

SHADOW RENDERING THROUGH TRANSLUCENT MATERIALS

Roman Mátl

Bachelor Degree Programme (3), FIT BUT

E-mail: xmatlr00@stud.fit.vutbr.cz

Supervised by: Jan Navrátil

E-mail: inavratil@fit.vutbr.cz

Abstract: This paper describes an efficient and practical method called Colored Stochastic Shadow Maps and its practical implementation in the OpenGL library. This method is used for rendering the correct shadows in the presence of colored translucent materials. Colored Stochastic Shadow Maps has comparable performance profile as traditional shadow maps algorithm.

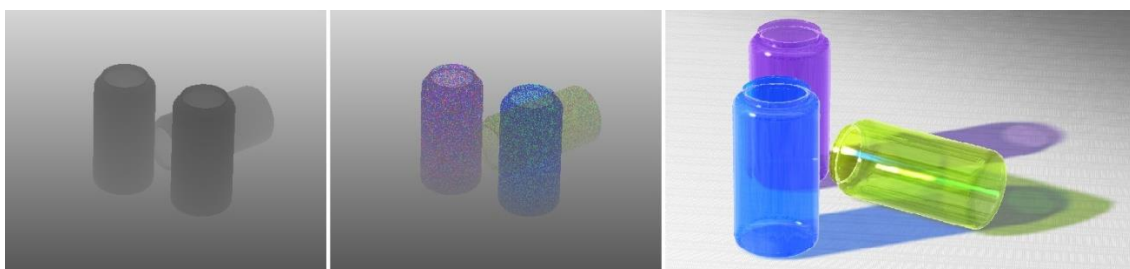
Keywords: CSSM, shadow maps, shadows, OpenGL

1. ÚVOD

V oblasti počítačové grafiky jsou stále více kladeny požadavky na co nejreálnější zobrazení. K tomuto patří i autentické vykreslování stínů, jejichž výsledná barva může být ovlivněna nejen barvou objektu, který jej vrhá, ale i jeho dalšími fyzikálními vlastnostmi. Zobrazení takovýchto stínů v reálném čase bývá často relativně hodně výpočetně i prostorově náročné. Jedním z řešení tohoto problému je algoritmus Colored stochastic shadow maps.

2. COLORED STOCHASTIC SHADOW MAPS

Colored Stochastic Shadow Maps (CSSM) je datová struktura, ale i metoda představena v [1]. Její název je odvozen jak z toho, že produkuje barevné stíny, tak pro svůj vzhled po zobrazení. Metoda rozšířením algoritmu Stochastic transparency [2]. A vychází z tradičního algoritmu stínových map [3], díky tomu je CSSM algoritmus poměrně rychlý na výpočet a lze na něj aplikovat většina rozšíření a antialiasingových úprav aplikovatelných na stínové mapy. Nevýhodou je, že se metoda potýká se stejnými druhy artefaktů.



Obrázek 1: Popis obrázku zleva: klasická stínová mapa, Colored Stochastic Shadow Map a zobrazená scéna (obrázky z [3]).

2.1. OBECNÝ ALGORITMUS (CSSM1)

Algoritmus se skládá ze dvou částí. V prvním kroku proběhne tvorba Colored Stochastic ShadowMap, která je složena z více hloubkových textur. Pro každou vlnovou délku světla jedna. Například pro RGB barevný model budeme potřebovat tři textury. Scénu zobrazíme z pohledu světla. Porovnáme náhodné číslo v rovnoměrném rozložení pravděpodobnosti ξ s konstantní hodnotou

ρ_λ , kterou vypočítáme na základě vlastností materiálu zobrazovaného objektu viz. [Tabulka 1]. Pokud je hodnota ξ menší než hodnota ρ_λ , tak do textury reprezentující stínovou mapu pro danou vlnovou délku zapisujeme hloubku fragmentu. V opačném případě do textury zapisujeme hodnotu reprezentující danou vlnovou délku (v našem případě hodnotu vyhodnocovaného barevného kanálu v intervalu [0,1]). Tento krok musíme zopakovat pro každou vlnovou délku λ .

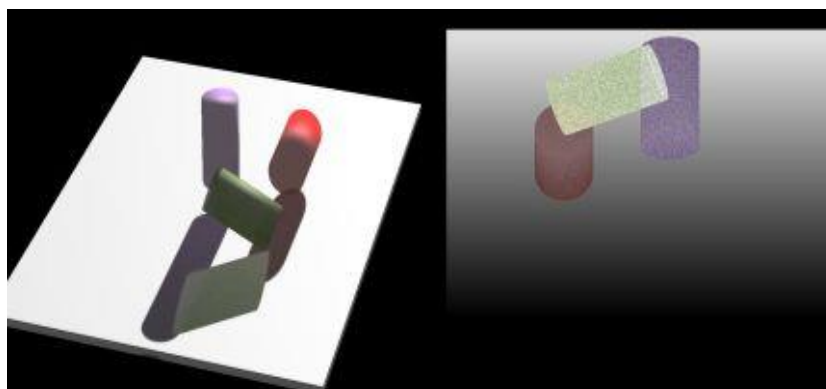
V dalším kroku zobrazujeme scénu z pohledu kamery. Souřadnice zpracovávaného fragmentu transformujeme do souřadnicového systému zdroje světla. Takto transformovanými souřadnicemi přistupujeme do jednotlivých stínových map, které jsme vytvořili v předešlém kroku. Získanou hodnotu porovnáme s hloubkou fragmentu a na základě výsledku porovnání upravíme barevný kanál fragmentu. Tento postup aplikujeme na stínové mapy pro všechny vlnové délky světla. Porovnání bychom měli provádět v rámci nějaké vzorkovací metody (rekonstrukční filtr). Celý tento algoritmus musí být implementován v rámci některé z metod pro zobrazování průhledných povrchů (například depth peeling).

Materiál	α	\vec{t}_r	\vec{t}_g	\vec{t}_b
Zelené sklo	1,00	0,1	0,9	0,1
Průsvitný nylon	0,25	0,5	0,5	0,5
Cihla	1,00	0,0	0,0	0,0
Černý nylon	0,25	0,0	0,0	0,0

Tabulka 1: Příklady vlastností materiálů pro výpočet \vec{p}_λ . Pravděpodobnosti fotonů zasáhnout povrch materiálu (α) a pravděpodobnosti jednotlivých vlnových délek světla na prosvícení materiálu (\vec{t}_λ) podle [1].

3. DEMONSTRAČNÍ APLIKACE

Vytvořil jsem jednoduchou aplikaci za účelem vyzkoušení algoritmu CSSM v praxi. Při implementaci vlastního algoritmu jsem narazil na několik problémů. Nezfetelnějším problémem byly velké barevné odlišnosti ve výsledném stínu, které jsou způsobeny stochastickým šumem. Ten je způsoben stochastickým vzorkováním při tvorbě CSSM. Tento jev jsem zredukoval zvýšením rozlišení CSSM a dále jsem experimentoval s různými rekonstrukčními filtry, jako například Poisson sampling [5], který podává poměrně hladké výsledky, ale jsou patrné “odstínové shluky” viz [Obrázek 2b] nebo Percentage closer filtering (PCF) [4] s různými počty vzorků, kde zobrazování probíhá rychle, ale výsledky nejsou dobré při rozumném počtu vzorků [Obrázek 2a]. Nejlepší



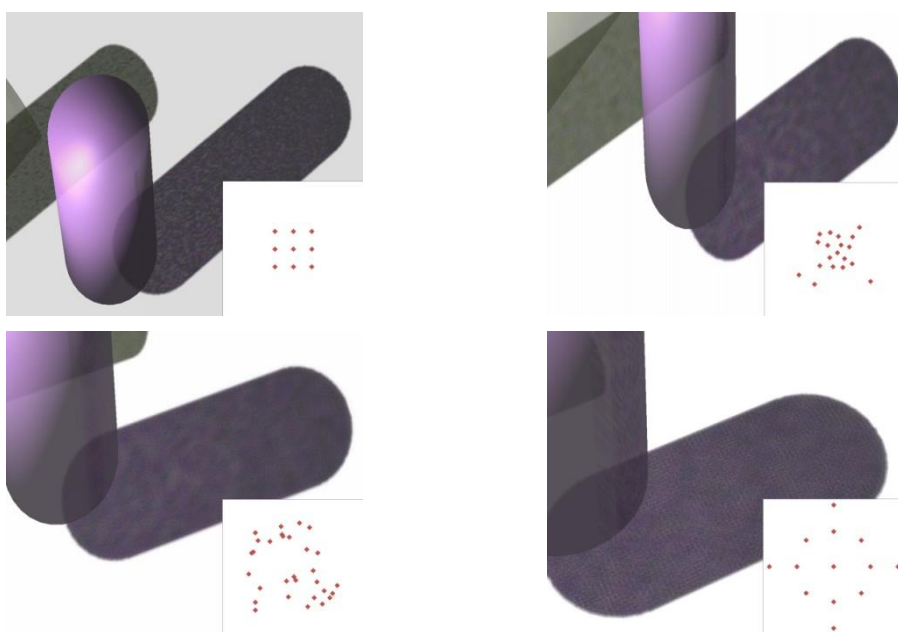
Obrázek 2: Snímek obrazovky demonstrační aplikace, nalevo je zobrazovaná scéna a napravo CSSM pro danou scénu

výsledky podával filtr ve tvaru čtverce kombinovaného s křížem s mikro-offsetem každého bodu podle [1], již při nízkém počtu vzorků [Obrázek 2d].

Dále bylo potřeba vyřešit problémy, které jsou spojené se stínovými mapami. Jako například shadow acne, který řeším použitím biasu při tvorbě stínů, Peter Panning a aliasing na hranách stínů.

4. ZÁVĚR

Metoda CSSM má velký potenciál v typických real-time systémech, jako například ve hrách díky její nízké časové a prostorové náročnosti. Dále jí lze pohodlně kombinovat s dalšími metodami pro zdokonalení vzhledu zobrazované scény. Otestoval jsem více druhů rekonstrukčních filtrů a způsobů generování náhodných čísel pro sběr stochastických vzorků a došel jsem k závěru, že jsou velice podstatné pro rychlost zobrazování a celkový vzhled zobrazené scény. Výstupy mé aplikace se liší od ukázkových obrázků autorů algoritmu, jelikož implementovali CSSM v rámci dalších metod jako je Ray tracing pro realističtější zobrazení lesklých povrchů. Do budoucna lze metodu vylepšit zdokonalením rekonstrukčního filtru.



Obrázek 3: Výsledné stíny při aplikaci různých rekonstrukčních filtrů **a)** PCF 9 vzorků (vlevo nahoře), **b)** poisson disc 20 vzorků (vpravo nahoře), **c)** rovnoměrné rozložení 30 vzorků (vlevo dole), **d)** upravený rekonstrukční filtr podle [1] 13 vzorků (vpravo dole).

REFERENCE

- [1] *McGuire, M., and Enderton, E.* Colored Stochastic Shadow Maps. In Proceeding I3D '11 Symposium on Interactive 3D Graphics and Games, 2011.
- [2] *Enderton, E., Sintorn, E., Shirley, P., and Luebke, D.* Stochastic transparency. In I3D '10: Proceedings of the 2010 symposium on Interactive 3D graphics and games, 2010.
- [3] *Williams, L.* Casting curved shadows on curved surfaces. In SIGGRAPH 1978.
- [4] *Isidoro, John R.* Shadow Mapping: GPU-based Tips and Techniques. Dostupné z: <http://developer.amd.com/wordpress/media/2012/10/Isidoro-ShadowMapping.pdf>,
- [5] Tutorials for OpenGL 3.3 and later: Shadow mapping. [online]. [cit. 2013-02-28]. Dostupné z: <http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/>