

# Pokročilé metody fotorealistického zobrazování

Ondřej Bahounek

14.5.2013

## Úvod

Motivace

Základní informace

## Ray-tracing

Složitost

Optimalizace

## Závěr

Shrnutí metod

Představení programu RayTracer

Reference

# Motivace

- ▶ snaha o vytvoření realistických obrázků
- ▶ konstrukce nejen reálných, ale i nereálných scén
- ▶ současnost: kombinace fotek s uměle vytvořenými obrázky
- ▶ použití: filmy, obrazové efekty, počítačové hry, architektura, ...

# Fotorealistické zobrazování

- ▶ populární oblast počítačové grafiky
- ▶ úzce spjata s ostatními obory:
  - ▶ matematika ((analytická) geometrie, diferenciální rovnice, ...)
  - ▶ fyzika (optika)
  - ▶ a samozřejmě programování
- ▶ různé přístupy k modelování scény:
  - ▶ empirický (Phongův model)
  - ▶ fyzikální (Straussův model)
  - ▶ „programátorský“ (Cook-Torrance model)

# Požadavky

- ▶ realistický výstup je standard obsahující:
  - ▶ materiály: sklo, zrcadlo, guma, měď, ...
  - ▶ základní efekty: stínování, stíny, odraz a lom světla
  - ▶ pokročilé efekty: kaustiky, motion blur, ...
- ▶ důraz se však klade na rychlost
  - ▶ naivní přístup v reálném čase nestačí
    - ▶ s vývojem výpočetní techniky rostou i požadavky na složitější efekty
  - ▶ optimalizační techniky

# Objekty scény

- ▶ základní tělesa: rovina, koule, kvádr, válec, kužel
- ▶ obecné těleso:
  - ▶ tvořené trojúhelníkovými ploškami
  - ▶ Phongova interpolace normál pro lepší výsledek
    - ▶ využití barycentrických souřadnic
- ▶ společná vlastnost objektů:
  - ▶ umět spočítat průsečík s paprskem
  - ▶ každý objekt má své materiálové vlastnosti:
    - ▶ barva, osvětlovací koeficienty, index lomu, koeficienty odrazu a lomu
    - ▶ ... závisí na zvoleném osvětlovacím modelu

# Ray-tracing

- ▶ populární zobrazovací technika (1980)
- ▶ na bázi zpětného sledování paprsku (vycházející od pozorovatele)
- ▶ paprsek prochází scénou a počítá barvu
- ▶ na rozdíl od ray-castingu:
  - ▶ rekurzivní sledování paprsku
  - ▶ globální osvětlovací model
  - ▶  $\Rightarrow$  pomalejší výpočet, ale realističtější výstup

# Složitost

- ▶ naivní metoda:
  - ▶ **lineární** složitost
  - ▶ ke každému paprsku prochází všechny objekty scény
  - ▶ u složitějších scén nepřijatelné
- ▶ potřeba optimalizace
  - ▶ snaha o **logaritmickou** složitost



# Optimalizace

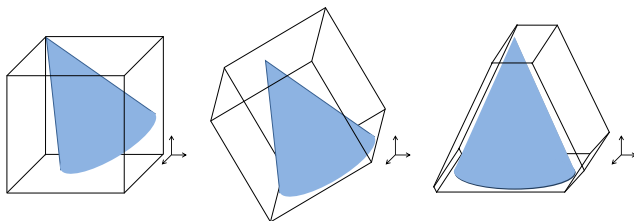
- ▶ porovnávání optimalizačních metod:
  - ▶ celkový čas výpočtu (nepřesné, subjektivní)
  - ▶ počet výpočtů průsečíků paprsku s objektem
- ▶ typy optimalizace:
  - ▶ efektivní výpočet průsečíku paprsku s objektem
  - ▶ akcelerující struktury
  - ▶ optimalizace stínových paprsků
  - ▶ optimalizace kódu

# Výpočet průsečíku

- ▶ zabírá 90% algoritmu ray-tracingu
- ▶ jednoduché objekty mají jednoduché výpočty
- ▶ objekty v základní poloze mají jednodušší výpočty:
  - ▶ objekt potřebujeme umět orientovat libovolně
  - ▶ využití homogenních souřadnic a afinních transformací
  - ▶ potřeba dekompozice rotační matice (Eulerovy úhly)
  - ▶ jednoznačný převod pomocí kvaternionů

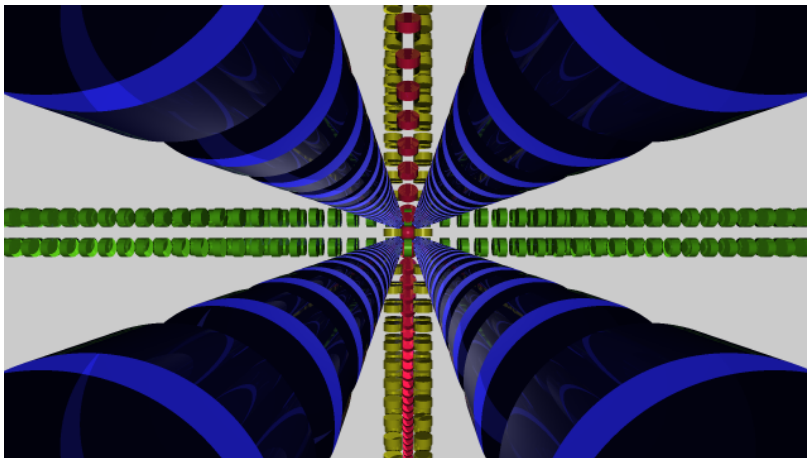
# Obálka

- ▶ ke každému objektu lze sestavit tzv. obálku
  - ▶ přiléhavost obálky vs. složitost výpočtu průsečíku
  - ▶ mnoho různých typů obálek
  - ▶ nejčastěji osově orientovaný kvádr
- ▶ před výpočtem průsečíku s objektem se nejprve otestuje průsečík s obálkou
- ▶ ještě nepřináší zrychlení ray-tracingu



# Akcelerující struktury

- ▶ zařadíme obálky do hierarchické struktury (strom)
- ▶ v listech uloženy objekty (jejich obálky)
- ▶ každý uzel obsahuje obálku vytvořenou ze svých potomků
- ▶ prochází-li paprsek obálkou uzlu, pokračuje k jeho potomkům; jinak zamítne celý podstrom
- ▶ snaha o vyvážený strom – logaritmická složitost
- ▶ 2 přístupy:
  1. prostorové rozdělení (mřížka, octree, kd-tree, BSP-tree)
  2. objektové rozdělení (r-tree, obecné hierarchie obálek)

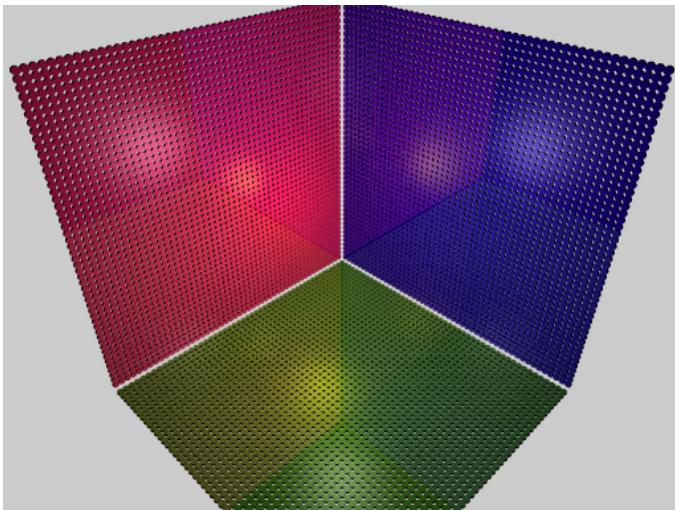


Obrázek: Scéna k porovnání metod 1 (1 313 válců).

technika optimalizace	počet výpočtů s objektem	počet výpočtů s obálkou
naivní	$24\,225 \cdot 10^6$	—
r-tree	$95 \cdot 10^6$	$1\,033 \cdot 10^6$
octree	$94 \cdot 10^6$	$466 \cdot 10^6$
kd-tree	$92 \cdot 10^6$	$608 \cdot 10^6$

Tabulka: Data ke scéně z obrázku 1.

- ▶ scéna: 1 313 válců, 2 bodová světla
- ▶ rozlišení:  $640 \times 360$
- ▶ hloubka rekurze: 1
- ▶ antialiasing: ano



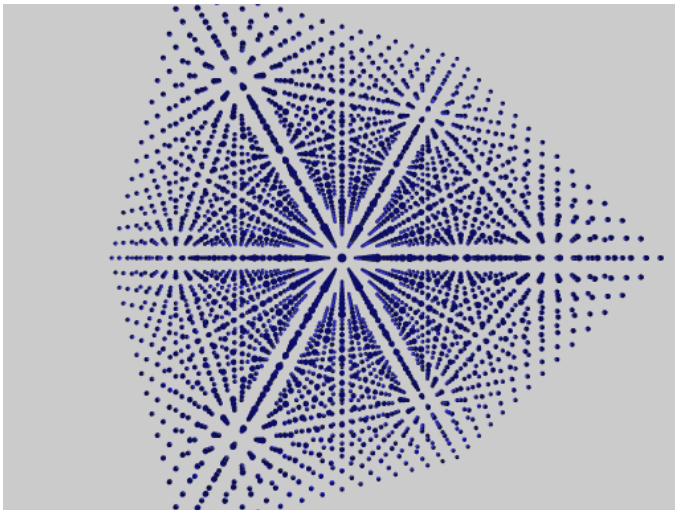
Obrázek: Scéna k porovnání metod 2 (7 500 válců).

technika optimalizace	počet výpočtů s objektem	počet výpočtů s obálkou
naivní	$67\,086 \cdot 10^6$	—
r-tree	$21 \cdot 10^6$	$308 \cdot 10^6$
octree	$21 \cdot 10^6$	$144 \cdot 10^6$
kd-tree	$21 \cdot 10^6$	$145 \cdot 10^6$

Tabulka: Data ke scéně z obrázku 2.

- ▶ scéna: 7 500 válců, 2 bodová světla
- ▶ rozlišení:  $320 \times 240$
- ▶ hloubka rekurze: 1
- ▶ antialiasing: ano





Obrázek: Scéna k porovnání metod 3 (3 375 koulí).

technika optimalizace	počet výpočtů s objektem	počet výpočtů s obálkou
naivní shadow-naive	$3\,169,1 \cdot 10^6$	—
naivní shadow-opt	$2\,293,9 \cdot 10^6$	—
kd-tree shadow-naive	$0,7 \cdot 10^6$	61 904 812
kd-tree shadow-opt	$0,6 \cdot 10^6$	61 904 812

**Tabulka:** Porovnání optimalizací vržených stínů k obrázku 3.

- ▶ scéna: 3 375 koulí, 2 bodová světla
- ▶ rozlišení:  $320 \times 240$
- ▶ hloubka rekurze: 0
- ▶ antialiasing: ano

# Shrnutí metod

- ▶ testování probíhalo na různých typech scén
- ▶ nejlepší výsledky vždy struktura kd-tree
  - ▶ vzhledem k její adaptibilitě na rozložení scény
- ▶ avšak ostatní struktury nezaostávaly o moc
- ▶ dělení objektů pomocí r-tree bylo vyzkoušeno prvně
  - ▶ nedoporučuje se – obsahuje vždy více testů s obálkami

# Představení programu RayTracer

- ▶ napsán v jazyce C# pod platformou .NET 3.5
- ▶ interaktivní 3D editor pro ray-tracing
- ▶ výstup v podobě obrázků i animací
  - ▶ různá rozlišení i formáty
  - ▶ animaci s trajektorií elipsy lze libovolně nastavit
- ▶ libovolný počet objektů i světelných zdrojů
- ▶ antialiasing (supersampling  $3 \times 3$ )
- ▶ měkké stíny
- ▶ import a export scény

# Reference



AKENINE-MÖLLER, T., HAINES, E., HOFFMAN, N. *Real-Time Rendering*. 3rd ed. 2008. ISBN 978-1-56881-424-7.



DUNN, F., PARBERRY, I. *3D Math Primer for graphics and Game Development*. Wordware Publishing, 2002. ISBN 1-55622-911-9.



EBERLY, D. H. *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*. Morgan Kaufmann Publishers, Academic Press, 2000. ISBN 1558605932.



KAY, T. L., KAJIYA, J. T. Ray Tracing Complex Scenes. *ACM SIGGRAPH Computer Graphics*, 1986, vol. 20, no. 4, p. 269–278.



KNUTH, D. E. Structured Programming with Goto Statements. *Computing Surveys*, 1974, vol. 6, no. 4, p. 261–301.



PHARR, M., HUMPHREYS, G. *Physically Based Rendering: From Theory to Implementation*. 2nd ed. Elsevier, 2010. ISBN 978-0-12-375079-2.



WILLIAMS, A., BARRUS, S., MORLEY, R. K., et al. An Efficient and Robust Ray-Box Intersection Algorithm. *Journal of Graphics Tools*, 2005, vol. 10, no. 1, p. 49–54.

Děkuji za pozornost.