Projekt do předmětu **VGE – Výpočetní geometrie**

Akcelerační datové struktury pro ray-tracing

řešitelé: **Bc.** **Ondřej** **Áč,** xacond00
Bc. **Jozef** **Bilko,** xbilko03
Bc. Marek Konečný, xkonec86

Zadání

Cílem zadání bylo implementovat a vyhodnotit některé z akceleračních datových struktur pro ray-tracing. Konkrétně jsme se rozhodli pro následující akcelerační struktury:

- Per mesh Bounding boxes
- Bounding Volume Hiearchy
- k-d tree (+ SAH)
- Bounding Interval Hiearchy se SAH binning (Surface Area Heuristic)

V rámci evaluace jsme nad jednotlivými strukturami vyhodnotili následující metriky:

- Čas vybudování a aktualizace
- Rychlost průchodu paprsky
- Paměťovou režii

Použité technologie

- Git
- C++17
- SDL (github.com/libsdl-org/SDL)
- Imgui (github.com/ocornut/imgui)
- PTCR 2.0 (github.com/Panjaksli/PTCR2.0) (repozitář O. Áče, pouze k inspiraci)

Použité zdroje

Při vývoji akceleračních struktur jsme vycházeli z těchto článků a dalších zdrojů:

- Binned BVH building, jacco.ompf2.com/2022/04/13/how-to-build-a-bvh-part-1-basics/
- Instant Ray Tracing: The Bounding Interval Hierarchy, ainc.de/Research/BIH.pdf
- Implementace BIH v node.js (pro inspiraci a pochopení algoritmu): github.com/imbcmdth/bxh
- Kdtree Introduction to K-D Trees | Baeldung on Computer Science

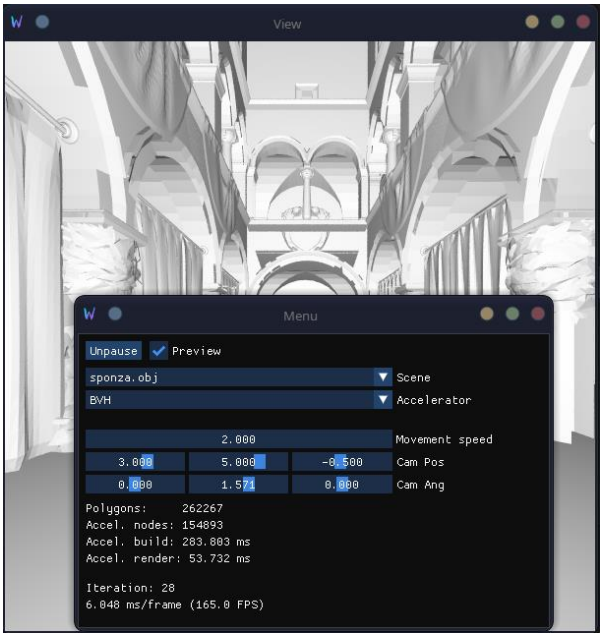
Pro vývoj a evaluaci jsme použili následující vzorové scény, tj. modely ve formátu OBJ:

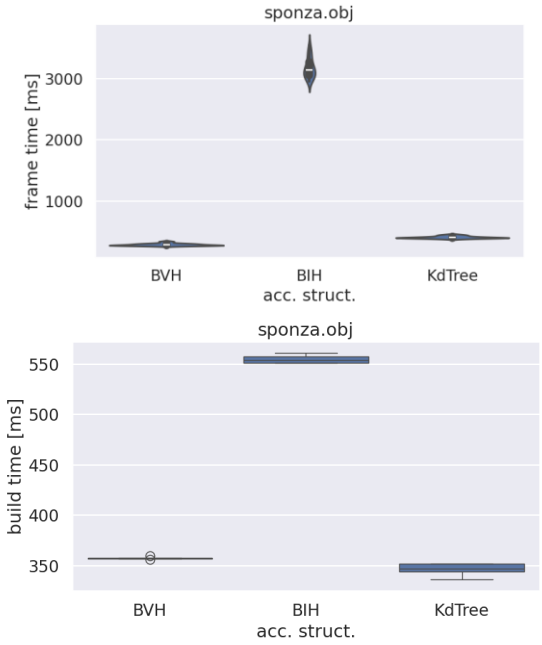
- Standford Armadillo (212574 poly.)
- Standford Bunny (4968 poly.)
- Standford Dragon (249882 poly.)
- Sponza (262267 poly.), model dostupný na: github.com/jimmiebergmann/Sponza

Modely ze Standfordského 3D repozitáře zpracované do formátu OBJ jsme převzali z: github.com/alecjacobson/common-3d-test-models.

Nejdůležitější výstupy

Program poskytuje real-time náhled vykreslovaných scén, pomocí grafického prostředí pak umožňuje načítání různých scén i změnu akcelerační struktury za běhu programu. Vypisuje statistiky o rychlosti vykreslení a času konstrukce struktury. Dále je zobrazena struktura programu a měřené časy běhu.





Ovládání vytvořeného programu

Instrukce k překladu programu jsou uvedeny v odevzdaném README. Spuštění programu: `./vgert`

Po spuštění programu se zobrazí dvě okna:

- View: slouží k náhledu samotného RT vykreslování. Po každém průchodu se náhled aktualizuje.
- Menu: slouží k ovládání náhledu a zobrazení statistik o scéně, akc. struktuře a výkonu.

Načítání scén

Program po spuštění automaticky skenuje adresář `../` a hledá soubory s příponou `.obj`. Pokud jsou v adresáři takové soubory nalezeny, jsou nabídnuty k volbě skrze *combobox* v okně Menu. Po volbě nové scény je tato scéna načtena a akcelerační struktura je přestavěna. Výchozí scéna je zvolena arbitrárně.

Navigace ve scéně

Program umožňuje dva způsoby navigace ve scéně. Zaprvé je v okně Menu možné pomocí posuvníků přímo manipulovat s transformací kamery (pozice i rotace). Zadruhé je možné v okně View měnit pozici kamery pomocí klávesnice (ovládání šipky + WSAD + Space + lCtrl). Pro druhý režim je v okně Menu dostupný posuvník rychlosti pohybu, tj. délka kroku při jednom stisku klávesy.

Ostatní ovládací prvky

Dále je v Menu *combobox* pro výběr akcelerační struktury. Po zvolení nové struktury je automaticky sestavěna na aktuální scéně.

Poslední ovládací prvek v okně Menu je tlačítko Pause/Unpause, jehož účel je pozastavení RT vykreslování v okně View.

Rozdělení práce v týmu

Jednotlivý členové se věnovali následujícím dílčím částem:

- Ondřej Áč:
  - Vedení týmu
  - Struktura ray tracing enginu (lin. algebra, scéna + parser, RT renderer, GUI aj.)
  - Implementace Bounding Volume Hiearchy
  - Implementace Per mesh Bounding Boxes

- Jozef Bilko:
  - Implementace k-d tree
  - Studium dostupných řěšení

- Marek Konečný:
  - Implementace Bounding Intervals Hiearchy
  - Dílčí dodělávky a úpravy aplikace

- Měření výsledků

Na dokumentaci a prezentaci členové pracovali společně.

Co bylo nejpracnější

Aplikace má nedostatek v uživatelském rozhraní. Jelikož okna View i Menu sdílí společný vykreslovací cyklus, při náročnějších scénách je uživatelské rozhraní (okno Menu a navigace kamery klávesnicí) značně neresponzivní. Toto jsme nestihli vyřešit, avšak pro demonstrační účely to nepovažujeme za problematické a částečně jsme to vyřešili tlačítkem Pause a méně náročným režimem vykreslení.

Struktura BVH ve scéně Sponza občas přeskočí polygony, což je způsobeno nepřesností použitého branchless algoritmu pro kontrolu průsečíku paprsku s AABB, a problém proto nejde vyřešit.

Zkušenosti získané řešením projektu

- Naučili jsme se napsat strukturu vykreslovacích programů s využitím ray/path tracingu.
- Prozkoumali jsme nové moderní funkcionality v jazyku C++17.
- Seznámili jsme se s novými typy akceleračních struktur a jejich metod optimalizace.

Autoevaluace

Koncept řešení: 70% (analýza, výběr článků, dekompozice problému, volba vhodných prostředků, …)

- Bylo přiděláno mnoho zbytečné práce s vývojem vlastní ray tracing enginu. Alespoň byla vyzkoušena nová architektura runtime modelu, jenž umožňuje využití "zero cost abstraction".

Realizace: 70% (kvalita získaných znalostí, kvalita a čitelnost kódu, obecnost řešení, znovupoužitelnost)

- Z hlediska obecnosti, je řešení skvělé pro přidání nových akceleračních struktur, které jsou implementovány formou základní třídy, ze které lze odvodit třídy další. Pro rozšíření je však nutno zaregistrovat každou novou třídu do runtime enginu a do nastavení programu.

Využití zdrojů: 90% (využití existujícího kódu a dat, využití literatury, …)

- Slušné využití různých teoretických materiálů a repositářů kódu volně dostupných na internetu, zdroje byly všechny řádně uvedeny v rámci dokumentace

Hospodaření s časem: 80% (rovnoměrné dotažení částí projektu, míra spěchu, chybějící části, …)

- Určitě jsou místa v projektu, co by šla doladit, kdyby bylo více času, ale i tak bylo zadání splněno a řešení odevzdáno do systému včas

Spolupráce v týmu: 90% (komunikace, dodržování dohod, vzájemné spolehnutí, rovnoměrnost, …)

- Už po konzultaci bylo jasné, že kdo bude co dělat a do kdy co je potřeba udělat, při práci tu byla ochota pomáhat i dělat věci nad rámec požadavků

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

- Projekt byl spíše náročný. Spousta věcí byla implementována úplně od základu (naše chyba).
- Zlepšili jsme se ve spolupráci na týmových projektech včetně práce s git.
- Ostatní získané dovednosti již byly jmenovány.
- Některé body zadání v IS jsou matoucí či si dokonce protiřečí (viz limit velikosti archivu či ter-míny odevzdání).

Doporučení pro budoucí zadávání projektů

- Nevyhovoval nám pouze limit 2 strany na dokumentaci, do kterého jsme se nezvládli vmístit, protože více členů znamená více nároků na projekt...
- Udělat revizi zadání v e-learningu.