



Počítačová grafika

# Radiozita

9. prosince 2017

Autoři: David Kolečkář,  
Matúš Motlík,

[xkolec07@stud.fit.vutbr.cz](mailto:xkolec07@stud.fit.vutbr.cz)  
[xmotli02@stud.fit.vutbr.cz](mailto:xmotli02@stud.fit.vutbr.cz)

Fakulta Informačních Technologii  
Vysoké Učení Technické v Brně

# Obsah

<b>Zadání</b>	<b>2</b>
<b>Zvláštní použité znalosti</b>	<b>3</b>
Tvorba scény . . . . .	3
Konfigurační faktory . . . . .	3
Radiozitní rovnice . . . . .	4
<b>Scéna</b>	<b>5</b>
<b>Práce na projektu</b>	<b>6</b>
Rozdělení práce v týmu . . . . .	6
Co bylo nejpracnější . . . . .	6
Zkušenosti získané řešením projektu . . . . .	6
<b>Autoevaluace</b>	<b>7</b>
<b>Ovládání vytvořeného programu</b>	<b>8</b>
Technologie potřebné pro spuštění programu . . . . .	8
Struktura adresáře . . . . .	8
Obsluha programu . . . . .	8
<b>Doporučení pro budoucí zadávání projektů</b>	<b>9</b>
<b>Literatura</b>	<b>9</b>

# Zadání

Cílem bylo implementovat algoritmus globální osvětlovací metody radiozity nad jednoduchou scénou vhodnou pro demonstraci tohoto algoritmu (přiměřeně složitý prostor, plošné světelné zdroje a různé materiálové vlastnosti). A nakonec zobrazit výslednou scénu pomocí knihovny OpenGL. Radiozita je časově náročná zobrazovací metoda, která se využívá ve fotorealistické grafice. Metoda šíří světelnou energii napříč celou scénou, používá se tak k renderování 3D scén.

# Zvláštní použité znalosti

## Tvorba scény

Základem projektu bylo vytvořit scénu, na které budeme algoritmus radiozity testovat. Scénu jsme vytvářeli v 3D programu Blender. V tomto programu jsme také provedli dělení velkých ploch na menší plošky v našem případě trojúhelníky. Scénu jsme jen následně vyexportovali ve formátu *.obj*, který obsahoval vrcholy a indexy trojúhelníků. Při tvorbě scény bylo důležité, aby scéna byla energeticky uzavřená a povrchy objektů byly neprůhledné. Následně jsme v našem programu naprogramovali metodu pro načítání tohoto souboru. Při jeho čtení jsme jednotlivým objektům nastavovali různé materiálové vlastnosti [2].

- Emission barva jakou ploška vyzařuje. Hodnoty jsou omezeny od 0 do 1, přičemž 0 znamená, že povrch veškerou energii pohlcuje. V opačném případě ji odráží.
- Reflektivity barva kterou ploška odráží. Hodnoty jsou od 0 do 1.

## Konfigurační faktory

Před výpočtem radiozity je důležité spočítat tzv. konfigurační faktory [1] (form factors), což je vliv jedné plošky na každou jinou plošku. K výpočtu konfiguračního faktoru je nutné zobrazit scénu z pohledu cílové plošky s ploškou vyzařující světlo. A to pro každou dvojici plošek. Při tomto vykreslování stačí vidět 180° okolí z pohledu plošky. K tomu se využívá vykreslování do polokrychle (hemicube). Jednotlivé trojúhelníky mají ve scéně odlišnou barvu, protože je nutné jednoznačně identifikovat plošky, které jsou osvětlené ze středu polokrychle. Tudíž se index plošky zakóduje jako jeho barva a následně se pomocí Z-buffer algoritmu vypočítá, či je daná ploška viditelná ze středu polokrychle.

Celkový konfigurační faktor (2) plošky je roven součtu tzv. delta faktorů (1) plošek na polokrychle. Uvedený vzorec pro výpočet delta faktoru je orientován pro horní stěnu polokrychle. Pro boční stěny je výpočet mírně odlišný.

$$w_z(x, y) = \frac{(\cos\phi_i)^4}{\pi} = \frac{1}{\pi(x^2 + y^2 + 1)^2} \quad (1)$$

$$\sum_{X=-P/2}^{P/2-1} \sum_{Y=-P/2}^{P/2-1} H_{ij}(X, Y) \cdot w_z(X, Y) \frac{1}{P^2} \quad (2)$$

# Radiozitní rovnice

Radiozitní rovnice pro každou plošku je následující:

$$B_i = E_i + R_i \sum_{j=1}^n B_j F_{ij} \quad (3)$$

- B je radiozita dané plošky
- E je vyzařovaná energie
- R je odrazivost materiálu
- F je konfigurační faktor

---

**Algoritmus 1:** Algoritmus počítající konfigurační faktory s využitím Z-bufferu

---

```
1:  for i = 1 to N for j = 1 to N do  $F_{ij} = 0$ 
2:  for i = 1 to N do
3:      camera = center of  $\Delta A_i$ ;
4:      for k = 1 to 5 do
5:          window = kth face of the hemicube;
6:          for x = 0 to P-1 do
7:              for y = 0 to P-1 do pixel[x,y]=0;
8:              Z-BUFFER ALGORITHM (color of surface j is j)
9:              for x = 0 to P - 1 do for y = 0 to P - 1 do
10:                 if(pixel[x,y] < 0) then
11:                      $F_{i,pixel[x,y]} += w_k(x - P/2, y - P/2)/P^2$ ;
12:                 endifor
13:             endfor
14:         endfor
15:     endfor
```

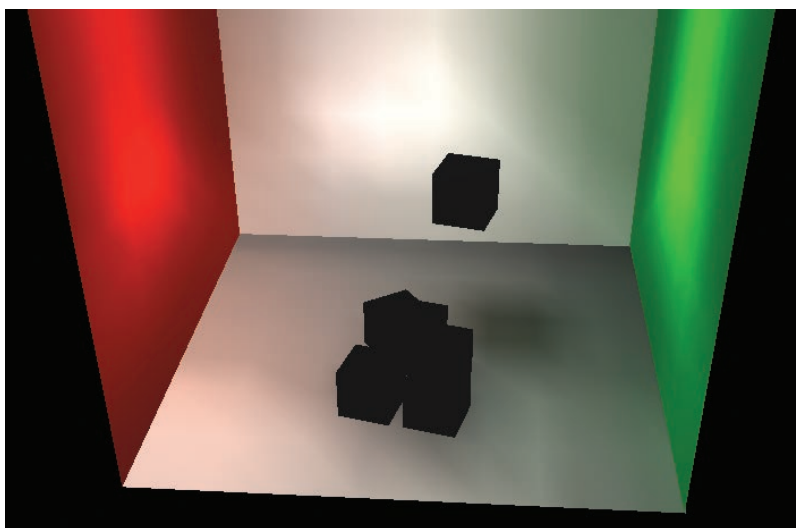
---



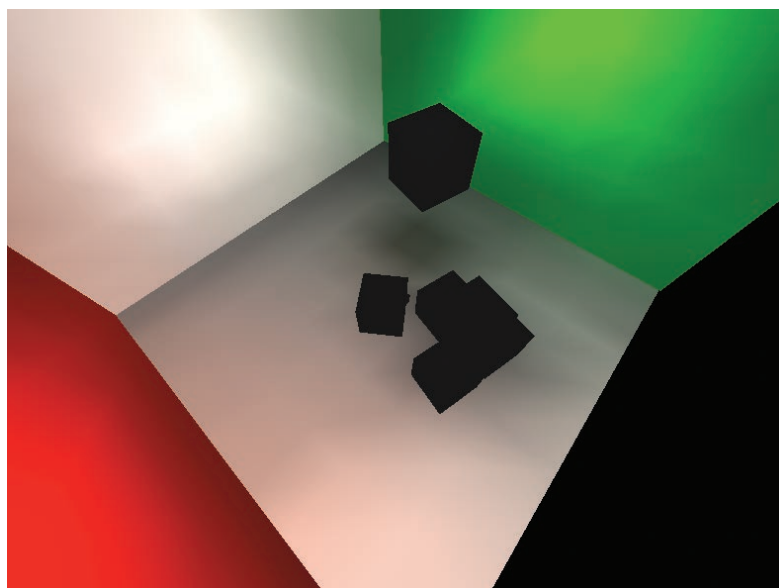
Obrázek 1: Ukázka vykreslené polokrychle.

# Scéna

Zde je zobrazena výsledná scéna, která obsahuje 1092 trojúhelníků (plošek).



Obrázek 2: První ukázka vykreslené scény.



Obrázek 3: Druhá ukázka vykreslené scény.

# Práce na projektu

## Rozdělení práce v týmu

- **David Kolečkář:** Vytvořena struktura programu, vytvoření scény, tvorba třídy reprezentující plošku (Triangle), dokumentace.
- **Matúš Motlík:** Výpočet delta faktoru, vykreslení polokrychle, výpočet konfiguračních faktorů a vykreslení scény.

## Co bylo nejpracnější

Největším problémem projektu, bylo nemožnost si výsledky jednotlivých výpočtů nějak ověřit či porovnat. Při vzniklé chybě jsme tak nevěděli, zda-li je některý z výpočtů špatný či nikoliv. A celkové debugování projektu nám dělalo velký problém.

## Zkušenosti získané řešením projektu

Jelikož jsme měli praktické zkušenosti s grafikou téměř nulové, byl pro nás projekt velkým přínosem. Rozšířili jsme si znalosti s programováním na GPU. Například při renderování samotné scény či polokrychle do textury. Navíc jsme si vyzkoušeli prostředí 3D programu Blender, což je profesionální nástroj na tvorbu 3D grafiky. Co se týká teoretických znalostí, tak jsme více do hloubky pochopili princip osvětlení scény a seznámili jsme se s různými metodami radiozity.

# Autoevaluace

V následujících odrážkách je uvedeno hodnocení jednotlivých částí projektu.

**Technický návrh (75%):** Provedli jsme vhodné rozdělení projektu na jednotlivé třídy. Program byl primárně vyvíjen pod systémem Linux, avšak mělo by být možné projekt spustit i v systému Windows (například za pomoci Visual Studia).

**Programování (75%):** V kódu není mnoho komentářů, avšak vhodně pojmenované metody a proměnné přispívají k čitelnosti kódu.

**Vzhled vytvořeného řešení (75%):** Samotný výpočet radiozity, běží obstojně. Ovšem perfektního výsledku jsme nedosáhli z důvodu zřejmě špatné interpolace mezi jednotlivými ploškami (trojúhelníky).

**Využití zdrojů (80%):** Z literatury jsme čerpali především vzorce pro výpočet delta faktoru a konfiguračních faktorů. Dále nám usnadnil práci pseudokód uvedený v [3].

**Hospodaření s časem (40%):** Projektem jsme se začali zabývat v celku brzy, avšak jen z teoretické části. Samotná implementace začala probíhat až 14 dní před odevzdáním, což se ukázalo jako velmi problematické.

**Spolupráce v týmu (100%):** Komunikace probíhala bez problému, většinou jsme preferovali osobní schůzky. Kód jsme sdíleli za pomoci verzovacího systému.

**Celkový dojem (50%):** Z počátku zadání nevypadalo tak časově náročné jak nakonec bylo, což se projevilo na kvalitě výsledného řešení. Největším problémem bylo debugování programu. Nevěděli jsme, jak si ověřit výsledky jednotlivých výpočtů.



# Ovládání vytvořeného programu

## Technologie potřebné pro spuštění programu

V následujícím seznamu jsou uvedeny knihovny, které jsou nutné pro správný běh programu.

**OpenGL:** Vykreslování scény a polokrychle (Hemicube) probíhá za pomoci této knihovny.

**SDL:** Tato knihovna zajišťuje práci s okny včetně vstupu z klávesnice a myši.

**PGR soubor:** V projektu jsme také využili soubory `pgr.c` a `pgr.h`, které jsme používali na počítačových cvičení. Cílem těchto souborů je přiložit všechny potřebné knihovny (GLEW, GLee). Zároveň zapouzdřuje práci s knihovnou SDL.

**GLM:** Je matematická knihovna, zajišťující práci s maticemi a vektory.

## Struktura adresáře

Zde je uvedena struktura odevzdaného adresáře.

- `src/` - obsahuje zdrojové kódy programu
- `doc/` - tato dokumentace ve formátu `.pdf` a `LATEX`
- `scene/` - vykreslovaná scéna ve formátu `.obj`
- `image/` - obrázky vykreslené scény pomocí radiozity

## Obsluha programu

Překlad programu probíhá pomocí *Makefile*, který je umístěn v adresáři se zdrojovými kódy. V tomto adresáři stačí zavolat příkaz *make*, který zajistí překlad zdrojových souborů. Pro spuštění programu je zapotřebí zavolat příkaz *make run*. Po spuštění programu je v okně programu zobrazena kostra scény a v příkazovém řádku vypsán počet trojúhelníků dané scény. Pro spuštění algoritmu radiozity je zapotřebí, na klávesnici stisknout klávesu *r*. Pomocí myši je scénu možné různě přibližovat či rotovat. Klávesa *s* umožňuje přepínat mezi scénou s radiozitou či bez ní. Lze také přepínat mezi obrysy trojúhelníků a vyplněnými trojúhelníky a to za pomoci klávesy *l* respektive *f*. Program lze ukončit klávesou Escape.

# Doporučení pro budoucí zadávání projektů

Projekt nakonec vyžadoval velkou časovou zátěž, ale ta plně odpovídá bodovému hodnocení. Zadání byla ponechána velká volnost, což v našem případě bylo určitě pozitivní.

## Literatura

- [1] David Šabata. Radiosita na GPU, 2011. Bakalářská práce. Vysoké Učení Technické v Brně, Fakulta Informačních Technologií.
- [2] Jiří Janoušek. Realistické zobrazování pomocí radiozity, 2007. Bakalářská práce. Vysoké Učení Technické v Brně, Fakulta Informačních Technologií.
- [3] Laszlo Szirmay-Kalos. Theory of Three Dimensional Computer Graphics, 1996. Dostupné na <http://sirkan.iit.bme.hu/~szirmay/radiosit.pdf>.