

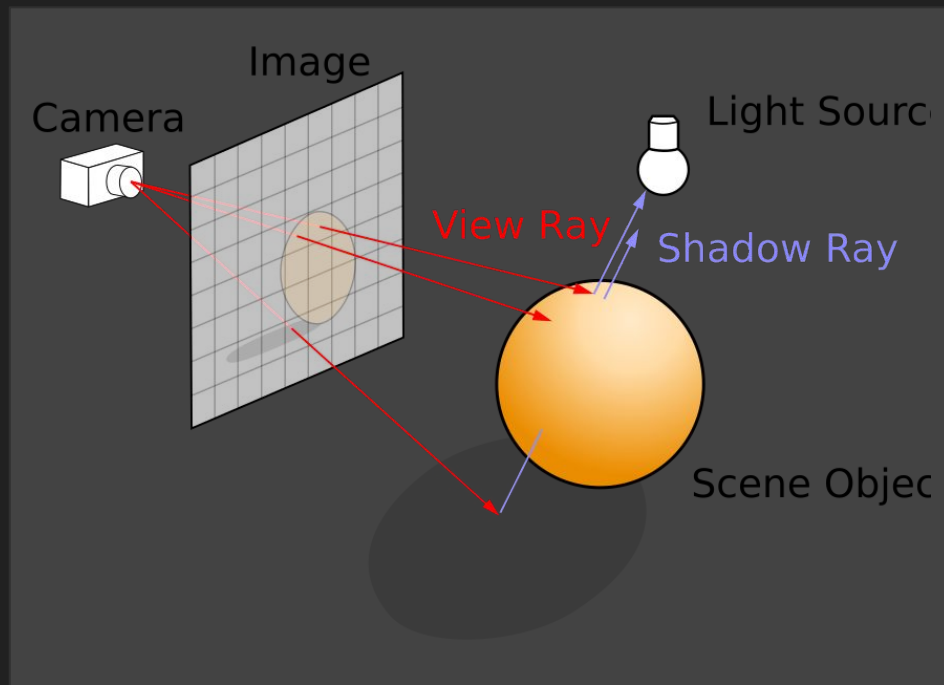
# Ray Tracer

Matúš Benček

8. 4. 2025

# Ray Tracing - Ako funguje

Ray tracing je technika vykresľovania obrazu, ktorá simuluje správanie svetla v reálnom svete. Sleduje lúče svetla od kamery cez scénu a vypočítava, ako sa odrážajú, lámu alebo pohlcujú pri kontakte s objektmi. Výsledkom sú realistické tieň, odlesky a osvetlenie.



# Použité technológie

- **Frontend (GUI):** Vue.js
- **Backend:** Go + Echo framework.
  - Beží asynchrone v separátnej goroutine.
  - Používa `unsafe` funkcie pre minimalizáciu manažmentu Go Rutín (lightweight Threads).
- **Ray Tracer:** Golang
  - Využíva Ebiten (2D engine) pre efektívne vykresľovanie pixelov.
  - Shader systém cez jazyk Kage (vytvorený špeciálne pre Ebiten).



echo



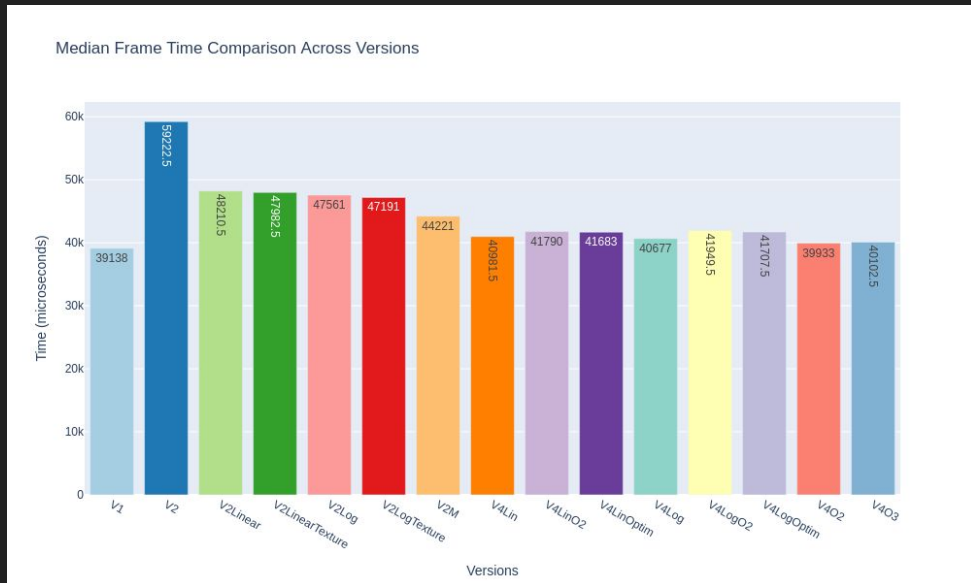
Vue.js



# Vývoj Ray Tracera

Počas vývoja bolo nevyhnutné riešiť výkonové problémy. Ray tracing je výpočtovo náročný, preto som:

- Zaviedol som systém benchmarkov slúžiaci na kontinuálne meranie zmien medzi verziami.
- Implementácia BVH a jej následná optimalizácia.
- Testoval som **SIMD** knižnice a GPU výpočty cez **CUDA**.
  - Bohužiaľ, kvôli kompromisom a časovému limitu neboli tieto riešenia implementované.



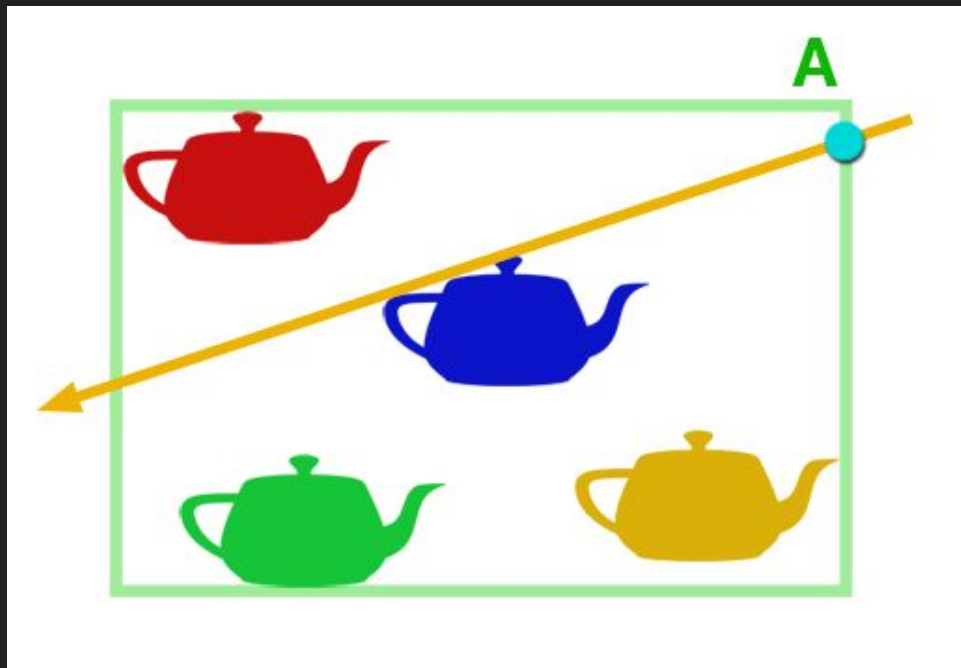
# Čo je BVH

## BVH – Bounding Volume Hierarchy

Je to hierarchická dátová štruktúra, ktorá slúži na efektívne rozdelenie 3D scény do menších celkov pomocou tzv. axis-aligned bounding boxov (AABB).

### Prečo BVH ?

- Namiesto toho, aby sme pri výpočtoch (napr. ray tracingu) testovali všetky trojuholníky v scéne ( $O(n)$ ),
- Testujeme len tie časti scény, ktoré sú potenciálne relevantné ( $O(\log n)$ ).

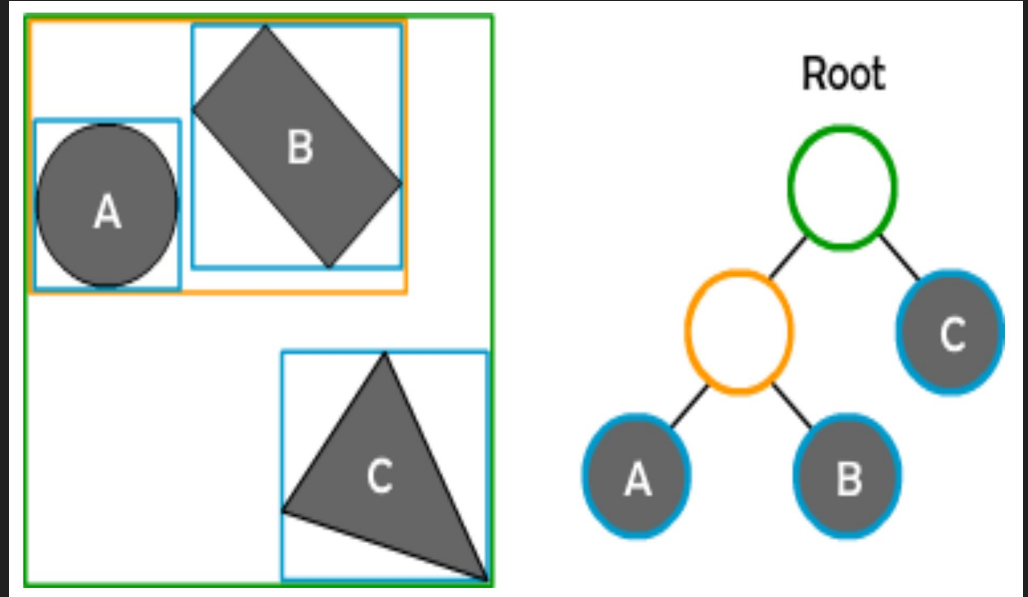


# Reprezentácie BVH

BVH sa zvyčajne implementuje ako **binárny strom**.

Alternatíva Reprezentácia: **Array reprezentácia** kde platí ľavý node =  $2n$  a pravý node =  $2n+1$

- Efektívnejšia pamäťovo aj výpočtovo.
- Z časových dôvodov nebola plne implementovaná.



# Optimalizácia BVH

Pôvodná verzia mala problém s narastajúcou veľkosťou z dôvodu narastajúceho množstva parametrov v priebehu vyvoja.

Od verzie 4 je BVH optimalizovaná:

- ~20% rýchlejšie.
- Zlepšená funkcia pre ray-box intersekciiu – teraz testuje oba boxy súčasne.
  - Výsledok: až **26% zrýchlenie** tejto operácie.

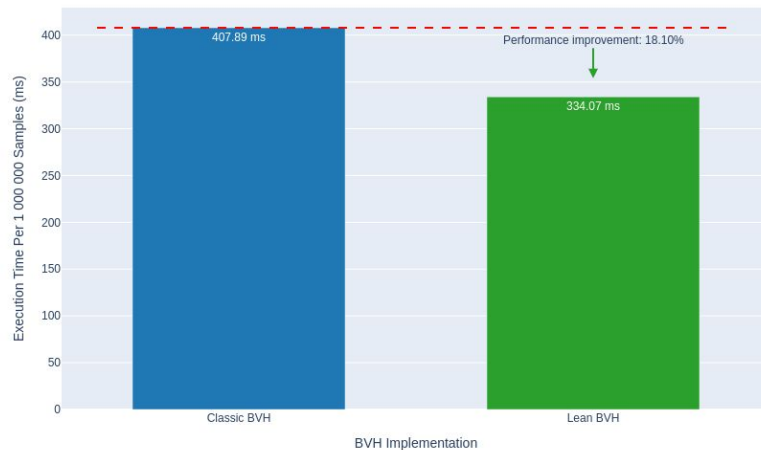
Stara Verzia:

```
type BVHNode struct { // size=136 (0x88)
    Left, Right *BVHNode
    BoundingBox [2]Vector
    Triangles   TriangleSimple
    active      bool
}
```

Nova Verzia:

```
type BVHLeanNode struct { // size=72 (0x48)
    Left, Right *BVHLeanNode
    TriangleBBOX TriangleBBOX
    active      bool
}
```

BVH Implementation Performance Comparison



# Vývojové verzie Ray Tracera

Postupne vznikali viaceré verzie, každá s novými funkciami a optimalizáciami:

- **Verzia 1**
  - Zakladna Implementacia RayTracera.
- **Verzia 2**
  - Viac Rialisticke spravenia svetla.
  - Podpora Gamma correction.  
linearny/logaritmicky model.
- **Verzia 4**
  - Podpora textúr.
  - Vylepšená BVH.

Každá nová funkcionálnosť však zvyšuje výpočtovú náročnosť.



Verzia 2M



Verzia 4Lin



# Voxel/Volume Rendering

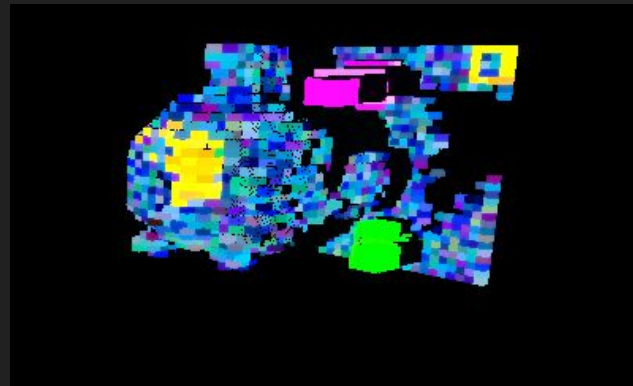
## Voxel Rendering

- Možnosť editovať voxeli počas behu aplikácie.
- Osvetlenie: naivná implementácia osvetlenia. Zisťuje iba, či je voxel v tieni a vzdialenosť od zdroja svetla.

Voxely uložené v 1D poli pre rýchlejšie indexovanie taktiež je použitý package unsafe aby sa vyhlo bounds checking-u.

## Volume Rendering

- Využíva Beer-Lambert aproximáciu.
- Recykluje existujúcu voxelovú infraštruktúru.



Render Voxelov



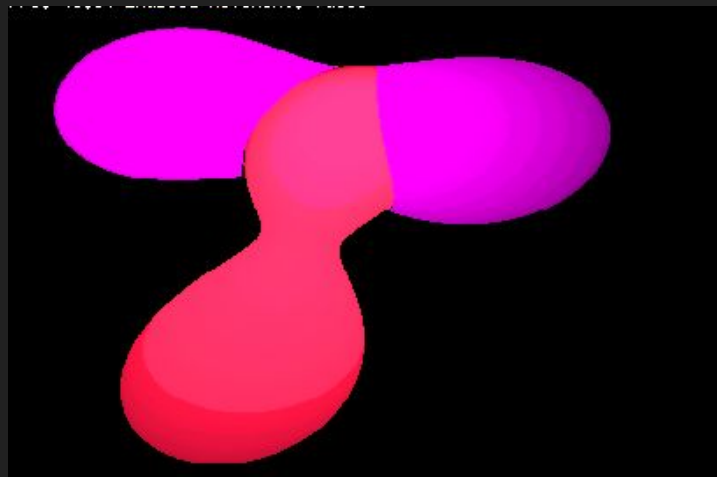
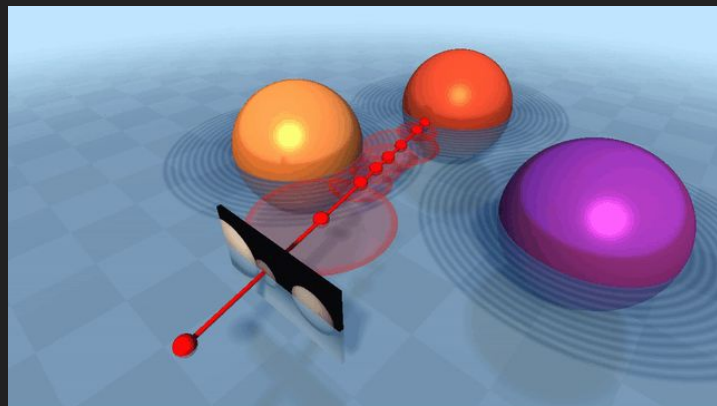
Render Volumu

# Ray Marching

Ray marching je pokročilá technika renderovania využívajúca SDF (Signed Distance Functions):

- **SDF** – matematické funkcie, ktoré pre každý bod v priestore vracajú vzdialenosť k najbližšiemu povrchu.
  - Kladná hodnota = bod je mimo objektu.
  - Záporná hodnota = bod je vnútri objektu.
  - Nulová hodnota = bod leží presne na povrchu.
- Princíp ray marchingu:
  - Lúč postupuje krokmi, ktorých veľkosť je určená vzdialenosťou k najbližšiemu objektu
  - Tam, kde  $SDF = 0$ , lúč narazí na povrch objektu.

Hlavnou výhodou Ray Marchingu je možnosť operácií medzi objektami ako union, subtraction, intersection.



Ďakujem za pozornosť