

PHOTON MAPPING

Tomáš Lysek

Master Degree Programme (1), FIT BUT

E-mail: xlysek03@stud.fit.vutbr.cz

Supervised by: Pavel Zemčík

E-mail: zemcik@fit.vutbr.cz

Abstract: The paper focuses one of the global illumination methods – the photon mapping method. After short summarization of photon mapping, the paper presents a decomposition of the method into a set of relatively simple algorithms and attempts to identify the computational bottleneck. For the computationally expensive algorithms used in photon mapping method, the paper also proposes some optimization techniques.

Keywords: photon mapping, global illumination, kd tree

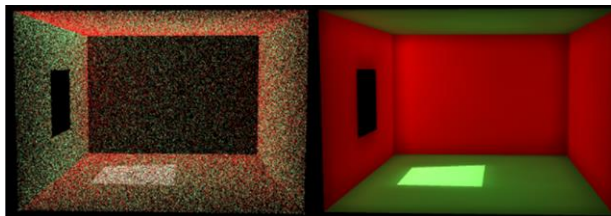
1. ÚVOD

Již od začátku vývoje počítačové grafiky byla snaha o co nejfotorealističtější výsledky obrazu. Bylo představeno mnoho fotorealistických metod od jednoduchých jako raytracing přes radiozitu až po složitější jako je například Path tracing. Tyto metody se liší různou mírou fotorealističnosti a taky časem zpracování. Jednou z nejnovějších metod, je metoda photon mappingu, která generuje velmi dobré fotorealistické výsledky a která je také (oproti path tracingu a radiozitě) rychlejší[1].

2. PHOTON MAPPING

Photon mapping je dvouprůchodová metoda globalního osvětlení, kterou představil Henrik Wann Jensen roku 1996 [1]. Metoda je založena na aproximaci zobrazovací rovnice[2] určením míry osvětlení z nejbližších fotonů bodu, pro který se osvětlení počítá. Pro výpočet nejbližších fotonů je nutné nejprve vypočítat fotonovou mapu pro celou scénu. A proto je tato metoda dvouprůchodová.

V prvním průchodu jsou vystřelovány fotony ze světelných zdrojů, fotony odpovídají zlomku energie světelného zdroje. Fotonům je prošetřována dráha kudy putují a pokud dopadnou na difúzní povrch jsou uloženy do fotonové mapy a dále distribuovány do scény směrem, který udává typ materiálu [3]. Fotony které projdou průhledným objektem vytváří na povrchu difúzního materiálu kaustiky a tyto fotony jsou uloženy v extra fotonové mapě.



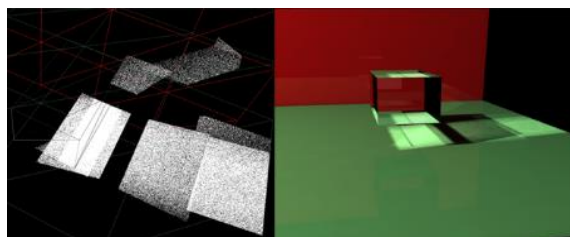
Obrázek 1: Nepřímé osvětlení¹

Výsledný obraz je možné získat několika způsoby. Pravděpodobně nejrealističtější výsledky má použít distribuovaného raytracingu [3]. Velmi dobré výsledky má také klasický raytracing doplněný o

¹ Obrázek je možné stáhnout v plném rozlišení na adrese <http://www.lyso.cz//dp/indirect.jpg>

výpočet nepřímého osvětlení a kaustik, kdy se místo ambientní složky světla, u phongova osvětlovačného modelu, připočte osvětlení odpovídající N nejbližším fotonům z fotonové mapy.

Na obrázku 1 je zobrazena scéna pomocí photon mappingu. V levé části obrazu je vizualizována fotonová mapa, v pravé části je zobrazena scéna pomocí raytracingu. Podobně i u obrázku 2, kde se jedná o kaustiky. Tyto obrázky byly pořízeny pomocí technik uvedených níže.



Obrázek 2: Kaustiky²

3. FUNKČNÍ BLOKY PHOTON MAPPINGU A JEJICH ANALÝZA

Úloha photon mappingu se dá rozdělit do několika funkčních bloků, podúloh, které budou postupně popsány a zanalyzovány. Časy a počty, které budou u jednotlivých bloků uvedeny, jsou zde uvedeny jen pro představu a pro porovnání mezi sebou, na jiných strojích se můžou a budou lišit. Bude se předpokládat, že zobrazována scéna se skládá z trojúhelníku a scéna se bude zobrazovat pomocí raytracingu.

Načtení a předzpracování scény

První blok se skládá z načtení scény ze souboru libovolného formátu. Některé formáty uložení scény mají uložené objekty a k nim uložený graf scény. Proto je v tomto bloku nutné objekty předzpracovat, vypočítat pozice trojúhelníku ve světových souřadnicích a získat jedno pole trojúhelníků. Tento blok je velmi rychlý, časově je oproti ostatním zanedbatelný. Rychlost tohoto bloku bývá určena způsobem uložení dat v souboru (xml, plain text, binary data), případně rychlostí načítání dat z disku. Předzpracování je možno provádět paralelně.

Vytvoření prostorového indexu nad scénou

Protože scéna bude zobrazována pomocí raytracingu, je vhodné vytvořit prostorový index pomocí kterého se bude rychleji určovat průsečík paprsku s trojúhelníkem. Existuje mnoho prostorových indexů, které je možné pro tento typ úlohy využít (kd tree, octree, bsp tree). Jeden z nejlepších prostorových indexů pro tuto úlohu je kd strom. Existuje mnoho metod jak vytvořit kd strom nad trojúhelníky scény. Nejlepší časovou složitostí vytvoření kd-stromu a jeho možnou paralelizací na GPU je možno dosáhnout pomocí nerekurzivní metody představené ve článku *Real-time KD-tree construction on graphics hardware* [4], kdy se pro tvorbu stromu používá takzvaná Surface area heuristika, která určuje kudy povede rozdělovací rovina. Jak již bylo zmíněno, tento blok lze paralelizovat a jeho rychlost je závislá na počtu trojúhelníků ve scéně, ale oproti ostatním blokům je i tento blok časově zanedbatelný.

Průsečík paprsku se scénou

Existuje mnoho metod jak vypočítat průsečík paprsku s trojúhelníkem. Jedna z nejrychlejších metod je představena v článku *Yet Faster Ray-Triangle Intersection* [5]. Tím, že se nad scénou vytvořil prostorový index, je nutné umět v něm rychle procházet. Pro toto procházení existuje velice vhodné rozšíření čistého prostorového indexu a tím jsou provazy (ropes). Provazy propojují sousední listy v kd stromu a umožňují nerekurzivní a tím pádem velmi rychlý průchod pouze těmi podprostory, kudy prochází vyšetřovaný paprsek. Více o provazech a procházení stromu je možné se dočíst v článku *Stackless KD-Tree Traversal for High Performance GPU Ray Tracing* [6].

² Obrázek je možné stáhnout v plném rozlišení na adrese <http://www.lyso.cz//dp/caustic.jpg>

Vytvoření fotonové mapy

V tomto bloku je nutné vhodně navzorkovat směr šíření světla ze světelných zdrojů (například pomocí rejection samplingu [3]) a pro vyslané fotony určit průsečík se scénou, zaznamenat si úbytek světelného toku, snížit sílu fotonu a odražený foton znovu prošetřit ve scéně na případné průsečíky s objekty scény. Pro urychlení tohoto kroku je možné využít principu takzvané ruské rulety[3]. Rychlost tohoto bloku je silně závislá na počtu vyslaných fotonů, na počtu trojúhelníků a na komplexnosti scény. Při použití výše uvedených akceleračních metod (provazy, kd strom) se povedlo vyslat cca 28 000 fotonů za sekundu (nepoužívá se žádného paralelismu) při jednoduché scéně.

Raytracing

Pro zobrazení scény se používá raytracing. Rychlost raytracingu je závislá na rychlém určení průsečíku paprsku s trojúhelníkem. U photon mappingu je rychlost raytracingu závislá také na počtu vyhledávaných fotonů ve fotonové mapě. Pokud se vypustí vyhledávání fotonů a použijí se akcelerační techniky uvedené výše, tak je možné docílit vypočítání cca 135 tisíc pixelů za sekundu. Tento výpočet se dá urychlit paralelizací, ale kvůli zpomalení vyhledáním nejbližších fotonu (níže) je to poměrně zbytečné.

Vyhledání fotonů

Pro určení osvětlení v bodech průsečíku paprsku se scénou se k lokálnímu osvětlovacímu modelu přičítá také hodnota nepřímého osvětlení z hodnot nejbližších fotonů uložených ve fotonové mapě. Je velmi vhodné použít vyhledávací strukturu, pomocí které se vyhledávání urychlí - například kd strom. Navíc není důležité naprosto přesně určit nejbližší fotony, postačí nejbližší fotony s chybou v řádech jednotek procent. Tomuto vyhledání se říká Approximate nearest neighbor a při použití například knihovny FLANN - Fast Library for Approximate Nearest Neighbors, která tuto metodu implementuje – je možné docílit až třetinového urychlení. I přes toto urychlení, vyhledávání fotonů způsobuje enormní zpomalení výpočtu výsledného obrazu. Při zapnutém vyhledávání fotonů je možné vykreslit pouze, v průměru, 1 000 pixelů za sekundu, což je podstatné zpomalení oproti čistému raytracingu a právě tento blok představuje největší zátěž při výpočtu.

4. ZÁVĚR

V tomto článku byla popsána metoda Photon mappingu. Pomocí této metody je možné docílit fotorealistických efektů jako například nepřímé osvětlení nebo zobrazení kaustik. Hlavním problémem této metody je poměrně dlouhá doba výpočtu nejbližších fotonů, která velmi zpomaluje výpočet výsledného obrazu.

REFERENCE

- [1] Jensen, H. W.: *Global illumination using photon maps*. In Proceedings of the eurographics workshop on Rendering techniques '96
- [2] Kajiya, J. T.: *The rendering equation*. SIGGRAPH Comput. Graph., ročník 20, č. 4, Srpen 1986: s. 143-150, ISSN 0097-8930, doi:10.1145/15886.15902.
- [3] Henrik Wann Jensen. 2001. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., Natick, MA, USA.
- [4] Kun Zhou, Qiming Hou, Rui Wang, and Baining Guo. 2008. *Real-time KD-tree construction on graphics hardware*. *ACM Trans. Graph.* 27, 5, Article 126 (December 2008), 11 pages.
- [5] Jiri Havel and Adam Herout. 2010. *Yet Faster Ray-Triangle Intersection (Using SSE4)*. *IEEE Transactions on Visualization and Computer Graphics* 16, 3 (May 2010)
- [6] Stefan Popov and Johannes Günther and Hans-Peter Seidel and Philipp Slusallek. *Stackless KD-Tree Traversal for High Performance GPU Ray Tracing*. The Eurographics Association and Blackwell Publishing Ltd. p. 415--424 2007