



Sveučilišni prijediplomski studij Informatika

Tin Švagelj

Metode rasterizacije volumetrijskih struktura u računalnoj grafici

Završni rad

Mentor: doc. dr. sc., Miran Pobar

Rijeka, 15. ožujka 2024.

(Iza naslovne stranice, na ovome mjestu, prilikom uvezivanja umetnite original zadatka završnog rada kojeg ste preuzeli od mentora)

Sažetak

Treba biti 100-300 riječi:

Cilj završnog rada je proći kroz različite metode rasterizacije volumetrijskih podataka u području računalne znanosti.

Rad započinje s uvođenjem različitih podatkovnih struktura i njihove primjene u različitim područjima poput medicine, geoprostornoj analizi, **DODAJ SADRŽAJ** i računalnim igrama. Zatim ulazi u temu rasterizacije takvih podataka i njihovog prikaza.

Br. riječi: 47

Ključne riječi: računalna grafika; vokseli; rasterizacija

SADRŽAJ

1. Uvod	1
1.1. Primjene volumetrijskih podataka	1
1.2. Komercijalni primjeri	1
2. Strukture za pohranu volumetrijskih podataka	2
2.1. 3D polja	2
2.2. Octree	2
2.2.1. SVO	2
2.2.2. DAG	2
2.3. Point-cloud data	2
3. Prijevremena rasterizacija	3
3.1. Ray casting	3
3.2. Splatting	3
3.3. Shear warp	3
4. Rasterizacija u realnom vremenu	4
4.1. GPU streaming	4
4.2. Metode optimizacije	4
5. Animacije	5
6. Usporedba s poligonima	6
6.1. Problemi	6
6.1.1. Izgled	6
6.2. Interaktivnost	6
6.3. Košta	6
7. Opis praktičnog dijela	7
7.1. Tech stack	7
7.1.1. Bevy PBR pipeline	7
7.2. Render pipeline	7
7.3. Analiza performansa	7
8. Zaključak	8
Literatura	9
Popis priloga	10

1. Uvod

- <https://www.sciencedirect.com/topics/computer-science/volumetric-dataset>
- <https://developer.nvidia.com/gpugems/gpugems/part-vi-beyond-triangles/chapter-39-volume-rendering-techniques>
- https://en.wikipedia.org/wiki/Volume_rendering
- https://web.cse.ohio-state.edu/~shen.94/788/Site/Reading_files/Leovy88.pdf

1.1. Primjene volumetrijskih podataka

- Medicina - <https://www.sciencedirect.com/topics/computer-science/volumetric-data>
 - Rendgenska tomografija
 - Elektronski mikroskopi
 - (engl. *Transmission Electron Microscopy*, TEM) i (engl. *Scanning Transmission Electron Microscopy*, STEM)
 - DICOM format
- Geoprostorna analiza
- Računalne igrice

1.2. Komercijalni primjeri

- C4, by Terathon <http://www.terathon.com/>
 - discontinued
- <https://voxelfarm.com/index.html>
- <https://gareth.pw/web/picavoxel/>
- Euclideon
 - Koristi point cloud mislim
 - claim: "Ne koristi GPU pipeline"

2. Strukture za pohranu volumetrijskih podataka

- Staviti primjere struktura u obliku slika i
struct

2.1. 3D polja

- Jednostavna i najčešća implementacija za real time render
- Postoji relativno puno primjera, alogritama, ...

2.2. Octree

Navesti raštrkanu i zbijenu/linearnu strukturu.

2.2.1. SVO

- Komplicirana implementacija
 - Postoji već gotov shader kod za ovo u par shading jezika negdje
- https://research.nvidia.com/sites/default/files/pubs/2010-02_Efficient-Sparse-Voxel/laine2010tr1_paper.pdf

2.2.2. DAG

- Varijanta SVOa, navesti razlike.
- Grozne karakteristike izmjena (po defaultu)
 - <https://github.com/Phyronnaz/HashDAG>

2.3. Point-cloud data

Spremljeno u Octreeu zapravo?

Laserski skeneri ju generiraju, česta primjena u geoprostornoj analizi.

- Dronovi ju koriste za navigaciju.

3. Prijevremena rasterizacija

- Primjene: medicina, statične scene i slike

3.1. Ray casting

- Skoro je i real time sada, mislim da je noisy, ima light propagation delay, ...

3.2. Splatting

Transparentni slojevi za brz prolazak kroz slojeve MRI slike.

3.3. Shear warp

4. Rasterizacija u realnom vremenu

- <https://0fps.net/2012/06/30/meshing-in-a-minecraft-game/>
- https://www.reddit.com/r/VoxelGameDev/comments/j89l6j/texturing_with_greedy_meshing/
 - Vokseli sa teksturama nisu vokseli, no ovo je temelj za povezivanje drugih podataka (boje u praktičnom dijelu, occlusion, normale, ...)

4.1. GPU streaming

Opisati kako LOD funkcioniра za svijet sastavljen od chunkova.

4.2. Metode optimizacije

- <https://acko.net/blog/teardown-frame-teardown/>

5. Animacije

- Ona metoda gdje se generira AABB za segmente koji međusobno ne colideaju i koristi skele-ton
- Metoda sa deformacijom vokselaa
 - Nije “pravi” voxel renderer
- Metoda gdje se u compute shaderu samplea animirani triangle mesh svaki frame
 - Izgleda meh i relativno je zahtjevno
- Metoda gdje je definirana funkcija koja mapira deltatime na konfiguraciju vokselaa
- Opisati kako je DAG neprikladan za animiranje - ili spor ili jako potrošan glede memorije

6. Usporedba s poligonima

S obzirom na brojne postojeće metode lažiranja fizike, prikaza, itd. radi li se o stilističkom izboru?

6.1. Problemi

- Kako zaobliti svijet
 - Je li vrijedno spomena, vrlo specifično za računalne igre...
 - Postoji negdje efekt sa shaderima; ne rješava probleme topologije
 - <https://www.youtube.com/watch?v=bJr4QlDxE ME>

6.1.1. Izgled

- Sampliranje uvijek narušava kvalitetu modela, ili zauzima previše memorije
 - Artisti su naviknuti raditi s trokutima
 - Nije bitno za proceduralan sadržaj
 - Marching cubes i djeca
- Vrlo teško modelirati nepravilne oblike

6.2. Interaktivnost

- “Svijet nije maketa od papira nego ima volumen”.
- Postoje metode lažiranja nekih vrsta interakcija.
 - Dodavanje svake zahtjeva zaseban trud dok je voxel engine uniforman (veliki upfront cost, lakše dodavanje sadržaja)

6.3. Košta

- Kako bi vokseli bili praktični u real time aplikacijama potrebno je jako puno rada u optimizaciji njihove strukture u GPU memoriji, rendering kodu, LOD sustavu, ...
- Nvidia paper je dobar početak ali ne zadovoljava mnogo zahtjeva modernih računalnih igara
 - Mislim da nema transparentnost?

7. Opis praktičnog dijela

7.1. Tech stack

Vizual sa strukturom enginea.

Navesti kako Bevy dozvoljava lagane izmjene cijelih gromada enginea i kako Traitovi iz Rusta tome pridonose.

7.1.1. Bevy PBR pipeline

7.2. Render pipeline

Neki cool vizual koji prikazuje tok podataka.

Objasniti gdje točno se uključuje moj kod.

- cca je kao objekt sa custom shaderom u Unityu/UE4/Godotu, samo taj shader shadea cijeli teren i varijanta istog bi objekte u svijetu (da postoji)

7.3. Analiza performansa

- Navesti da je dodavanje transparentnosti veliki minus za performanse.

8. Zaključak

Literatura

Popis priloga