

# Základy počítačové grafiky

## Ray tracing

Jiří Havel  
Michal Španěl

Ústav počítačové grafiky a multimédií  
Fakulta informačních technologií  
Vysoké učení technické v Brně

Brno 2013

## Cíl

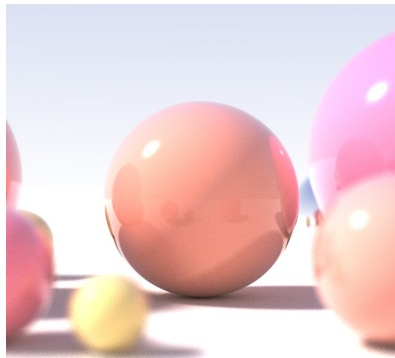
Cílem této přednášky je seznámit se s první/základní metodou realistického zobrazení 3D scény.

Zařadíme Ray tracing do kontextu ostatních zobrazovacích metod. Vysvětlí princip metody a definuje její vlastnosti.

# Realistické zobrazení

## Realističnost

- Generujeme zobrazení virtuální scény, na základě počítačového modelu
- Požadujeme, aby získané zobrazení maximálně odpovídalo realitě
- Foto-realistické zobrazení, srovnání s fotografií
- Posuzujeme vizuální charakteristiky výsledku
- Lidsky subjektivní posouzení realističnosti



# Realistické zobrazení

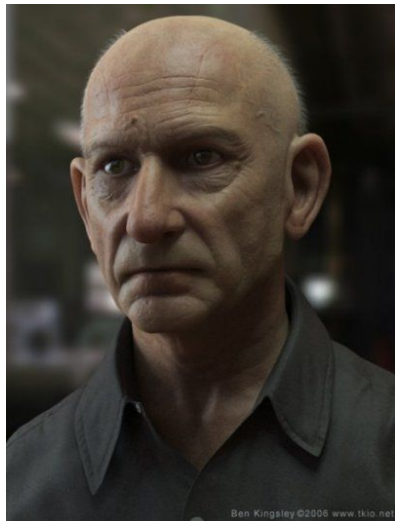
Je to fotografie nebo počítačově generovaný obraz?



# Faktory realistického zobrazení

## Významné faktory

- Geometrie objektů (polygonální model atd.)
- Vliv prostředí (vzduch, voda, vakuum, mlha, atd.)
- Vlastnosti světla (paprsek, energie, kvanta, atd.)
- Optické vlastnosti materiálů (difuze, reflexe, lom, průhlednost, atd.)
- Struktura povrchů (drsnost, reliéf, struktura, textura, anizotropie, atd.)



# Nereálnost realistického zobrazení

## Je to možné?

- Skutečně realistické zobrazení není možné!
- Realita je stochastická, ne empirická / deterministická
- Naše možnosti modelování jsou velmi omezené:
  - Zjednodušená polygonální geometrie
  - Prostředí jako vakuum
  - Empirické osvětlovací modely, omezené
  - Struktura povrchu jako textura, izotropie
  - Globálnost osvětlení



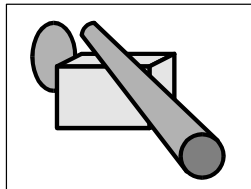
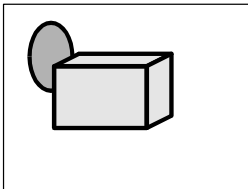
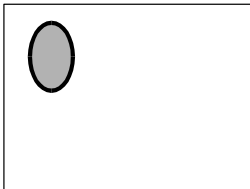
# Realistické zobrazení

## Definice

*Realistické zobrazení* je proces generování obrazu virtuální počítačové scény tak, aby vizuální vlastnosti a kvalita výsledného zobrazení odpovídaly obrazu (fotografii) příslušné reálné scény

# Objektová vizualizace

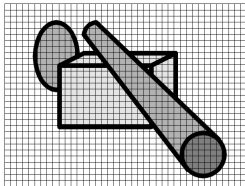
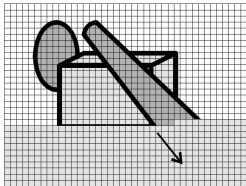
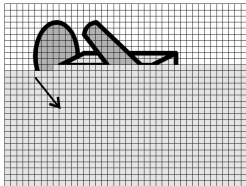
- Objekty scény jsou zpracovávány sekvenčně
- Viditelnost je řešena explicitně, nejčastěji prostřednictvím Z-buffer
- Nejsou řešeny žádné vztahy mezi jednotlivými objekty, proto:
  - Objekty nevrhají stíny
  - Objekty scény se nestávají sekundárními zdroji světla
  - Není zrcadlení okolí na povrchu lesklých objektů
- Toto je standardní zobrazovací metoda s HW akcelerací. Takto pracují knihovny jako je OpenGL a Direc3D





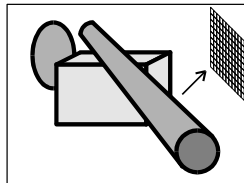
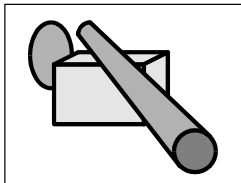
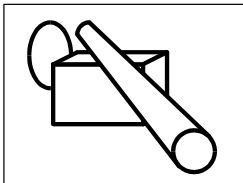
# Obrazová vizualizace

- Pixely obrazu scény jsou zpracovávány sekvenčně
- Scéna je prohledávána z pohledu všech pixelů zobrazení scény, implicitní řešení viditelnosti
- Geometrický přístup, ale neřeší globální vztahy ve scéně, proto:
  - Bodové zdroje světla, ostré stíny
  - Zrcadlení okolí na povrchu lesklých objektů
  - Objekty scény se nestávají sekundárními zdroji světla
- Představitelem tohoto přístupu je Ray tracing



# Komplexní vizualizace

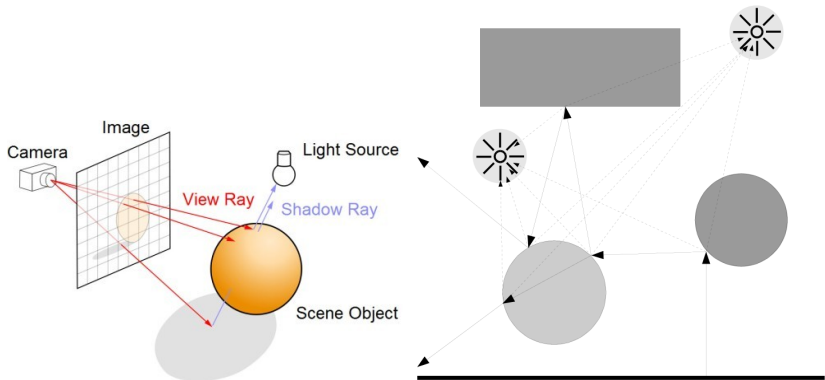
- Scéna je zpracovávána komplexně, celá najednou
- Zobrazení výsledku nemusí být součástí zpracování
- Detailní vlastnosti výsledného zobrazení závisí na použité zobrazovací metodě, objektová / obrazová
- Globální řešení osvětlení ve scéně, energeticky / stochasticky:
  - Implicitní řešení stínů, plošné zdroje světla, měkké stíny
  - Objekty scény se stávají sekundárními zdroji světla
- Představitelem tohoto přístupu je Radiosita



# Raytracing

## Zpětné (rekurzivní) sledování paprsku

- Sledujeme paprsek od kamery zpět ke zdroji světla.
- Hledáme množství světla, které paprsek přináší.



# Druhy paprsků

## Primární

- Vychází z kamery - pixely obrazovky.
- Společný počátek (nebo rovnoběžné).

## Stínové

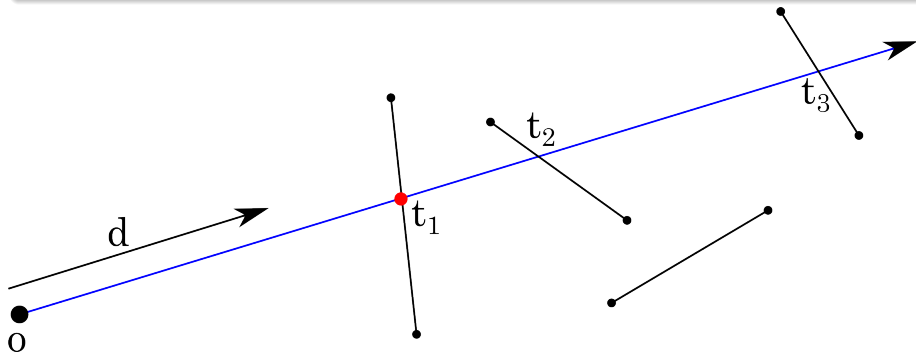
- Z místa dopadu paprsku do každého světla.
- Jen vidíme/nevidíme, přesný průsečík nás nezajímá.
- Budeme pro toto světlo počítat osvětlovací model?

## Sekundární

- Vycházejí z místa dopadu primárního/sekundárního paprsku.
- Vznikají odrazem a lomem.
- Chaotičtější než primární paprsky.

# Paprsek

- Polopřímka nebo úsečka.
- $\vec{p} = \vec{o} + t\vec{d}$
- Hledáme nejbližší průsečík (nejmenší  $t$ ).

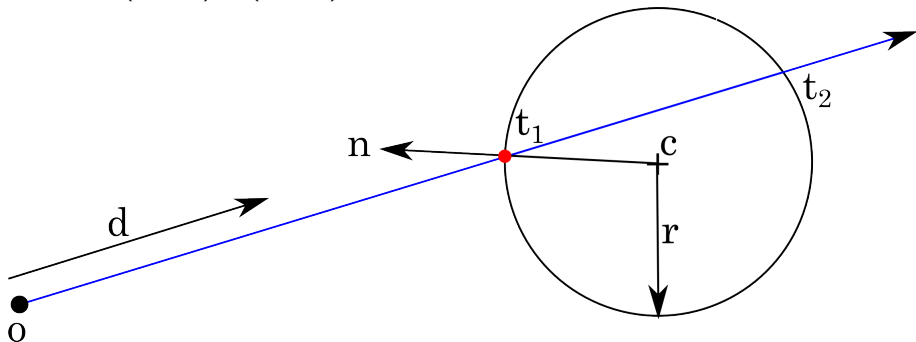


# Jednoduchý objekt - koule

$$(\vec{o} + t\vec{d} - \vec{c})^2 = r^2$$

$$t^2\vec{d}^2 + 2t\vec{d}(\vec{o} - \vec{c}) + (\vec{o} - \vec{c})^2 - r^2 = 0$$

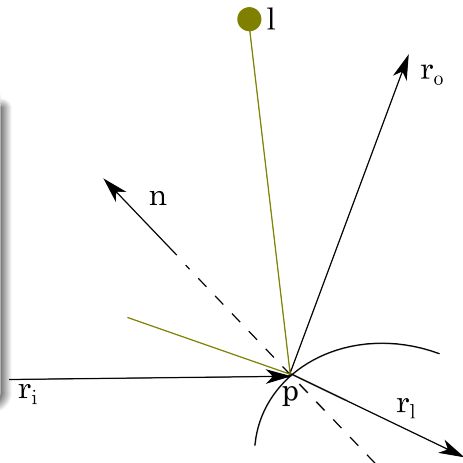
$$\vec{n} = \frac{\vec{o} + t\vec{d} - \vec{c}}{|\vec{o} + t\vec{d} - \vec{c}|}$$



# Osvětlení průsečíku

$$\text{color} = \text{phong}(\vec{r}_i, \vec{p}, \vec{n}, \vec{l}) + k_o \text{trace}(\vec{p}, \vec{r}_o) + k_l \text{trace}(\vec{p}, \vec{r}_l)$$

- Pokud je světlo vidět (stínový paprsek), vypočti osvětlovací model.
- Nezávisle přidej barvu z odraženého a lomeného paprsku.
- Koeficienty  $k_o$  a  $k_l$  závisí na materiálu a úhlech.



# Naivní složitost

## Primární paprsky

- Okno  $w \times h$ ,  $n$  objektů ve scéně.
- $wh$  paprsků, **nwh** výpočtů průsečíků.

## + světla

- $l$  světel.
- Paprsek do každého světla -  $nwh + lnwh = (l + 1)nwh$  výpočtů.

## + odraz

- $r$  odrazů (hloubka rekurze).
- $(r + 1)(l + 1)nwh$  výpočtů.

## + lom

- $(1 + 2 + 4 + \dots + 2^r)(l + 1)nwh = (2^{(r+1)} - 1)(l + 1)nwh$



# Co můžeme vylepšit?

$$(2^{(r+1)} - 1)(l + 1)nwh$$

- Optimalizovat výpočet průsečíku.
- Zmenšit rozlišení ( $wh$ ).
- Zjednodušit scénu ( $n$ ).
- Ubrat světél ( $l$ ).
- Omezit rekurzi ( $r$ ).

Ale tak, aby to nebylo poznat :)

# Optimalizace

## Rekurze

- Odražené paprsky ovlivňují obraz míň.
- Máme jen 8b na kanál.
- A víme dopředu, kolik bitů paprsek ovlivní.

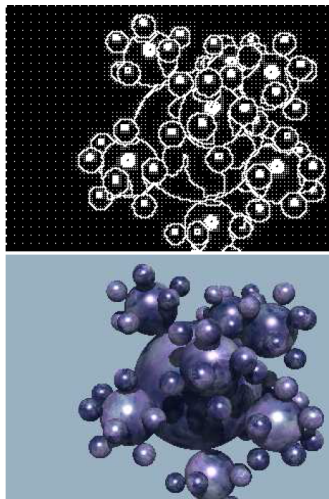
## Světla

- Intensita klesá se čtvercem vzdálenosti.
- Omezený dosvit.
- Pokud je světlo dost daleko, můžeme ho ignorovat.

# Menší rozlišení

## Adaptivní podvzorkování

- Nastřílet paprsky řídce.
- Podle potřeby vzorkovat hustěji.
- Malé objekty můžeme minout úplně :(



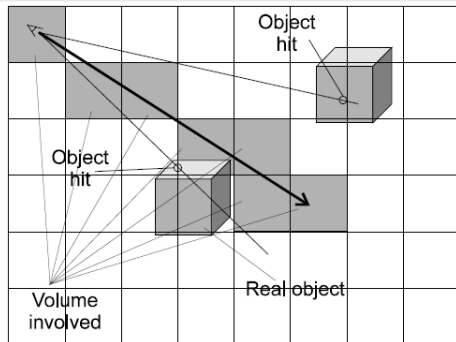
# Rozdělení scény

## Cíl

- Zahodit více objektů jednoduchým testem.
- Projít objekty ve směru paprsku.

## Různé druhy dělení

- Obalová tělesa.
- Hierarchie obálek.
- Uniformní mřížka.
- Binární stromy (BSP, KD).
- Quadtree, Octree.
- ...



# Co jsme zanedbali?

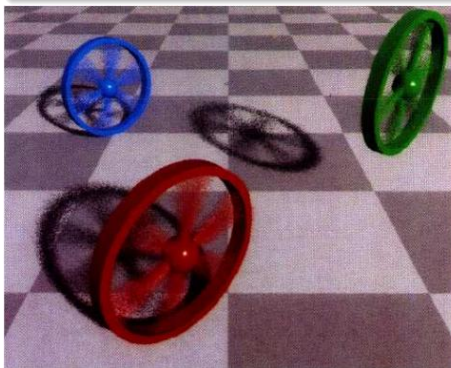
- Světla jsou bodové.
- Odraz a lom jsou idealizované.
- BRDF se počítá jen pro přímé osvětlení.
- Zobrazuje se v jednom okamžiku.
- Čočka kamery je bez chyb.

## Chce to víc paprsků!

- Do více bodů světla - neostré stíny.
- Odraz a lom do více směrů - matné odrazy, průsvitné sklo.
- Víc paprsků jedním pixelem.
  - Rozprostřené v čase - motion blur.
  - Různým místem čočky - depth of field.

# Distributed raytracing

- Distribution - rozložení pravděpodobnosti.
- Paprsky se větví do svazků.
- Získané barvy se průměrují s váhovou funkcí.
- Strom paprsků



# Path tracing

- Simuluje se dráha fotonu.
- Náhodně se vybírá jeden z možných směrů.
- Posílá se mnoho paprsků pro jeden pixel.
- Mnoho jednoduchých drah.

