několika osvětlovacích modelů – Cook-Torrance, Torrance-Sparrow aj.
- Mikroplošky ovlivní zrcadlovou (specular) složku osvětlení

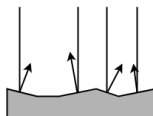
- Výpočet:

$$S = \frac{F_\theta}{\pi} \cdot \frac{D}{N \cdot L} \cdot \frac{G}{N \cdot V}$$

$F_\theta$  ... Fresnel term - úhel dopadu světla a jeho odražení

$D$  ... distribuční faktor mikroplošek (jejich orientace, reprezentuje počet mikroplošek otočených ve směru vektoru  $H$ )

$G$  ... geometrický faktor mikroplošek (jak orientace mikroplošek blokuje přístup světla nebo množství, které je blokováno po odrazu k pozorovateli)

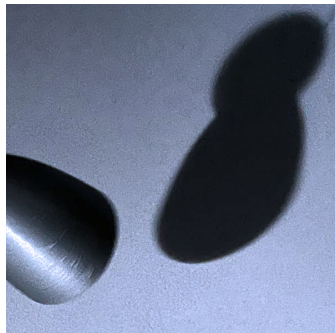


# Empirické modely

- Představené modely simulují fyzické vlastnosti objektů
- Určení parametrů (spekulární a difúzní složka) je ale těžké
- Nesprávná volba vede k tomu, že objekt nevypadá, jak očekáváme
- Existují modely, které pracují s intuitivnějšími parametry, například barva, hladkost, kovovost (metalness), průhlednost, lom světla
- Například Strauss model, Ward model

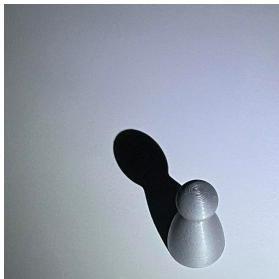
# Stíny

- Stíny hrají roli v tom, jak chápeme svět kolem nás
- Díky stínům dokáže člověk určit, kterým směrem je zdroj světla, relativní pozici objektů
- Jak se jeden objekt vzdaluje od jiného, mění se jeho stín
- Pokud objekt leží na zemi – stín je relativně ostrý a dotýká se objektu. Pokud je nad zemí, stín se nedotýká objektu a má rozmazané okraje (čím je výš, tím jsou rozmazanější).



# Stíny

- Tvar a velikost stínu závisí na vzájemné poloze světelného zdroje a stínícího objektu



# Stíny

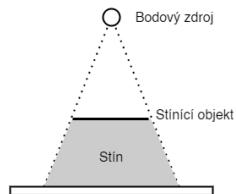
- Tvar a velikost světelného zdroje ovlivňuje charakter stínu
- Dva typy světelných zdrojů
  - Bodové
  - Plošné



# Stíny

## Bodové světelné zdroje

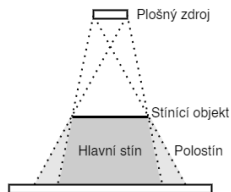
- **Bodové světelné zdroje** – pro svojí jednoduchost jsou oblíbené v počítačové grafice
- Vytváří **ostré stíny** (hard shadows) s přesně vymezenou hranicí
- Pro každý bod ve scéně můžeme jednoznačně určit, zda je z něj bodový zdroj vidět nebo zda je zastíněn jiným objektem
- V reálném světě se moc nevyskytují a stíny s ostrou hranicí nepůsobí věrohodně



# Stíny

## Plošné světelné zdroje

- **Plošné světelné zdroje** – jsou přirozenější
- Tvoří tzv. **měkké stíny** (soft shadows)
- Skládají se ze dvou částí
  - **Hlavní stín** – umbra
  - **Polostín** – penumbra
- Polostín není stejný ve všech částech, je to plynulý přechod mezi hlavním stínem a místem, kde stín není
- Čím je světelný zdroj větší, tím se hlavní stín zmenšuje a polostín zvětšuje





# Stín

## ■ Druhy stínu:

- Vlastní stín
- Vržený stín

- **Vržený stín** – stín, který vrhá jeden objekt na druhý
- Pomáhá rozpoznat umístění objektů ve scéně



# Stín

- **Vlastní stín** – stín, který vrhá objekt sám na sebe
- Ve vlastním stínu leží všechny plochy tělesa odvrácené od světla
- U těchto ploch je problém určení stínu vyřešen přímo ve fázi stínování (shading)
- Zajímavější jsou stíny, které vrhá jedna část objektu na sebe sama
- Některé algoritmy mají s těmito stíny problém



# Stín

- Generování stínu pro neprůhledné objekty je vyřešený problém
- **Problémy s poloprůhlednými objekty**
- Procházející světlo je při průchodu poloprůhledným objektem částečně pohlceno a stupeň tohoto pohlcení se liší pro různé vlnové délky
- Simulování absorpce by mělo modelovat změnu barvy procházejícího světla
- Dalším problémem je **disperze** (rozptyl) směru šíření světla, dráha světla se mění (může se měnit pro různé délky různě)
- Algoritmy, které tyto situace modelují jsou časově náročné a uvedeme si je později



# Stínování

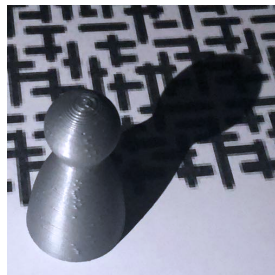
- **Hlavní myšlenka stínování** – co je vidět z místa světelného zdroje je tímto zdrojem ovlivněné, zbytek scény je vůči tomuto zdroji ve stínu → úlohu nalezení stínu můžeme převést na úlohu řešení viditelnosti
- Dále uvedeme metody jsou definované pro jeden světelný zdroj, ale je možné je rozšířit na scény s více světelnými zdroji
- **Přidání stínu**
  - Identifikace toho, kde se stín nachází
  - Úprava osvětlení v místech, kde se stín nachází (výběr barvy v místech, kde se stín nachází)

# Stínování

## Barva stínu

### Triviální řešení

- Použijeme černou nebo hodně tmavě šedou barvu všude, kde se nachází stín v době renderování objektu
- Tento proces je rychlý
- Výsledky nejsou příliš uspokojivé
- Ve skutečném světě sice oblast, kde je stín, ztmavne, ale stále prosvítá část toho objektu, kam stín dopadá
- Pokud má objekt, na který stín dopadá, texturu, textura by měla být viditelná

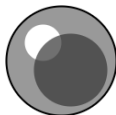


# Stínování

## Barva stínu

## Ztmavení

- Stíny nemění vzhled objektů, jen je ztmavují – tím zachováme texturu objektu ve stínu
- Pokud se ve scéně objevuje více světelných zdrojů, můžeme snižovat světlost pro každý zdroj a místa, kam dopadá více stínů, budou tmavší (v místech, kde se stíny překrývají)
- Nejtěžší je odhadnout, jak moc ztmavit
- Konstantní empiricky zvolená hodnota nezohledňuje fakt, že některé zdroje světla jsou jasnější než jiné
- Odhad na základě intenzity světelného zdroje – stín je odstranění přírůstku světla, takže pokud stíníme světlejšímu zdroji, musíme redukovat více světla
- Neintuitivní (nerealistické) na tomto přístupu je to, že pokud světlo na objektu tvoří silný odlesk a jiný objekt na toto místo vrhá stín, tak odlesk bude patrný, jen bude tmavší



# Stínování

## Barva stínu

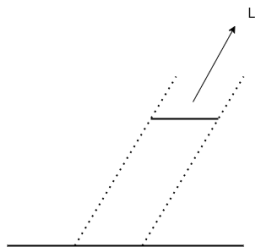
### Nepřidávání světlosti

- Nejrealističtější přístup je ten, že při výpočtu osvětlení nepřidáváme světlost ze zdroje, který je zde stíněn

# Projekční metody

## Zdroj v nekonečnu

- Pokud je zdroj světla v nekonečnu – všechny paprsky světla jsou ve scéně rovnoběžné
- $L$  reprezentuje směr, kterým je zdroj světla (opačný směr ke směru, kterým cestují světelné paprsky)
- Všechny body objektu tímto směrem vrhají stín do roviny země





# Projekční metody

## Zdroj v nekonečnu

- Rovnice přímky od světelného zdroje k bodu  $P$ :

$$S(t) = P - tL$$

3 rovnice – pro každou souřadnici jedna

- V místech, kde tato přímka protíná rovinu země ( $y = 0$ ) vzniká stín

Počítáme  $t$ , pro které platí  $y = 0$

$$t = \frac{y_P}{y_L}$$

- Pokud je  $y_L = 0$  – světlo svítí horizontálně, nemá se zemí žádný průnik, nevzniká stín

- Díky vypočítanému  $t$  můžeme dopočítat souřadnice  $x$  a  $z$ , kde bod vytváří stín

$$x_S = x_P - \frac{y_P}{y_L} x_L = x_P - \frac{x_L}{y_L} y_P$$

$$z_S = z_P - \frac{y_P}{y_L} z_L = z_P - \frac{z_L}{y_L} y_P$$

- Maticově

$$\begin{bmatrix} x_S \\ 0 \\ z_S \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -x_L/y_L & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -z_L/y_L & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix}$$

# Projekční metody

Zdroj v nekonečnu

## Příklad

Ve scéně je umístěna krychle se souřadnicemi vrcholů  $[0, 20, 0]$ ,  $[10, 20, 0]$ ,  $[10, 20, 10]$ ,  $[0, 20, 10]$ ,  $[0, 30, 0]$ ,  $[10, 30, 0]$ ,  $[10, 30, 10]$ ,  $[0, 30, 10]$ . Určete tvar stínu, který vrhá, pokud je zdroj v nekonečnu a směr světla je  $(10, 10, 10)$ .

# Projekční metody

Zdroj v nekonečnu

## Příklad

Ve scéně je umístěna krychle se souřadnicemi vrcholů  $[0, 20, 0]$ ,  $[10, 20, 0]$ ,  $[10, 20, 10]$ ,  $[0, 20, 10]$ ,  $[0, 30, 0]$ ,  $[10, 30, 0]$ ,  $[10, 30, 10]$ ,  $[0, 30, 10]$ . Určete tvar stínu, který vrhá, pokud je zdroj v nekonečnu a směr světla je  $(10, 10, 10)$ .

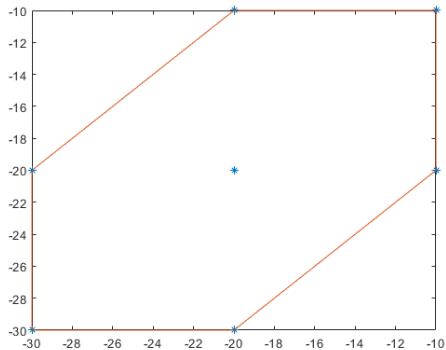
# Projekční metody

## Zdroj v nekonečnu

### Příklad

Ve scéně je umístěna krychle. Určete tvar stínu, který vrhá, pokud je zdroj v nekonečnu a směr světla je  $(10, 10, 10)$ .

$$S = \begin{bmatrix} -20 & -10 & -10 & -20 & -30 & -20 & -20 & -30 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -20 & -20 & -10 & -10 & -30 & -30 & -20 & -20 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



# Projekční metody

## Zdroj světla v bodě

- Zdroj je v bodě  $L$  místo v nekonečnu ( $L$  není v tomto případě vektor)
- Proces výpočtu je obdobný, jen se použije perspektivní promítání místo rovnoběžného
- Výpočet

$S(t) = P - t(P - L)$  Opět chceme, aby  $y = 0$

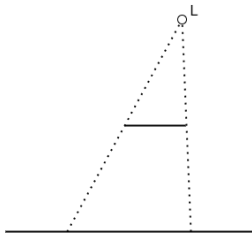
$$t = \frac{y_P}{(y_P - y_L)}$$

$$x_S = \frac{x_L \cdot y_P - x_P \cdot y_L}{y_P - y_L}, \quad z_S = \frac{z_L \cdot y_P - z_P \cdot y_L}{y_P - y_L}$$

- Maticově

$$\begin{bmatrix} x' \\ 0 \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} -y_L & x_L & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & z_L & -y_L & 0 \\ 0 & 1 & 0 & -y_L \end{bmatrix} \cdot \begin{bmatrix} x_P \\ y_P \\ z_P \\ 1 \end{bmatrix}$$

$$x_S = \frac{x'}{(y_P - y_L)}, \quad z_S = \frac{z'}{(y_P - y_L)}$$



# Projekční metody

## Zdroj světla v bodě

### Příklad

Ve scéně je umístěna krychle se souřadnicemi vrcholů  $[0, 20, 0]$ ,  $[10, 20, 0]$ ,  $[10, 20, 10]$ ,  $[0, 20, 10]$ ,  $[0, 30, 0]$ ,  $[10, 30, 0]$ ,  $[10, 30, 10]$ ,  $[0, 30, 10]$ . Určete tvar stínu, který vrhá, pokud je zdroj v bodě  $[0, 100, 100]$ .

# Projekční metody

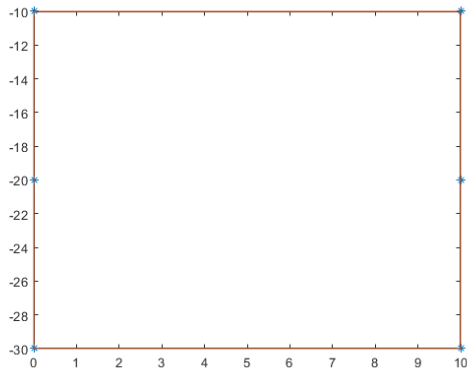
## Zdroj světla v bodě

### Příklad

Ve scéně je umístěna krychle. Určete tvar stínu, který vrhá, pokud je zdroj v bodě  $[0, 100, 100]$ .

body = 2x8

0	10	10	0	0	10	10	0
-20	-20	-10	-10	-30	-30	-20	-20



# Projekční metody

## Projekce na libovolnou plochu

- Doposud jsme promítali stín na zem, uvedené metody můžeme zobecnit, aby byl vržený stín **na libovolné ploše**

- Plocha leží v rovině  $\vec{n} \cdot X + d = 0$

$\vec{n}$  ... normálový vektor plochy

$X$  ... její libovolný bod

- Dosazením  $X$  za  $S$  ve vztahu

$$S = L + t(P - L)$$

získáme

$$t = \frac{\vec{n} \cdot \vec{L} + d}{\vec{n} \cdot (L - P)}$$

$$\vec{L} = L - [0, 0, 0]$$

- Maticově

$$M = \begin{bmatrix} \vec{n} \cdot \vec{L} + d - L_x \vec{n}_x & -L_y \vec{n}_x & -L_z \vec{n}_x & -n_x \\ -L_x \vec{n}_y & \vec{n} \cdot \vec{L} + d - L_y \vec{n}_y & -L_z \vec{n}_y & -n_y \\ -L_x \vec{n}_z & -L_z \vec{n}_z & \vec{n} \cdot \vec{L} + d - L_z \vec{n}_z & -n_z \\ -L_x d & -L_y d & -L_z d & \vec{n} \cdot \vec{L} \end{bmatrix}$$



# Projekční metody

Zdroj v nekonečnu

- Proces generování scény:
  - 1 Vygenerujeme scénu tak, jako obvykle
  - 2 Současnou transformační matici vynásobíme maticí stínu
  - 3 Opět vykreslíme objekty (tím, že byly vynásobeny maticí stínu, vykreslí se v rovině země)
- Stínový polygon může zasahovat i mimo cílovou plochu na kterou má dopadnout → musíme ho ořezat
- Díky numerickým chybám se může polynom (jeho části) promítnout lehce nad/pod rovinu a tím dojde ke ztmavení jen některých pixelů

# Projekční metody

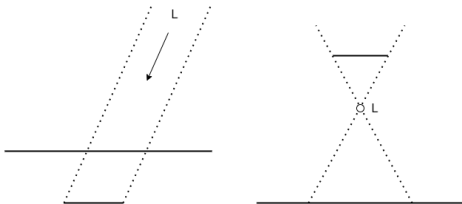
Zdroj v nekonečnu

- Lepší řešení je využít kombinaci z-bufferu a šablony (**stencil buffer**)
- Nejprve se vyřeší viditelnost scény běžným způsobem
- Pro každý výsledný pixel se do šablony zaznamená identifikátor viditelné plochy
- Ve druhé části se postupně vytváří polygony vržených stínů a opět se zobrazují z pohledu kamery
- U nich se neřeší viditelnost (neprovádí se hloubkový test), pouze se porovnává, zda v daném pixelu došlo ke shodě identifikátoru viditelné plochy a identifikátoru k ní příslušnému stínovému polygonu
- Pokud ano, je tento pixel ve stínu a je třeba ztmavit jeho barvu
- Musíme si zapamatovat i to, že pixel byl již v rámci právě vyhodnocovaného světelného zdroje jednou ztmaven, aby nedošlo k chybnému opakovanému ztmavení – **double blending**

# Projekční metody

## Nevýhody

- Algoritmus je určen jen pro scény obsahující jen objekty s ploškovou reprezentací – cílová plocha je částí roviny
- Transformace popsaná maticí zobrazuje zvolený objekt do plochy bez ohledu na jeho umístění ve scéně

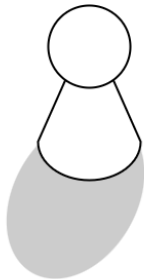


- K tomu, aby těleso vrhlo stín, musí ležet mezi plochou a světelným zdrojem
- V případě, že tomu tak není, vrhá **falešný stín**

# Projekční metody

## Zjednodušení

- Cílové plochy, na které dopadá stín, jsou pouze vybrané objekty – podlaha, stěna
- Extrémní zjednodušení – nepočítáme tvar stínu, ale vytváříme **pseudostín** (fake shadow) ve tvaru elipsy, jejíž rozměry odpovídají maximálním rozměrům objektu, který vrhá tento stín

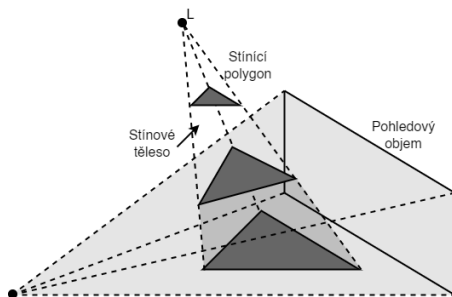


# Stínové těleso

- **Stínové těleso** – shadow volume
- Když světlo dopadne na objekt, vrhá stín za tímto tělesem
- Stínové těleso je definované pro každý světelný zdroj a polygon (stínové těleso je definované polygonem a světelným zdrojem)
- Existuje rozšíření i na nepolygonální plochy – nerovinné plochy a non-manifolds)
- Pracuje pouze s bodovým osvětlením → poskytuje pouze ostré stíny

# Stínové těleso

- **Stínové těleso** – ohraničuje část prostoru ve scéně, ze kterého není přes stínící objekt (polygon) světelný zdroj vidět (vymezuje zdrojem neosvětlený prostor)
- Tělesa, která leží v tomto stínovém objemu, nebudou zdrojem osvětlena
- Tmavá část představuje průnik stínového a pohledového objemu



# Stínové těleso

- Při zobrazování objektů musíme zjistit jejich vzájemnou polohu se stínovým tělesem (vzájemnou polohu polygonů aproximující tento objekt)
- **Vzájemná poloha:**
  - Polygon leží celý uvnitř stínového tělesa – leží ve stínu
  - Polygon leží celý vně stínového tělesa – je osvětlený světelným zdrojem
  - Polygon leží částečně ve stínovém tělese – je nutné provést rozdělení polygonu na osvětlenou a neosvětlenou část
- **Určení stínového tělesa** – pro jeden stínící polygon je to snadné
- Pro obecný objekt je nutné nejprve najít jeho obrys (hranici stínu)
- **Obrys** je tvořen obrysovými hranami (viz viditelnost) – každá obrysová hrana definuje jednu stěnu stínového tělesa
- Vytvoření stínových těles ze siluety není nutnou podmínkou, je možné vytvořit stínové těleso pro každou ke světlu přivrácenou plochu objektu, což zvyšuje počet stínových těles → prodlužuje výpočet stínu

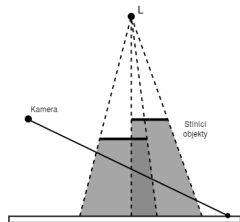
## Stínové těleso

- Při vykreslování polygonů pracujeme v rastru místo přesné geometrické reprezentace
- Ze středu promítání vysíláme (testovací) paprsky každým pixelem směrem ke zobrazovanému povrchu ve scéně
- Během cesty se počítá, kolikrát paprsek vstoupil do nějakého stínového tělesa a kolikrát je opustil
- Pokud je rozdíl nenulový, znamená to, že paprsek neopustil všechna stínová tělesa, do kterých vstoupil, a bod na povrchu tělesa, ke kterému dorazil, musí ležet ve stínu
- To, že paprsek vstoupil do stínového tělesa znamená, že protnul přivrácenou stěnu stínového tělesa
- To, že vystoupil znamená, že protnul odvrácenou stranu tělesa



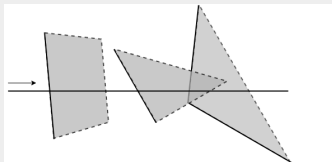
# Stínové těleso

- Počítadlo nastavíme na 0
- Vždy když paprsek protne přivrácenou stěnu nějakého stínového tělesa, přičteme jedna
- jedničku odečteme, když protne odvrácenou stranu



## Příklad

Do obrázku (představující pohled shora) zaznačte počty průchodů paprsku stínovými objemy.

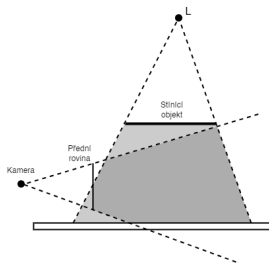


# Stínové těleso

- Tento postup se nazývá **Depth-pass algorithm**
- Předpokládali jsme ale, že pozorovatel neleží ve stínu (čítač začínal na 0)
- Je možné, že i pozorovatel leží ve stínu jednoho (i více) polygonů
- Musíme tedy určit, zda pozorovatel leží ve stínu, nebo ne
- Z bodu pozorovatele vedeme paprsek v libovolném směru směrem do „nekonečna“ a počítáme, kolikrát protne přivrácené/odvrácené stěny stínových těles
- „Nekonečno“ musí ležet mimo stínové těleso
- Stínové těleso je samo o sobě nekonečné. Můžeme brát jako průnik stínového tělesa s pohledovým objemem – uzavřít ho. Uzávěrům říkáme přední a zadní uzávěr (ve směru ke zdroji světla).
- Pokud protne více odvrácených než přivrácených stran, leží pozorovatel uvnitř stínového tělesa – čítač nezačíná od 0 (od čeho začíná?)

# Stínové těleso

- **Problém** může nastat, pokud je stínový polygon (objekt) ořezán přední nebo zadní rovinou pohledového objemu (near/far)
- Ve stínovém tělese mohou vzniknout trhliny → chybné určení stínu
- Při ořezávání stínového tělesa musíme přidat stěny ležící v přední a zadní rovině, aby bylo těleso uzavřeno
- Vyhnout se tomu můžeme i když obrátíme smysl tohoto algoritmu – nepočítáme průchody stínovými tělesy směrem od pozorovatele k pixelu, ale naopak = **Robust stencil volumes**



## Stínová paměť hloubky

- **Stínová paměť hloubky** – shadow depth map
- Pracuje v obrazovém prostoru, není přesný, může se objevit aliasing
- Pracuje s libovolnou reprezentací, ne jen ploškovou
- **Myšlenka**: Tělesa, která jsou nejbližší ke zdroji světla nejsou ve stínu / vrhají stín
- Pokud je nějaký objekt před jiným (ve směru od zdroje světla) vrhá na něj stín
- **Z-buffer shadow** (stínová mapa) – buffer uchovává informaci co je osvětleno zdrojem světla (přesnější by tedy bylo označení z-buffer illumination)
- V tomto bufferu uchováváme informaci o vzdálenosti od zdroje světla pro každou pozici, která je osvětlena
- Potřebujeme transformaci, která převede zdroj světla do místa pozorovatele a namíří jej do středu scény (souřadný systém světla musíme převést na souřadný systém kamery = pozorovatele)

## Stínová paměť hloubky

### Příklad

Zdroj světla je v souřadném systému kamery na pozici  $[0, 30, -30]$ , střed scény je v bodě  $[0, 0, -20]$ . Jak bude vypadat transformace, která převede souřadný systém světla do souřadného systému kamery?

## Stínová paměť hloubky

### Příklad

Zdroj světla je v souřadném systému kamery na pozici  $[0, 30, -30]$ , střed scény je v bodě  $[0, 0, -20]$ . Jak bude vypadat transformace, která převede souřadný systém světla do souřadného systému kamery?

- Otočením o  $-45^\circ$  kolem osy  $x$  převedeme zdroj světla na pozici  $[0, 0, 42.4]$
- Posuneme zdroj světla o  $-42.4$  ve směru osy  $z$  a tím ho posuneme do počátku

## Stínová paměť hloubky

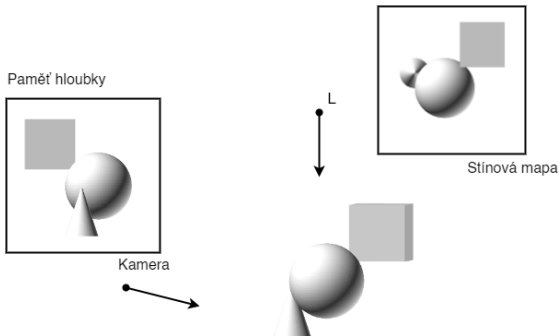
### Příklad

Zdroj světla je v souřadném systému kamery na pozici  $[0, 30, -30]$ , střed scény je v bodě  $[0, 0, -20]$ . Jak bude vypadat transformace, která převede souřadný systém světla do souřadného systému kamery?

- Otočením o  $-45^\circ$  kolem osy  $x$  převedeme zdroj světla na pozici  $[0, 0, 42.4]$
- Posuneme zdroj světla o  $-42.4$  ve směru osy  $z$  a tím ho posuneme do počátku

## Stínová paměť hloubky

- Transformaci pak aplikujeme na všechny objekty (na scénu se díváme z pozice zdroje světla)
- Na scénu teď aplikujeme standardní z-buffer algoritmus
- Pro scénu s více zdroji světla bude existovat více bufferů





## Stínová paměť hloubky

- Jakmile máme spočítanou stínovou mapu, renderujeme scénu (např. pomocí z-buffer algoritmu)
- Úprava – pro každý bod objektu zjišťujeme, zda má být vykreslen nebo ne, pokud ano, aplikujeme na něj inverzní transformaci k transformaci, kterou jsme použili pro výpočet stínové mapy
- Z minulého příkladu posun o 42.4 ve směru  $z$  a pak otočení o  $45^\circ$  kolem  $x$
- Bod na souřadnicích  $[x, y, z]$  v souřadném systému kamery přenášíme na  $[x', y', z']$  souřadného systému zdroje světla
- Ve stínové mapě máme uloženu informaci o nejbližším (nezastíněném) bodě
- Porovnáme  $z'$  s hodnotou mapy na souřadnicích  $[x', y']$ , pokud je tato hodnota menší, je objekt zastíněn, pokud je shodná zastíněn není

# Stínová paměť hloubky

- Kvalita vytvářených stínů závisí na rozlišení stínové mapy
- Kvůli rasterizaci (diskrétnímu z-bufferu) vzniká alias
- Zaokrouhlovací chyba v  $z'$  a stínové mapě – polygony vrhají stín sami na sebe  
**self-shadow alias**
- Řešení:
  - Přidáme nějakou toleranci
  - Oři transformaci posuneme bod  $z'$  blíže ke zdroji světla a kontrolujeme, zda je  $\leq$  než hodnota stínové mapy

# Polostíny

- Polostín vzniká při použití plošného světla místo bodového
- Z plošného světla vybereme náhodně několik bodů  
Je-li světlo koule, náhodně vybereme několik bodů uvnitř koule
- Počet bodů závisí na velikosti zdroje, čím je větší, tím by jich mělo být více
- Každý tento bod je pak považován za bodové světlo
- Pokud máme  $n$  bodů, každá zdroj scénu osvětluje intenzitou  $I/n$ , kde  $I$  je původní intenzita