

FAST RAYTRACING USING SPATIAL SUBDIVISION

Martin Stríž

Bachelor Degree Programme (3), FIT BUT

E-mail: xstriz02@stud.fit.vutbr.cz

Supervised by: Adam Herout

E-mail: herout@fit.vutbr.cz

ABSTRACT

Raytracing is one of frequently used methods in rendering 3D scenes. Although it is not a photorealistic method, it produces results with high image quality in relatively short rendering time. This paper describes spatial subdivision methods, namely BSP trees and KD trees, and compares their effectivity.

1 ÚVOD

Zobrazení metodou sledování paprsku (*raytracing*) je jednou z používaných metod zobrazení 3D scény. Dosahuje vysoké kvality zobrazení při relativně rychlém zpracování.

2 ZOBRAZENÍ METODOU SLEDOVÁNÍ PAPRSKU

Princip metody je následující:

1. Pro každý bod obrazu vyšleme paprsek z kamery.
2. Najdeme nejbližší objekt, který byl paprskem zasažen a vypočítáme průsečík.
3. Určíme intenzitu rozptýleného světla pro získaný průsečík.
4. Pokud povrch odráží světlo, vyšleme rekurzivně podle zákona odrazu nový paprsek.
5. Sečtením rozptýleného a odraženého světla získáme výslednou barvu bodu.

3 OPTIMALIZACE POMOCÍ DĚLENÍ PROSTORU

Výše uvedený přístup je na první pohled velmi neefektivní, protože s každým paprskem se provádí výpočty průsečíků se všemi objekty. Se zvyšující se složitostí scény značně stoupá časová náročnost, proto je algoritmus nutné optimalizovat. Jednou z možností optimalizace je rozdělit prostor scény tak, aby se při zobrazování minimalizovalo množství počítaných průsečíků s objekty. Způsobů existuje celá řada, zde se budeme zabývat pouze BSP a KD stromy, protože jsou v současnosti nejvíce používány.

3.1 BSP STROMY

BSP (*binary space partitioning*) je metoda, která rekurzivně dělí prostor na dva podprostory pomocí dělicí roviny. Výsledkem dělení je datová struktura známá jako BSP strom. Konstrukce stromu probíhá rekurzivně, vstupem je osově zarovnaný kvádr (*voxel*) ohraničující scénu. Při každém kroku rekurze je rozdělen na polovinu podle nejdelší osy. Následně je provedeno rozdělení objektů podle toho, do které poloviny náleží. Pokud objekt protíná dělicí rovina, je přiřazen do obou polovin. Rekurzivní dělení je prováděno do té doby, než je splněno jedno ze dvou ukončujících kritérií. Prvním je minimální počet objektů, druhým maximální hodnota rekurze. Výsledný strom obsahuje dva typy uzlů:

- Vnitřní uzel: obsahuje rozměr dělení (X, Y, Z) a pozici dělicí roviny
- List: obsahuje seznam objektů scény, které leží v daném podprostoru

Procházení struktury stromu při vykreslování scény je řešeno algoritmem TA_{rec}^B , který je podrobně popsán v [1].

3.2 KD STROMY

KD stromy mají podobnou strukturu jako BSP stromy, liší se pouze v umístění dělicí roviny. V případě BSP stromu je umístěna vždy doprostřed děleného voxelu, při použití KD stromu je pozice dělicí roviny libovolná.

Vhodná pozice je taková, která vytvoří co největší prázdný prostor. Tím se značně sníží počet testů průsečíků s objekty. Pro zjištění optimální pozice dělicí roviny použijeme heuristiku SAH (*surface area heuristic*), která je založena na geometrické pravděpodobnosti. Pravděpodobnost jevu, kdy paprsek protne voxel, je závislá na jeho povrchu. Pokud známe pravděpodobnost, můžeme ji použít k výpočtu ceny za průchod voxellem:

$$K = C_T + P \cdot C_I \cdot N \quad (1)$$

C_T a C_I jsou vhodně zvolené konstanty pro cenu za průchod a cenu za průsečík s objektem, N označuje počet objektů a P pravděpodobnost zásahu paprskem.

Při dělení voxelu V na dvě části A a B můžeme cenu vypočítat jako součet cen obou voxelů:

$$K = C_T + C_I (P_A N_A + P_B N_B) \quad (2)$$

$$K = C_T + C_I \left(\frac{S_A}{S_V} N_A + \frac{S_B}{S_V} N_B \right) \quad (3)$$

Pravděpodobnosti podle SAH odpovídají podílům povrchů dílčích voxelů k voxelu původnímu. Pro každou možnou pozici dělicí roviny vypočítáme cenu za rozdělení K a poté z nich vybereme minimum. Protože je počet pozic nekonečný, omezíme výpočty pouze na takové pozice, kde se mění počet objektů v dílčích voxelích, tzn. na okrajích jednotlivých objektů. Pokud je minimální cena za rozdělení větší než cena neděleného voxelu

$$K = C_I N_V,$$

nebudeme v dělení pokračovat. Více o cenových modelech a SAH naleznete v pracích [1] a [3].

4 TESTOVÁNÍ

Testování bylo provedeno ve vlastním experimentálním raytraceru implementovaném v jazyce C++ na platformě GNU/Linux. Charakteristiky testovacích scén jsou popsány tabulce číslo 1.

Scéna	Trojúhelníky	Koule	Světla
ships	1004	13	2
tree	180182	0	7
balls	2	597871	3

Tabulka 1: Charakteristiky testovacích scén.

Vykreslování bylo prováděno v rozlišení 1024×1024 s odrazy světla do třetí úrovně a stíny.

Scéna	Konstrukce		Vykreslení		Celkem		
	BSP	KD	BSP	KD	BSP	KD	bez BSP/KD
ships	0,01 s	0,01 s	4,58 s	3,85 s	4,59 s	3,86 s	120,35 s
tree	0,97 s	2,22 s	1,32 s	0,82 s	2,29 s	3,04 s	13 h 46 min
balls	0,70 s	4,58 s	94,42 s	12,13 s	95,12 s	16,71 s	19 h 52 min

Tabulka 2: Naměřené výsledky.

5 ZÁVĚR

Pomocí vlastního raytraceru jsem srovnal rychlosti vykreslování bez použití optimalizací a s použitím BSP a KD stromů. Z tabulky číslo 2 je vidět, že pro komplexní scény s velkým počtem objektů je postup bez optimalizace prakticky nepoužitelný vzhledem k dlouhému času zpracování.

Při použití BSP stromů pro dělení prostoru dosáhneme značného snížení počítaných průsečíků s objekty a tím i zrychlení výpočtu. KD stromy poskytují navíc možnost efektivní konstrukce využitím heuristiky, která výkon tohoto řešení ještě zvýší.

REFERENCE

- [1] HAVRAN, V.: *Heuristic Ray Shooting Algorithms*. Dizertační práce, Department of Computer Science and Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Listopad 2000.
URL <<http://www.cgg.cvut.cz/~havran/phdthesis.html>>
- [2] WALD, I.: *Realtime Ray Tracing and Interactive Global Illumination*. Dizertační práce, Computer Graphics Group, Saarland University, 2004.
URL <<http://www.mpi-sb.mpg.de/~wald/PhD/>>
- [3] WALD, I.; HAVRAN, V.: On building fast kd-trees for ray tracing, and on doing that in $O(N \log N)$. Zář 2006: s. 61–69.