

# MEDICAL DATA RENDERING IN REAL-TIME

**Kristián Lengyel**

Master Degree Programme(2), FIT BUT  
E-mail: xlengy00@stud.fit.vutbr.cz

Supervised by: Radovan Jošth

E-mail: ijosth@fit.vutbr.cz

## ABSTRACT

This paper shows the basic steps for rendering medical volume data in real-time. The main objective of this work is to accelerate rendering on today's graphics cards by using programmable hardware, like vertex and pixel shaders for imaging large three-dimensional volume data. By using the latest technology, CUDA, we can also parallelize time-consuming operations on graphics hardware. Similar applications are used outside of medicine and other fields, such as chemistry and biology for displaying molecular structures and microorganisms. The rendering technique used in this work is raycasting, which provides us the most suitable resulting image.

## 1. ÚVOD

Súčasná medicína používa viacero technológií na snímanie ľudského organizmu, pričom všetky sú založené na určitých fyzikálnych či chemických procesoch prebiehajúcich v tele človeka. Medzi najpoužívanejšie snímacie metódy patria CT (Computed Tomography), MRI (Magnetic Resonance Imaging) a PET (Positron Emission Tomography). Všetky tieto zariadenia snímajú objekt z rôznych uhlov a potom zložitými algoritmi rekonštruujú výsledný obraz. Vo výsledku dostávame sériu 2D snímok (*rezov*) s rôznymi úrovňami hodnoty šedej, podľa hustoty materiálu, ktoré spolu tvoria 3D maticu skalárnych hodnôt. Túto maticu sme schopný reprezentovať 3D textúrou na grafickej karte a zobrazovať pomocou známych renderovacích techník, akou je napr. vrhanie lúča (*raycasting*), ktorý použijeme aj v tejto práci. Aby sme dosiahli čo najrealistickejší výstupný obraz, musíme tieto dáta určitým spôsobom predspracovať ešte pred vykreslením. Tieto operácie bývajú časovo náročné na výpočet, a to najmä pri veľkých objemových dátach. Preto ich bude treba hardwarovo urýchliť, k čomu nám poslúži najnovšia technológia CUDA, vyvinutú firmou NVIDIA, ktorá poskytuje efektívnu paralelizáciu úloh na grafickej karte pomocou systému multiprocessorov a vlákien.

## 2. VRHANIE LÚČA

Metóda vrhania lúča funguje na princípe premietania lúča z kamery do objemových dát a projektovaním farby na 2D rovinu obrazu. Tento lúč prechádza objemom, ktorý lineárne vzorkuje s nejakým pevne definovaným krokom. Body v okolí pritom musia byť trilineárne interpolované, o čo sa už postará textúrovacia jednotka grafickej karty. Ako lúč prechá-

dza objemovými dátami, akumuluje do seba hodnoty voxelov a následne počíta výstupnú farbu a alfa zložku pixelu podľa definovaných kompozitných funkcií. Lúč je orezaný na objem dát, aby sa ušetril čas výpočtu. Algoritmus vrhania lúča pozostáva zo štyroch základných krokov – **vrhanie lúča, vzorkovanie 3D textúry, výpočet lokálneho osvetľovacieho modelu, kompozícia výslednej farby**.

Výhodou techniky vrhania lúča je, že poskytuje najkvalitnejší grafický výstup, avšak celý výpočet prebieha na grafickej karte a nie je možné rozdeliť úlohy medzi GPU a CPU, čím sa táto metóda radí medzi najpomalšie. Pri dnešných rýchlych grafických kartách a najmä ich uniformnom shaderovacom systéme však tento problém zaostáva a s novými technológiami vyvíjanými v súčasnosti, ako napr. CUDA, sa dokonca grafické karty stávajú užitočnými pre rýchle paralelné spracovanie časovo náročných operácií.

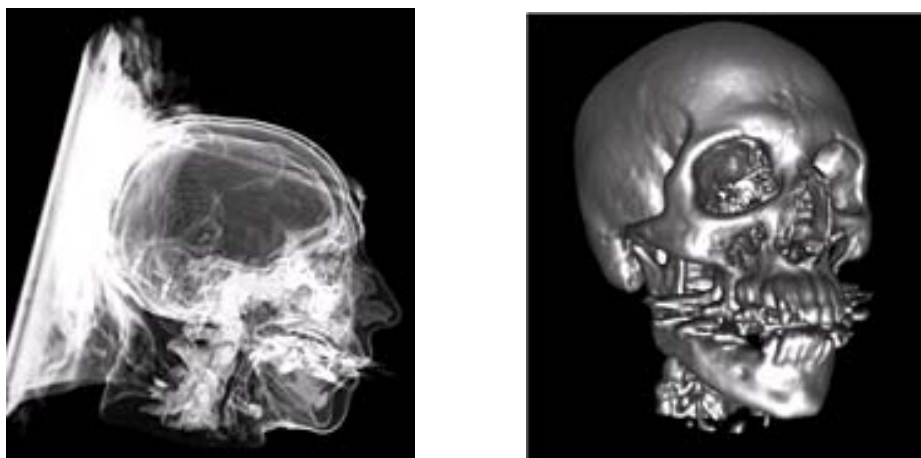
Existuje viacero projekčných metód, z ktorých najvýznamnejšie a najčastejšie používané sú **projekcia maximálnej intenzity** a **kompozitná metóda**. Tieto dve metódy budú implementované aj v našej aplikácii.

## 2.1. PROJEKCIA MAXIMÁLNEJ INTENZITY

Lúč prechádzajúci objemom hľadá maximálnu hodnotu v dátach a automaticky priradí túto hodnotu na korešpondujúci pixel obrazu. V medicínskej praxi sa často pridávajú kontrastné látky do krvného obehu, aby mali cievy a tepny vysokú výslednú hodnotu šedej po nasnímaní. Lúče, ktoré neprejdú oblasťami s vysokou hodnotou šedej, ostanú vo výslednom obraze tmavšie. V konečnom výsledku dostávame semitransparentný obraz objemových dát, kde svetlé oblasti ako cievy „svietia“, kým tmavšie oblasti ako koža sú viditeľné slabšie.

## 2.2. KOMPOZÍCIA

Kompozitná metóda určuje výslednú farbu pixelu na základe zmesi všetkých vzorkovaných hodnôt farby a priehľadnosti voxelu získaných pri prechode lúča objemovými dátami. Pre každú vzorku sa zisťuje farba a priehľadnosť pomocou tzv. **prenosovej funkcie**. Vstupom do tejto funkcie môže byť jednak samotná hodnota voxelu alebo nejaká iná hodnota, napr. vypočítaný gradient daného voxelu. Prenosová funkcia v podstate klasifikuje každý voxel v objeme ako súčasť určitého typu tkaniva. Výhodou je, že vďaka tejto metóde máme možnosť zobrazit' každý typ tkaniva inou farbou a inou hodnotou priehľadnosti. Prenosová funkcia bude vo výslednej aplikácii reprezentovaná 1D textúrou.



**Obrázek 1:** Výstup súčasnej aplikácie - CT sken hlavy zobrazený kompozitnou a MIP metódou s osvetlením

### 3. ARCHITEKTÚRA CUDA

Technológia CUDA je paralelná výpočtová architektúra, ktorá obsahuje tzv. CUDA Instruction Set Architecture (ISA) a paralelný výpočtový stroj v GPU. Na programovanie CUDA sa používa jazyk C, ktorý patrí medzi najznámejšie a najpoužívanejšie vysoko úrovňové programovacie jazyky v súčasnosti, a ktorý poskytuje vysokú rýchlosť programov.

Grafické karty s podporou CUDA obsahujú stovky jadier, ktoré dokážu kolektívne spustiť tisíce vláken (*threadov*). Každé jadro má zdieľané zdroje zahŕňajúce registre a pamäť. Zdieľaná pamäť umožňuje zdieľať dáta pre paralelné úlohy bez potreby posielania týchto dát po systémovej zbernici.

Pre nás bude mať CUDA veľký význam pri paralelizácii časovo náročných operácií, akou je napr. výpočet gradientov pre osvetľovací model, filtrácia týchto gradientov, príp. iné operácie predspracovania vstupných dát.

### 4. ZÁVER

Aplikácia je schopná v súčasnom stave zobrazovať objemové dáta, ktoré boli nasnímané niektorou skenovacou technológiou. Miesto projekcie maximálnej intenzity je v súčasnosti implementovaná projekcia typu *isoplocha*, ktorá sa líši v tom, že sa hľadá špecifická hodnota v dátach a zobrazia sa len voxely s touto hodnotou. Na tejto projekčnej metóde bol testovaný aj výpočet Blinn-Phongovho osvetľovacieho modelu. Na implementáciu algoritmu vrhania lúča sa používajú programovateľné shadere, pričom najdôležitejšiu časť algoritmu implementuje pixel shader, v ktorom sa vykonáva vzorkovanie 3D textúry a počíta sa výsledná farba a priehľadnosť pixelu. V pixel shaderi sa tiež implementuje tzv. per-pixel osvetľovací model, ktorý dáva našim objemovým dátam realistickejší vzhľad. Medzi budúce práce patrí vytvorenie kvalitnej prenosovej funkcie, urýchlenie niektorých operácií pomocou CUDA a vylepšenie užívateľského rozhrania.

PodĎakování: Tato práce vznikla částečně za podpory grantu VUT FIT, FIT-S-10-2 a specifického výzkumu MSM0021630528.

### LITERATÚRA

- [1] Žára, J., Beneš, B., Sochor, J., Felkel, P.: *Moderní počítačová grafika*. Computer Press, Brno, 2005. ISBN 80-251-0454-0
- [2] Triers, P.: GPU raycasting tutorial [online], [cit. 2009-10-15]. Dostupné na URL: <[http://www.daimi.au.dk/~trier/?page\\_id=98](http://www.daimi.au.dk/~trier/?page_id=98)>
- [3] Winter, A.: Volume Dataset Portal [online], [cit. 2009-10-01]. Dostupné na URL: <<http://www.custard.org/~andrew/visualization/datasets/>>
- [4] Volvis: Volumetric datasets [online]. [cit. 2009-10-01]. Dostupné na URL: <<http://volvis.org/>>
- [5] NVIDIA CUDA ZONE: CUDA API [online], [cit. 2009-12-28]. Dostupné na URL: <[http://www.nvidia.com/object/cuda\\_develop.html](http://www.nvidia.com/object/cuda_develop.html)>