

PGR – Počítačová grafika

# Rasterizace na CPU

13. prosince 2018

Autoři: Kamil Michl,  
David Skácel,

[xmichl02@stud.fit.vutbr.cz](mailto:xmichl02@stud.fit.vutbr.cz)  
[xskace12@stud.fit.vutbr.cz](mailto:xskace12@stud.fit.vutbr.cz)

Fakulta Informačních Technologii  
Vysoké Učení Technické v Brně

# Obsah

<b>Zadání</b>	<b>2</b>
<b>Nejdůležitější dosažené výsledky</b>	<b>3</b>
70 000 trojúhelníků v reálném čase i s osvětlením (optimalizace) . . . . .	3
Scéna s texturami . . . . .	4
Nasvětlení pomocí Phongova modelu . . . . .	5
<b>Zvláštní použité znalosti</b>	<b>6</b>
<b>Práce na projektu</b>	<b>7</b>
Rozdělení práce v týmu . . . . .	7
Co bylo nejpracnější . . . . .	7
Zkušenosti získané řešením projektu . . . . .	7
<b>Autoevaluace</b>	<b>8</b>
<b>Ovládání vytvořeného programu</b>	<b>9</b>
Technologie potřebné pro spuštění programu . . . . .	9
Programy a služby použité při tvorbě . . . . .	9
Obsluha programu . . . . .	9
<b>Literatura</b>	<b>10</b>

# Zadání

Tento projekt řeší rasterizaci na CPU implementací v C++. Vzhledem k volnějším zadání jsme se rozhodli implementovat následující funkcionalitu:

- Zobrazení 3D modelů načtených ze souboru pomocí rasterizace trojúhelníků.
- Optimalizace vykreslování trojúhelníku pomocí 2D obalového tělesa, jednoduchého ořezu scénou a zahazování odvrácených stran.
- Řešení viditelnosti s pomocí jednoduchého hloubkového bufferu a konvence „top-left edge” zahazování hran.
- Texturování modelů obsahujících textury. Jednoduché stínování podle úhlu plochy vůči scéně.
- Zobrazení modelů neobsahujících textury s pomocí Phongova osvětlovacího modelu.
- Možnost volby mezi tvrdým a hladkým gouraudovým stínováním pokud model obsahuje informace o normálách ve vrcholech.
- K implementaci jsme využili následujících knihoven:
  - GLM pro vektorové počty. <sup>1</sup>
  - Assimp pro načtení modelů. <sup>2</sup>
  - stb\_image.h - hlavičková knihovna pro načítání obrazových souborů (textur). <sup>3</sup>
  - SDL pro zobrazení okna a obsluhu klávesnice. <sup>4</sup>

---

<sup>1</sup><https://glm.g-truc.net/0.9.9/index.html>

<sup>2</sup><http://www.assimp.org/>

<sup>3</sup><https://github.com/nothings/stb>

<sup>4</sup><https://www.sdl.com>

# Nejdůležitější dosažené výsledky

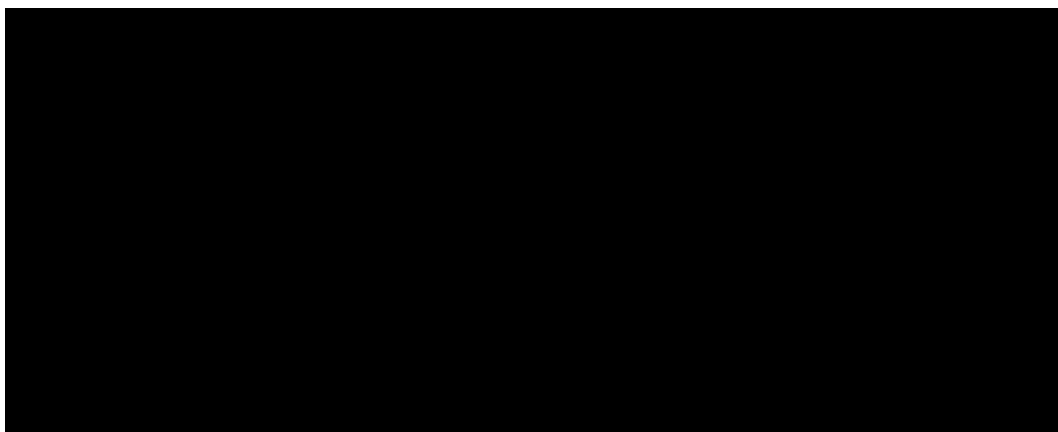
Následující 3 podkapitoly popisují výslednou funkcionalitu kterou považujeme za stěžejní.

## 70 000 trojúhelníků v reálném čase i s osvětlením (optimalizace)

Schopnost vykreslit model o bezmála 70 000 trojúhelnících i s osvětlením v reálném čase patří mezi přední vlastnosti tohoto projektu. Na mobilním procesoru intel i7 7. generace s frekvencí 3,5GHz aplikace běží průměrně s 15 snímků za vteřinu v závislosti na vzdálenosti od vykreslovaného objektu<sup>5</sup>.

O tuto vlastnost se zaslouhují především tyto 3 optimalizace:

- 2D minimální obalové těleso zarovnané s osami X a Y – triangle 2D axis aligned bounding box
- Základní ořezání výhledovým tělesem – naive frustum - culling
- Zahazování odvrácených stran trojúhelníků – backface - culling



Obrázek 1: Stanford bunny. 69630 trojúhelníků, 2 bodové světla (vlevo) a k nim navíc 1 směrové (vpravo).

---

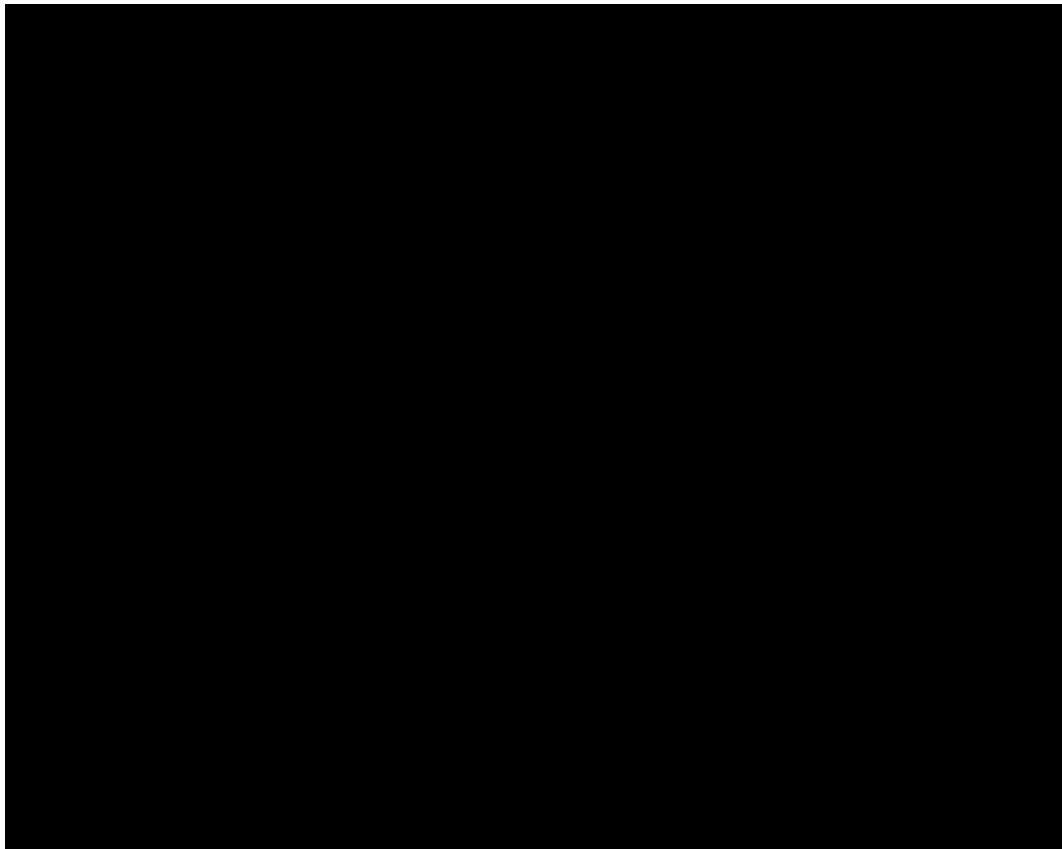
<sup>5</sup>Vzhledem k povaze první optimalizace je zřejmé, že výsledný výkon se odvíjí především od toho, jak velkou část obrazu model zabírá.

Scéna	Trojúhelníků	Osvětlení	Textury	Snímky/sekundu
Sponza	13 472	1 Směrové	Ano	$\sim 4.5$
Stanford Bunny	69 630	2 bodové, směrové	Ne	$\sim 15$
Suzanne	15 488	2 bodové, směrové	Ne	$\sim 15$

Tabulka 1: Vzorová tabulka

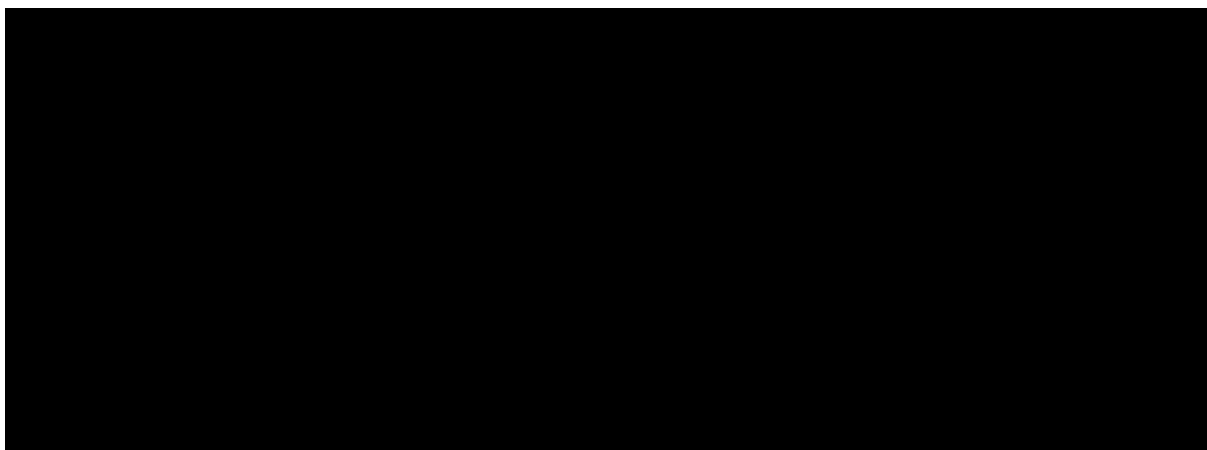
## Scéna s texturami

Rasterizér umožňuje načtení modelů s texturami. Je možno přeložit aplikaci se zapnutým opakováním textur s pomocí makra `#define TEXTURE_REPEAT`. Vykreslování scén s texturami je o něco rychlejší díky absenci bodových světél a Phongova osvětlovacího modelu. Použito je pouze základní stínování v závislosti na úhlu, který plocha trojúhelníku svírá s vektorem rovnoběžným s osou Z v záporném směru, kde souřadnice X a Y odpovídají bodu na trojúhelníku. Tedy  $\vec{V} = \text{vec3}(\text{Fragment}_x, \text{Fragment}_y, -1.0)$ . Je-li pak normálový vektor v daném bodě  $N$ , pro výsledný výpočet barvy platí vztah  $\text{Color} = \text{Color}_{\text{texture}} * \text{dot}(\vec{V}, \vec{N})$ , kde  $\text{dot}()$  je skalární součin vektorů odpovídající cosinu úhlů jimi svíraným.



Obrázek 2: Sponza - 13472 trojúhelníků s texturami.

## Nasvětlení pomocí Phongova modelu



Obrázek 3: Blender association Suzanne - 15 488 trojúhelníků.

# Zvláštní použité znalosti

Uvedte informace, které byly potřeba nad rámec výuky probírané na FIT. Vysvětlete je pomocí obrázků, schémat, vzorců apod.

**Rozsah:** podle potřeby a vlastního uvážení

# Práce na projektu

## Rozdělení práce v týmu

Pokud to bude vhodné, použijte odrážky místo souvislých vět.

**Rozsah:** co nejstručněji tak, aby bylo zřejmé, jak byla dělena práce a za co v projektu je kdo zodpovědný.

- **Kamil Michl:** Vykreslovací optimalizace, osvětlovací modely, texturování.
- **David Skácel:** Optimalizace kódu, refaktORIZACE a návrh struktury kódu.

## Co bylo nejpracnější

potýkat, co zabralo čas.

**Rozsah:** 5-10 řádků

- **Kamil Michl:** Pochopení rasterizační pipeline, optimalizace kódu jako takového.
- **David Skácel:** Implementace konvencí optimalizací a osvětlovacího modelu.

## Zkušenosti získané řešením projektu



# Autoevaluace

**Technický návrh (50%):** (analýza, dekompozice problému, volba vhodných prostředků, ...)

**Programování (50%):** (kvalita a čitelnost kódu, spolehlivost běhu, obecnost řešení, znovupoužitelnost, ...)

**Vzhled vytvořeného řešení (50%):** (uvěřitelnost zobrazení, estetická kvalita, vzhled GUI, ...)

**Využití zdrojů (50%):** (využití existujícího kódu a dat, využití literatury, ...)

**Hospodaření s časem (50%):** (rovnoměrné dotažení částí projektu, míra spěchu, chybějící části řešení, ...)

**Spolupráce v týmu (50%):** (komunikace, dodržování dohod, vzájemné spolehnutí, rovnost, ...)

**Celkový dojem (50%):** (pracnost, získané dovednosti, užitečnost, volba zadání, cokoliv, ...)

# Ovládání vytvořeného programu

## Technologie potřebné pro spuštění programu

Následující knihovny:

**glm,sdl,stb\_image.h, assimp** Matematická knihovna GLM, SDL knihovna pro vytvoření okna pro vykreslování, **STB\_image.h** pro načtení obrázků z souboru, **knihovna Assimp** pro načtení 3D modelu.

## Programy a služby použité při tvorbě

Zde vypište, jaké technologie vaše řešení používá – co potřebuje k běhu, co jste použili při tvorbě, atd. Text strukturujte, použijte odrážky, číslování. . .

**Rozsah:** odrážky, dle potřeby

## Obsluha programu

Stručně popište, jak se program ovládá (nejlépe odrážky rozdělené do kategorií). Pokud se ovládání odchyluje od zkratk a způsobů obvykle používaných v okýnkových nadstavbách operačních systémů, zdůvodněte, proč se tak děje.

# Literatura