

Voxelhashing

3D rekonštrukcia v reálnom čase

Viktor Lučkanič
Patrik Herčút

Riadenie mobilných robotov
LS 2019/2020

Obsah

- Charakteristika
- Využitie
- Princíp (postup)
- Dátová štruktúra
- Kolízie
- Hash operácie
- Alokácia voxel blokov
- Integrácia voxel blokov
- Extrakcia povrchu
- Streaming
- Príklad - výsledky

Charakteristika

- metóda online rekonštrukcie priestoru
 - skenovanie pomocou depth kamery
-
- fúzia prekrývajúcich sa hĺbkových máp
 - lineárna hash schéma → rýchly prístup k dátam
-
- geometria priestoru je neznáma → dynamická alokácia

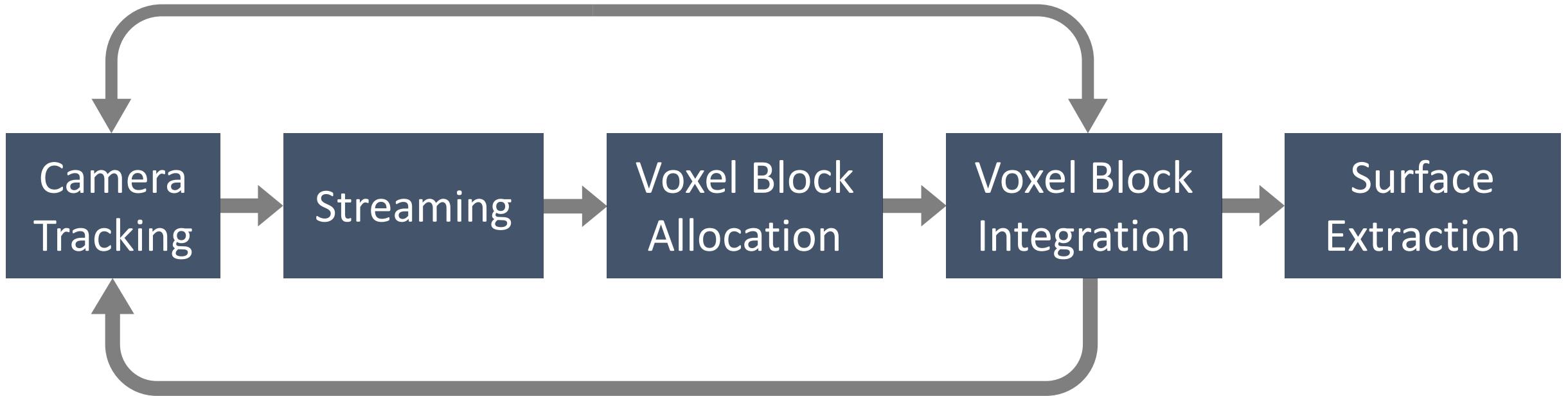
Charakteristika

- nízka výpočtová náročnosť
- menšie vplyvy driftu
- minimalizácia počtu kolízii
- tabuľka neobsahuje duplicity
- nízka pamäťová náročnosť → prázdne sa neukladá

Využitie

- okamžitá spätná väzba pri skenovaní priestoru pri vytváraní 3D objektov
- 3D priestorové mapy pre navádzanie robotov
- rozšírená realita

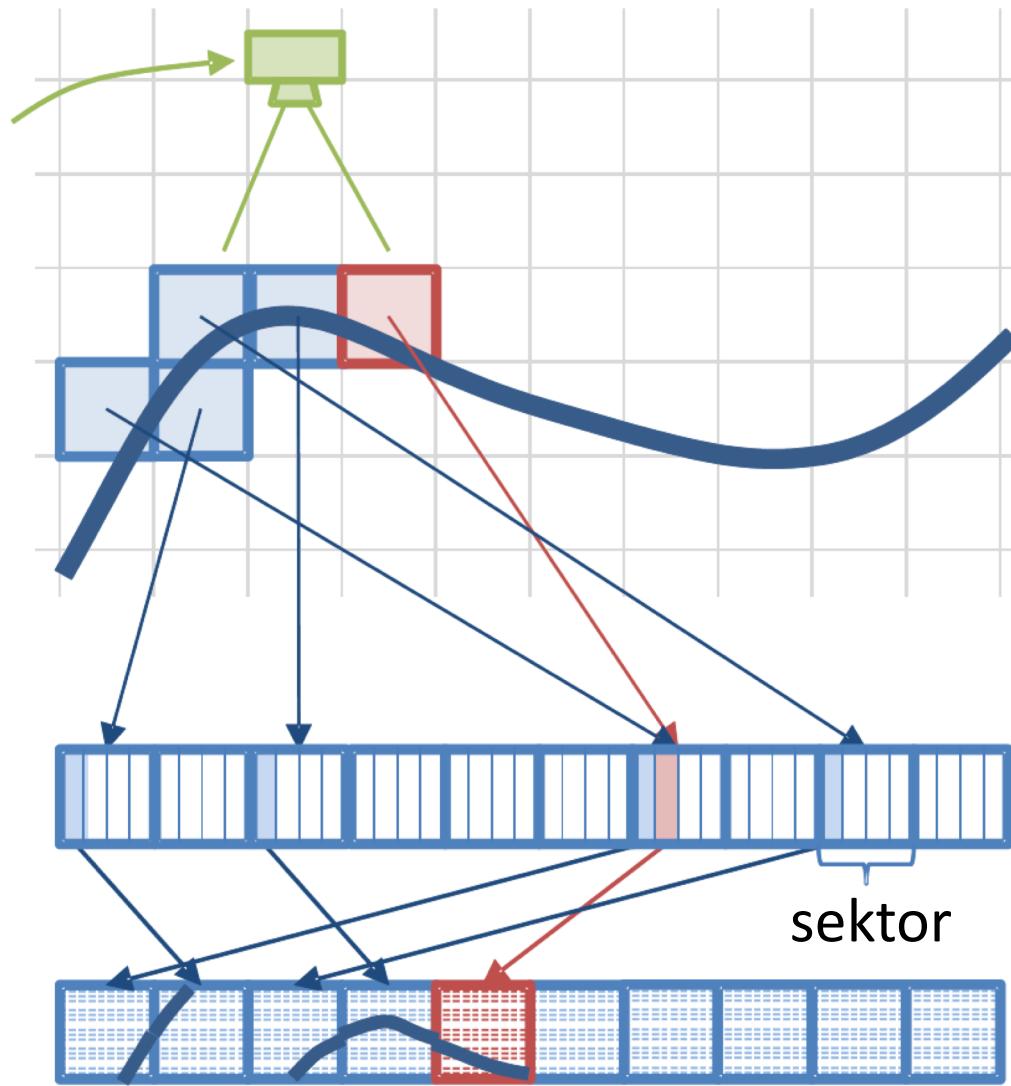
Input Depth Map



Princíp voxel hash

- voxel bloky alokované len okolo zrekonštruovanej plochy
- hash tabuľka obsahuje položky
 - každá obsahuje ukazovateľ prideleného voxel bloku, svetové súradnice, príp. offset na ďalšie položku
- voxel bloky sa dajú získať z hash tabuľky pomocou svetových súradníc
- mapovanie zo svetových súradníc (x,y,z) na hash hodnotu $H(x,y,z) \rightarrow$ funkcia:

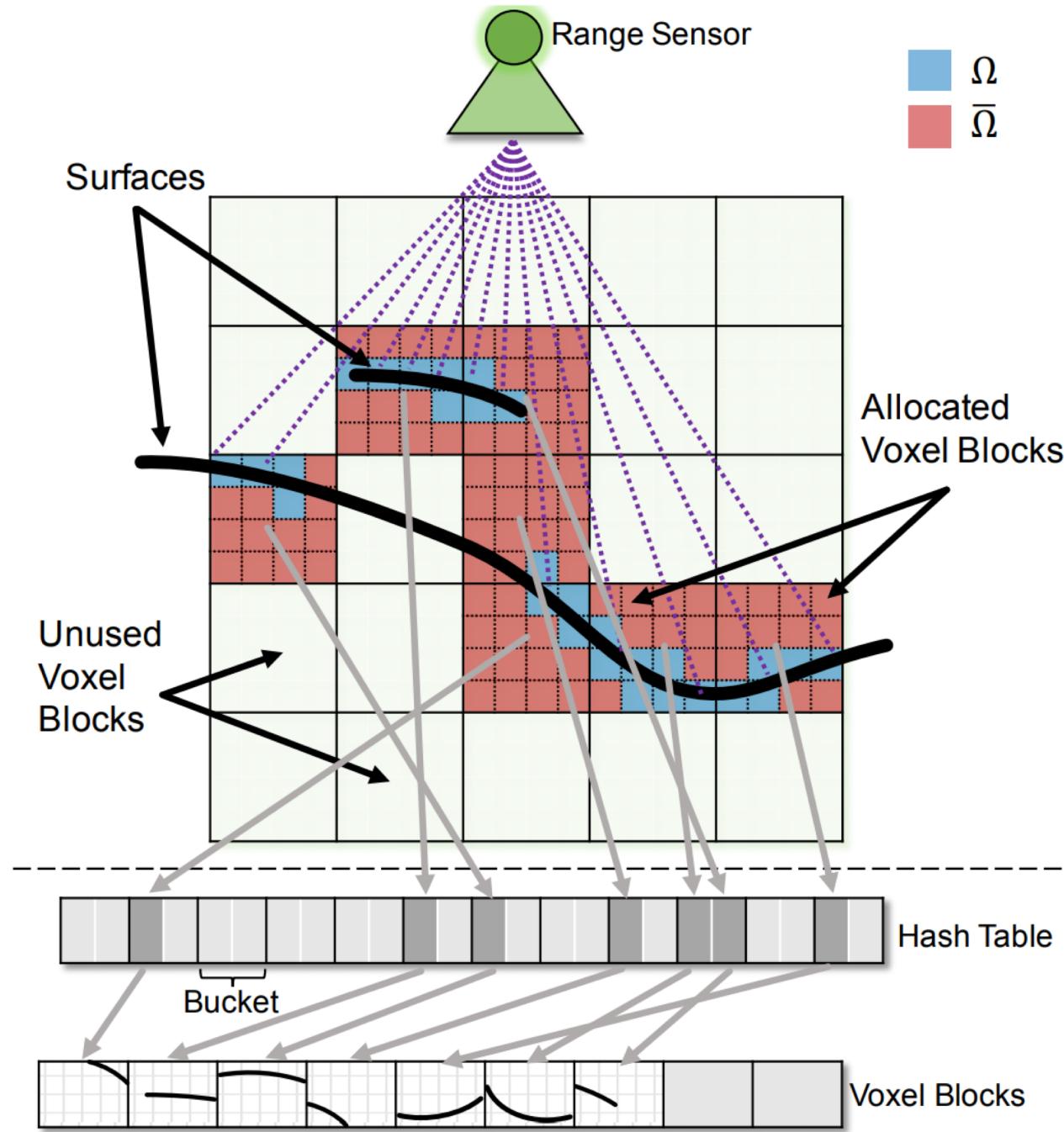
$$H(x, y, z) = (x \cdot p_1 \oplus y \cdot p_2 \oplus z \cdot p_3) \bmod n$$



svet

hashovacia
tabuľka

voxelové
bloky



Dátová štruktúra

```
struct Voxel{  
    float sdf;  
    uchar colorRGB[3];  
    uchar weight;  
};
```

```
struct HashEntry{  
    short position[3];  
    short offset;  
    int pointer;  
};
```

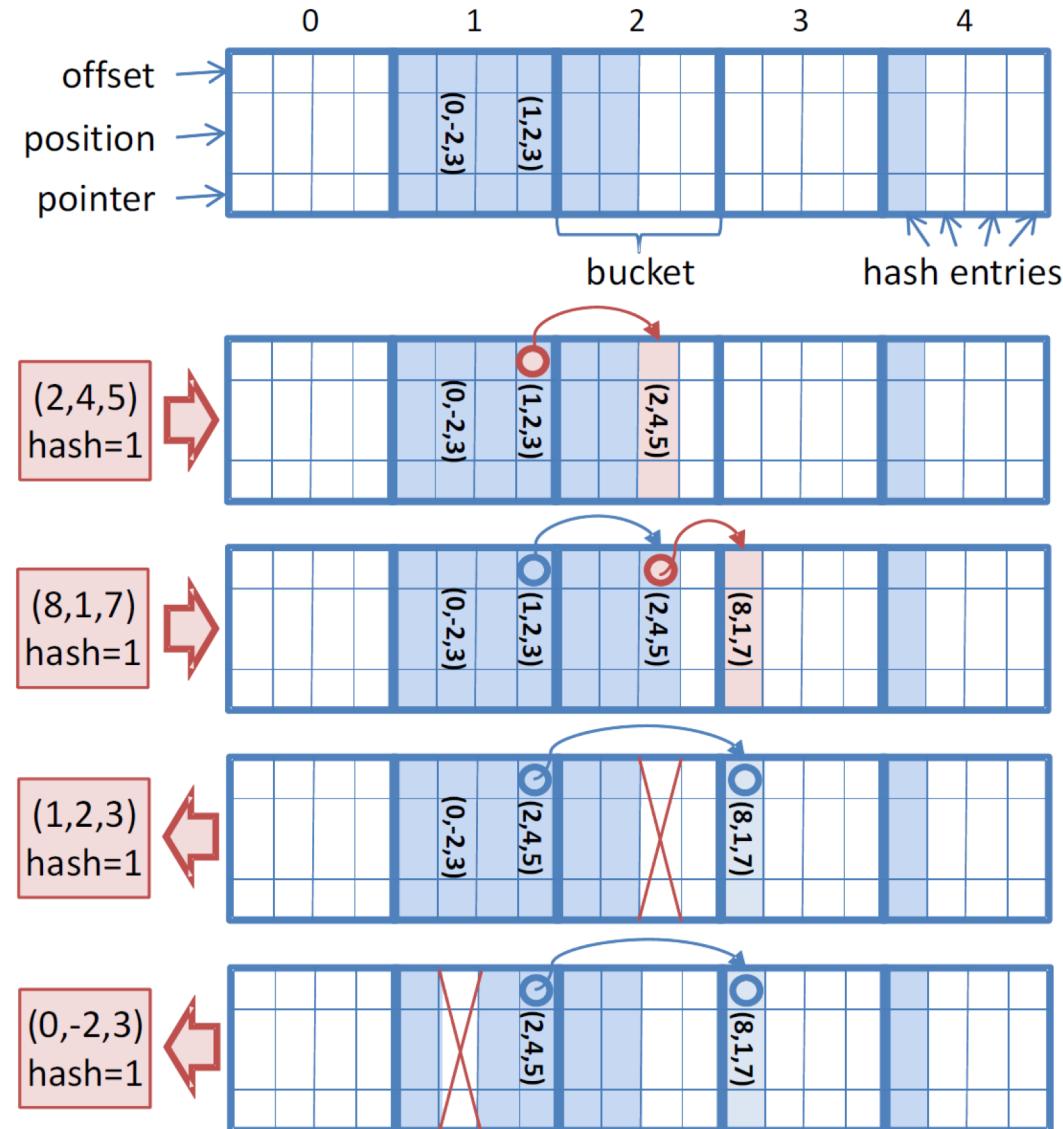
Kolízie

- alokácia blokov na rovnakú hash hodnotu
- riešenie → sektor
- sektor obsahuje niekoľko hash vstupov
- pri pretečení sektoru → pripojenie zretázeného zoznamu
- pointer na zretázený zoznam uložený v offsete

Hashovacie operácie

- 1. Vkladanie**
 - výpočet hash funkcie
 - nájdenie prázdnego miesta
 - ak nie je prázdne miesto v sektore → pripojenie linked list
- 2. Čítanie**
 - výpočet hash funkcie
 - prehľadanie príslušného sektoru (celý, aj keď by boli prázdne miesta, príp. linked list)
- 3. Mazanie**
 - pre svetové súradnice výpočet hash funkcie
 - pri prehľadaní sektora, ak nie je posledný (neobsahuje pointer na linked list), tak môžeme vymazať
 - ak je posledný (má offset hodnotu) táto hodnota sa skopíruje na posledný vstup v sektore
 - ak sa nachádza v linked list → nasledujúci slajd

Hashovacie operácie - vizualizácia



Alokácia voxel blokov

- pre každú vzorku vstupnej hĺbkovej mapy sa pomocou lúčov skenuje viditeľný región
- určenie, kde lúč zasahuje
- vstupy voxel blokov vložíme do hash tabuľky
- alokácia pamäte na GPU

Signed Distance Field

Truncated Signed Distance Field

Integrácia voxel blokov

TSDF = Truncated Signed Distance Function

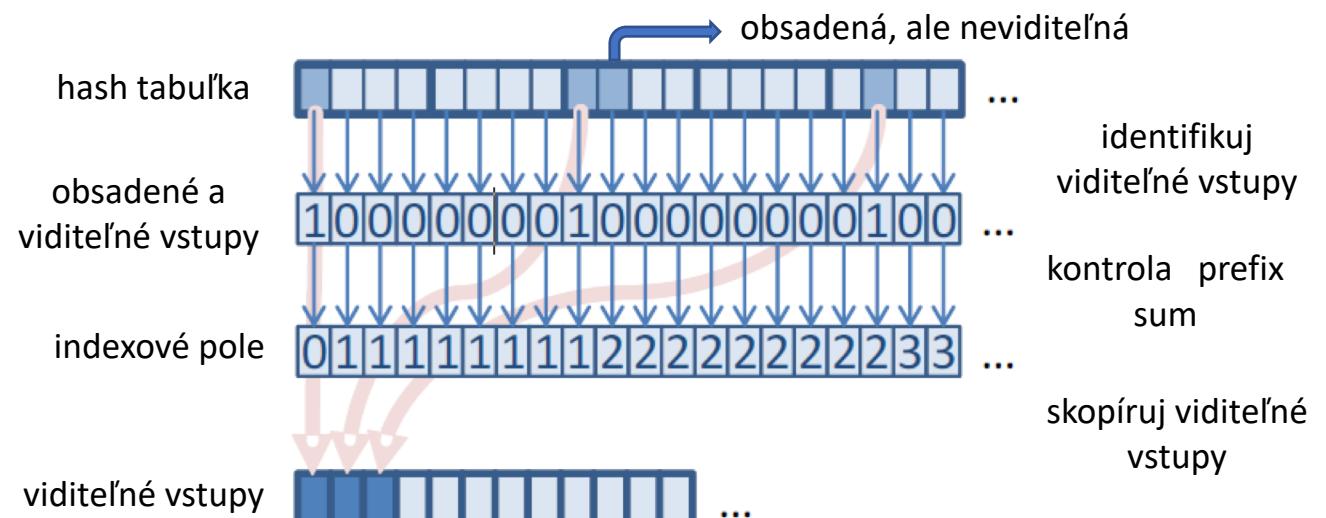
Selekcia voxel blokov

- uloženie hash hodnôt voxel blockov do nového buffra
- iba tie, ktoré sa nachádzajú v aktuálnom pohľade (sú viditeľné)



Aktualizácia povrchu

- list hash vstupov je potom paralelne spracovaný pre aktualizáciu TSDF hodnôt



Voxel block integration

Aktualizácia povrchu

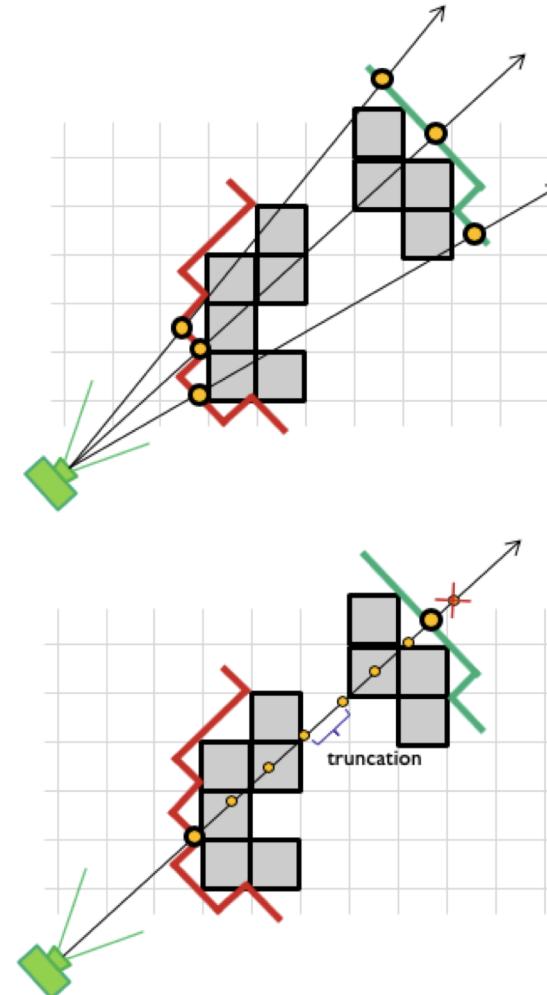
- bližšie vzdialenosť majú väčšiu váhu → menší šum. Takisto je to s farbami.
- identifikácia všetkých blokov danej oblasti pre nájdenie kandidátov do ďalšej fázy

Garbage collection

- odstránenie voxel blokov zašumených okrajov a pohybujúcich sa plôch
- výpočet minima abs. hodnoty TSDF a maxima weight pre každý pridružený voxel blok
- ak maximálna weight je 0 alebo minimum TSDF je viac ako treshold → vymazanie bloku

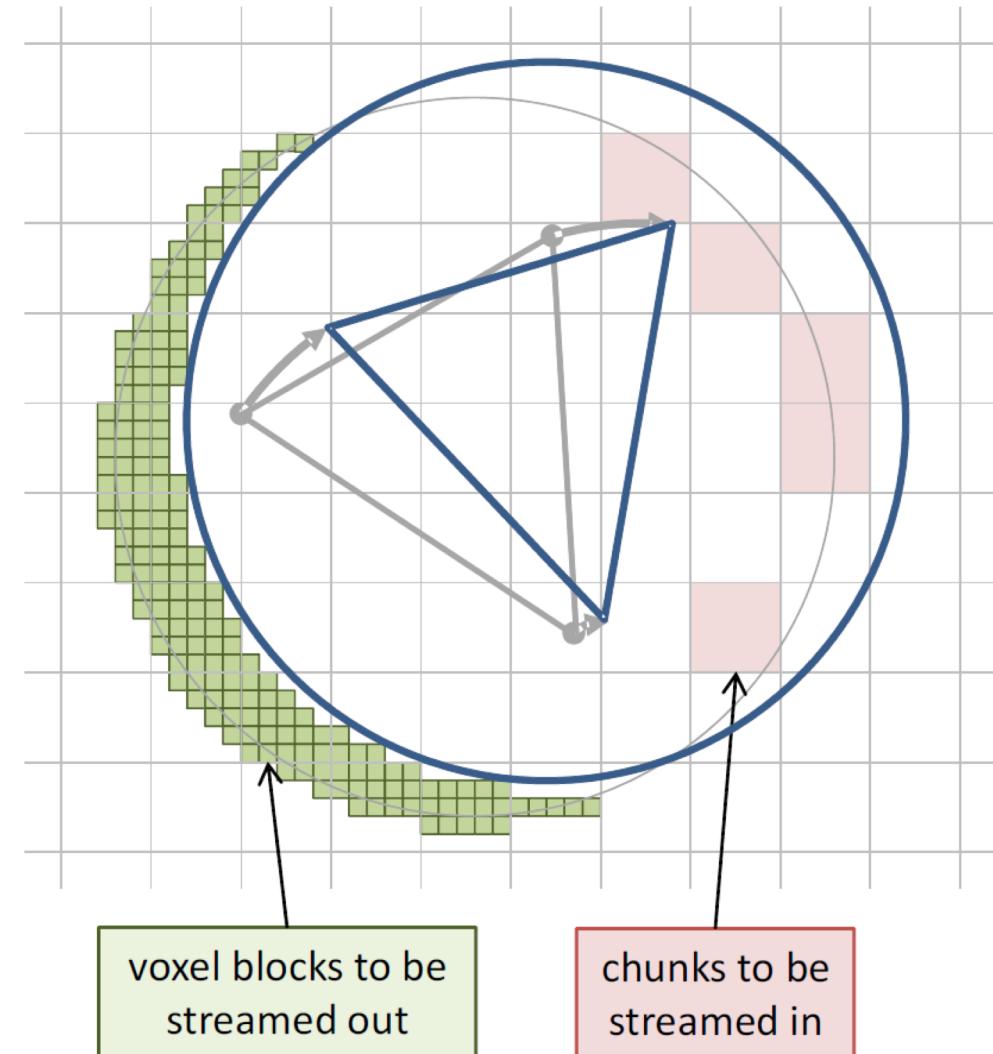
Extraktia povrchu

- hranice minimálnej a maximálnej hĺbky (ray interval)
 - paralelne rastrovanie všetkých voxel blokov
-
- pre každý lúč ohodnotenie TSDF
 - v prípade prázdnego priestoru trilineárna interpolácia



Camera tracking

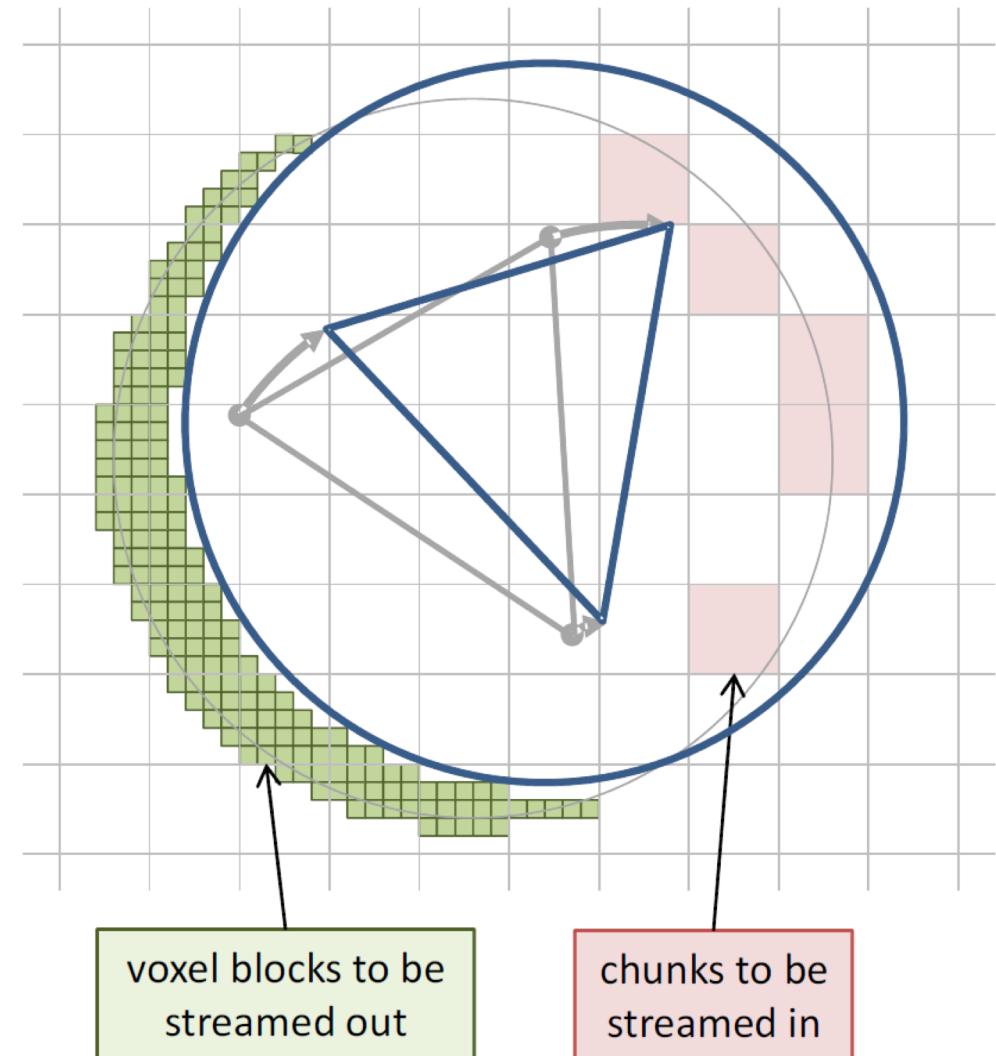
- pomocou nasledujúceho vstupného framu a hĺbkovej mapy vieme určiť odhad polohy
- pri pohybe kamery zľava doprava zelené voxel bloky ostávajú mimo obrazu, zatiaľ čo červené sa do obrazu dostávajú



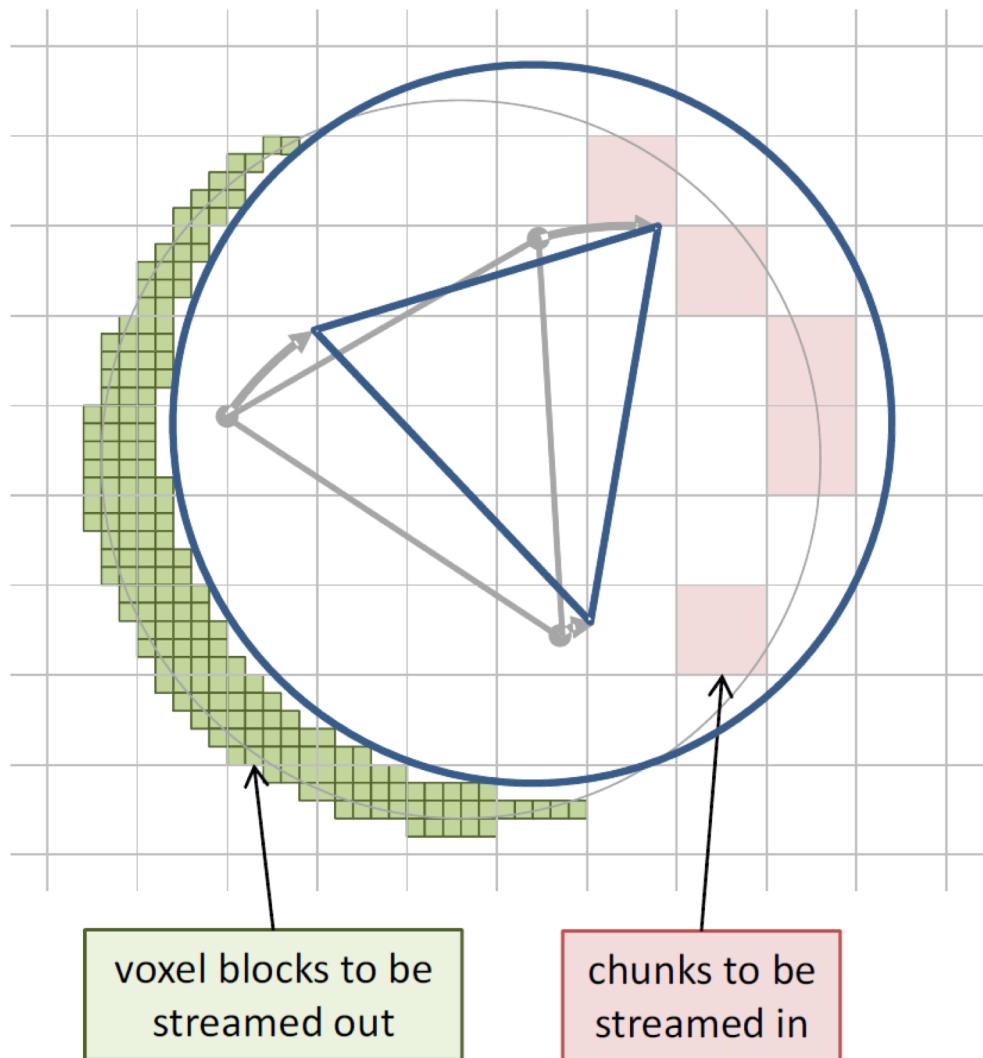
Streaming

GPU-to-Host

- pri odoberaní voxel blokov z aktívnej oblasti
- hash vstupy pre dané bloky sú vymazané a cez pomocný buffer presunuté do heapu pre ďalšie použitie
- v host voxel dáta nie sú v hash tabuľke ale v chunks



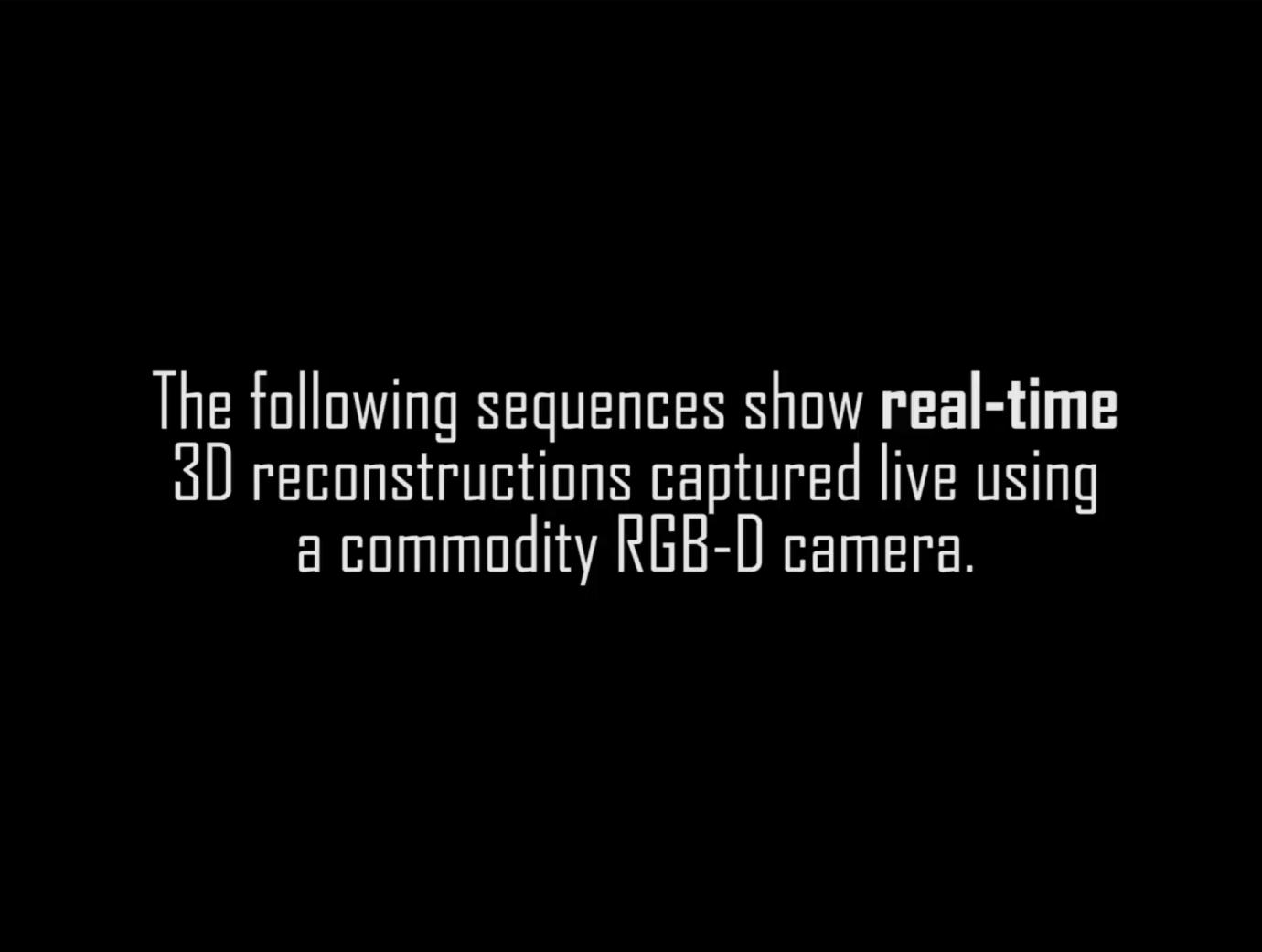
Streaming



Host-to-GPU

- identifikácia chunks, ktoré celé padnú do aktívnej oblasti pri pohybe do už zrekonštruovanej oblasti
- vtedy budú všetky voxely prenesené do GPU

Online rekonštrukcia



The following sequences show **real-time** 3D reconstructions captured live using a commodity RGB-D camera.

Výstupný obraz



Výstupný obraz



Zdroje

- Dostupné na internete:
<https://niessnerlab.org/papers/2013/4hashing/niessner2013hashing.pdf>
- <https://arxiv.org/pdf/1604.03734v1.pdf>
- <http://www.graphics.stanford.edu/~niessner/niessner2013hashing.html>



**Ďakujeme za
pozornosť**