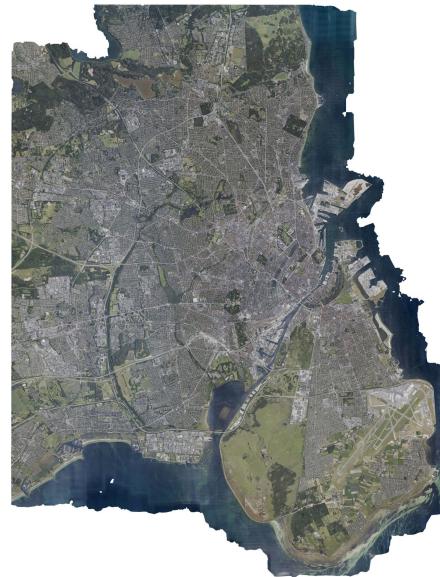


Out-of-Core Surface Reconstruction via Global TGV Minimization

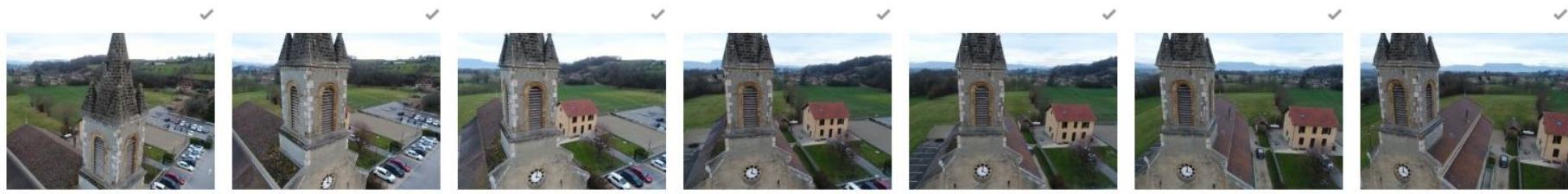
Николай Полярный

polarnick@agisoft.com

Agisoft



Фотограмметрия



DJI_0127

DJI_0128

DJI_0129

DJI_0130

DJI_0131

DJI_0132

DJI_0133



DJI_0134

DJI_0135

DJI_0136

DJI_0137

DJI_0138

DJI_0139

DJI_0140



DJI_0141

DJI_0142

DJI_0143

DJI_0144

DJI_0145

DJI_0146

DJI_0147

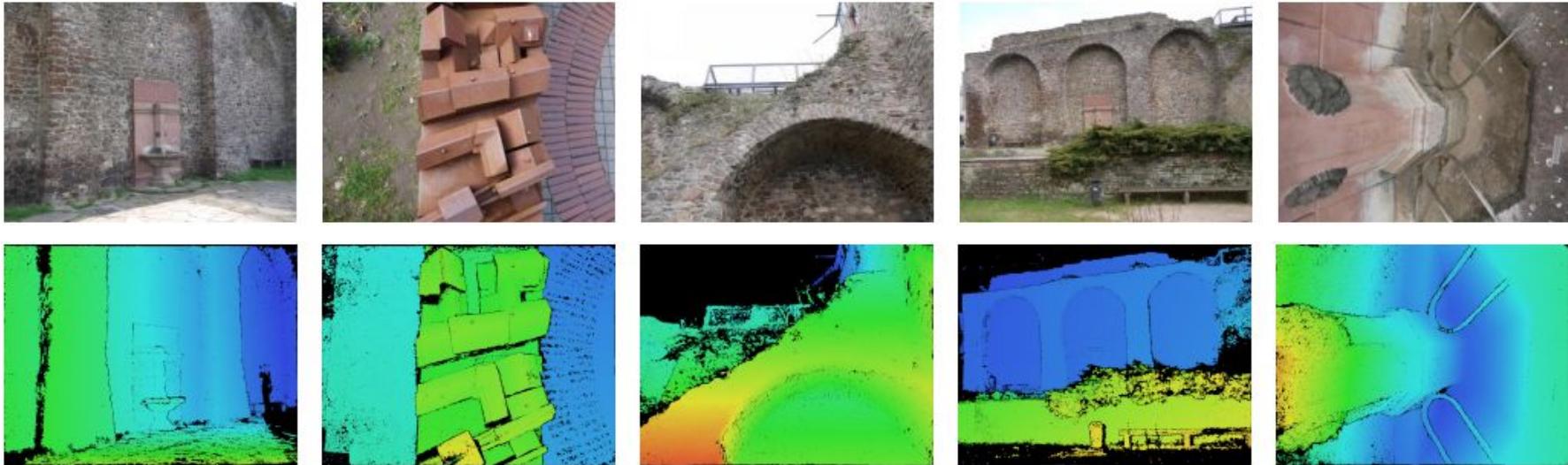
Фотограмметрия

По множеству фотографий
восстановить 3D модель:

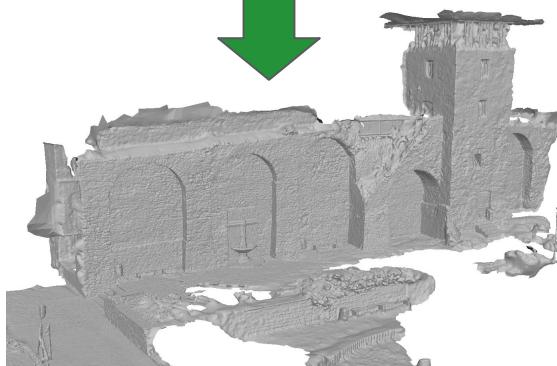
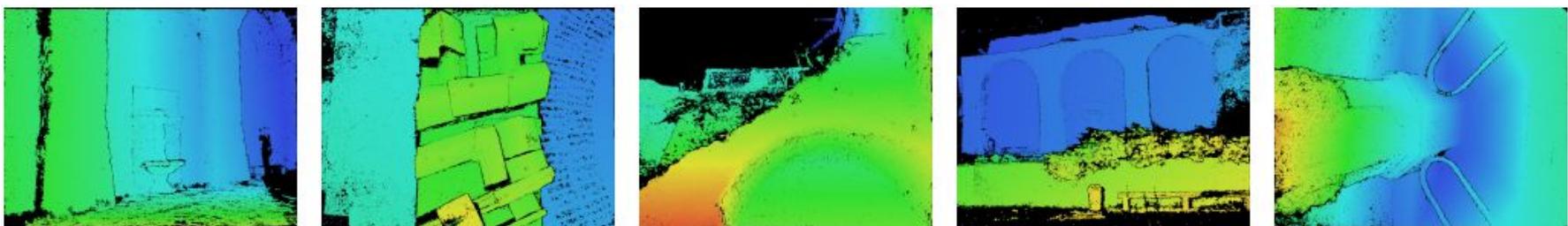


Данные предоставил Stéphane Prodent

Входные данные: карты глубины (Depth Maps)



Входные данные: карты глубины (Depth Maps)



План

- 1) Численный метод на регулярной решетке [**Zach2007**]
- 2) Численный метод на адаптивном октодереве **GDMR** [**Ummenhofer2017**]
- 3) Out-of-Core адаптация [**Poliarnyi2021**]

A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration, Zach et al., 2007

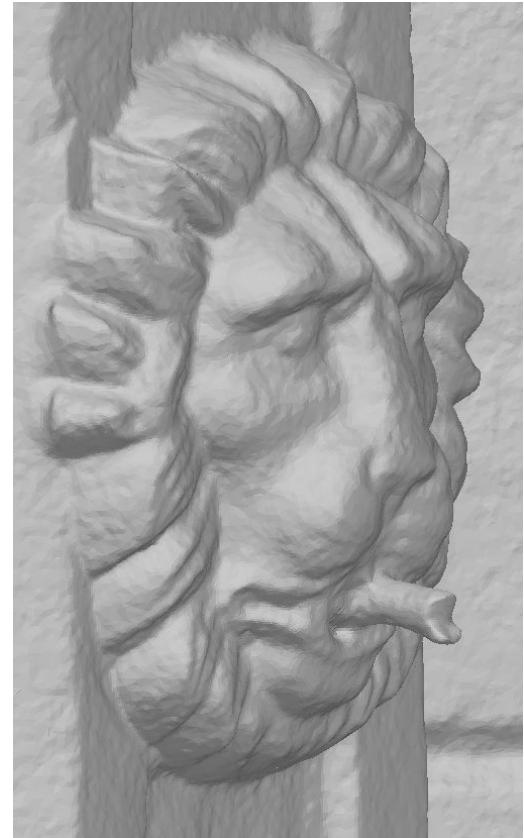
Global, Dense Multiscale Reconstruction for a Billion Points, Ummenhofer et al., 2017

Out-of-Core Surface Reconstruction via Global TGV Minimization, Poliarnyi, 2021

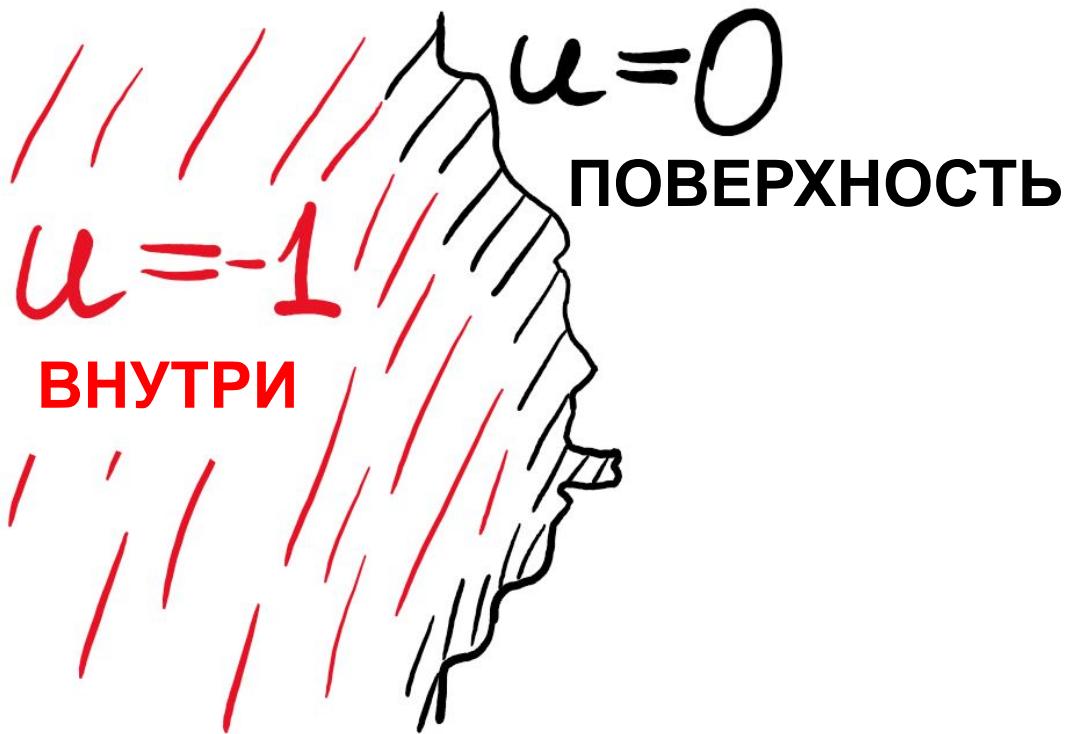
Представление пространства
(индикаторное поле):



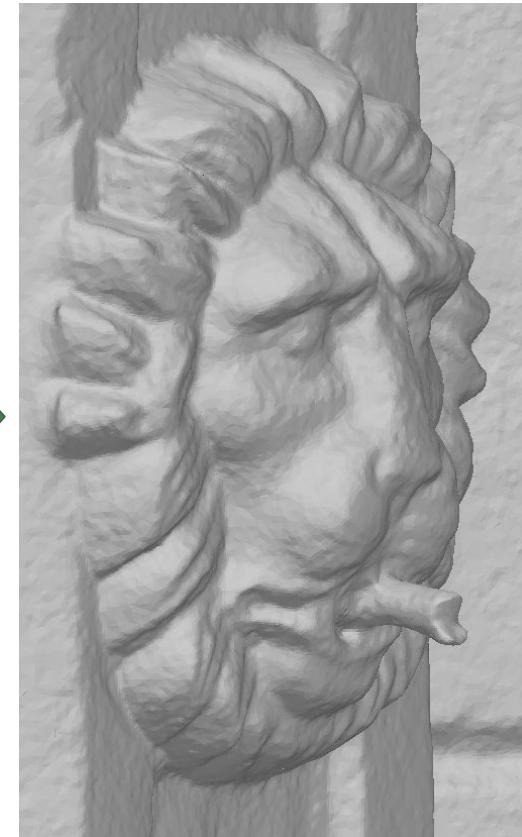
Полигональная
модель:



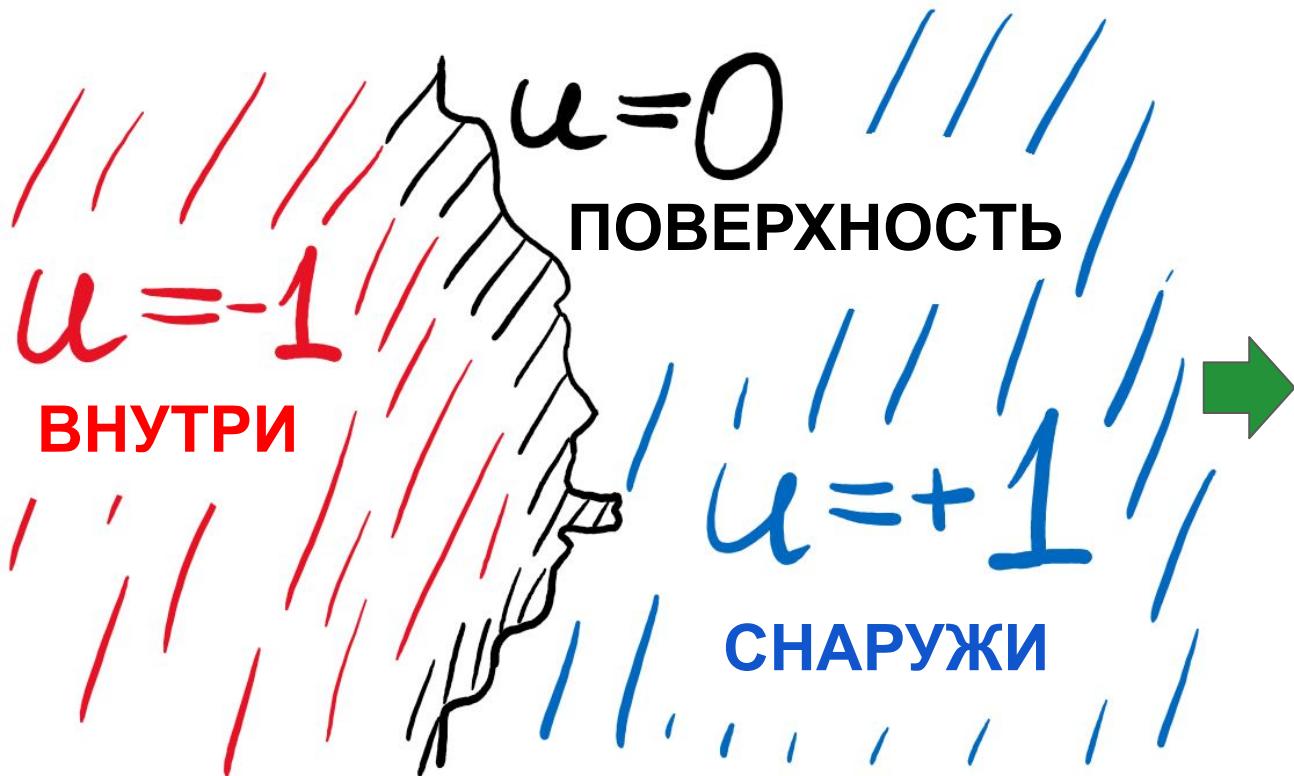
Представление пространства
(индикаторное поле):



Полигональная
модель:



Представление пространства
(индикаторное поле):



Полигональная
модель:



Требования к искомому индикаторному полю

- 1) Должно быть гладким (**regularization term**) - минимизируем полную вариацию
- 2) Должно быть согласовано с картами глубины (**data term**)

TV-L¹:

$$E = \int_{\Omega} \left\{ |\nabla u| + \lambda \sum_{i \in \mathcal{I}(\vec{x})} w_i(\vec{x}) |u - f_i| \right\} d\vec{x}$$

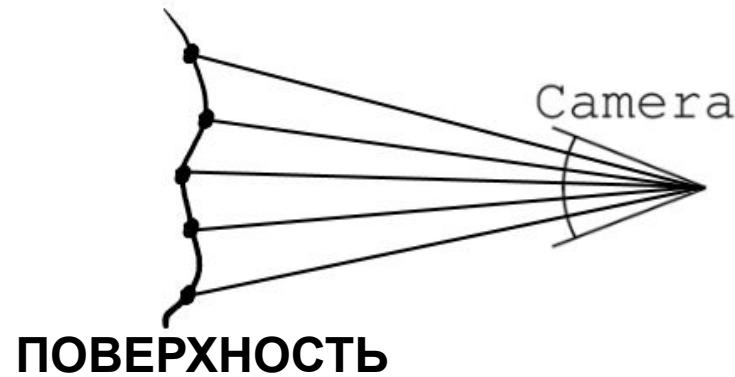
u — индикаторное поле

f_i — индикаторное поле
соответствующее
карте глубины i

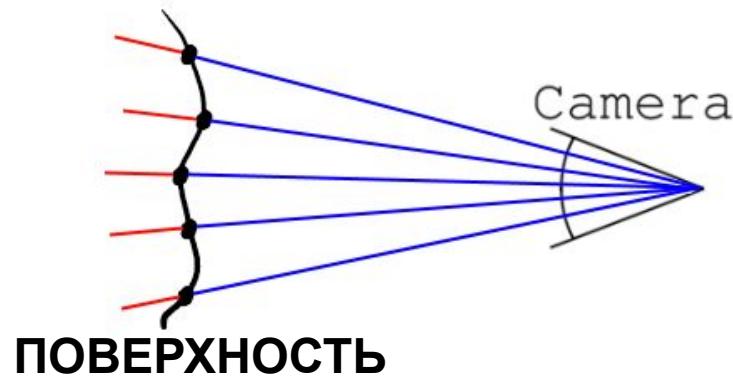
Карта глубины в виде индикаторного поля



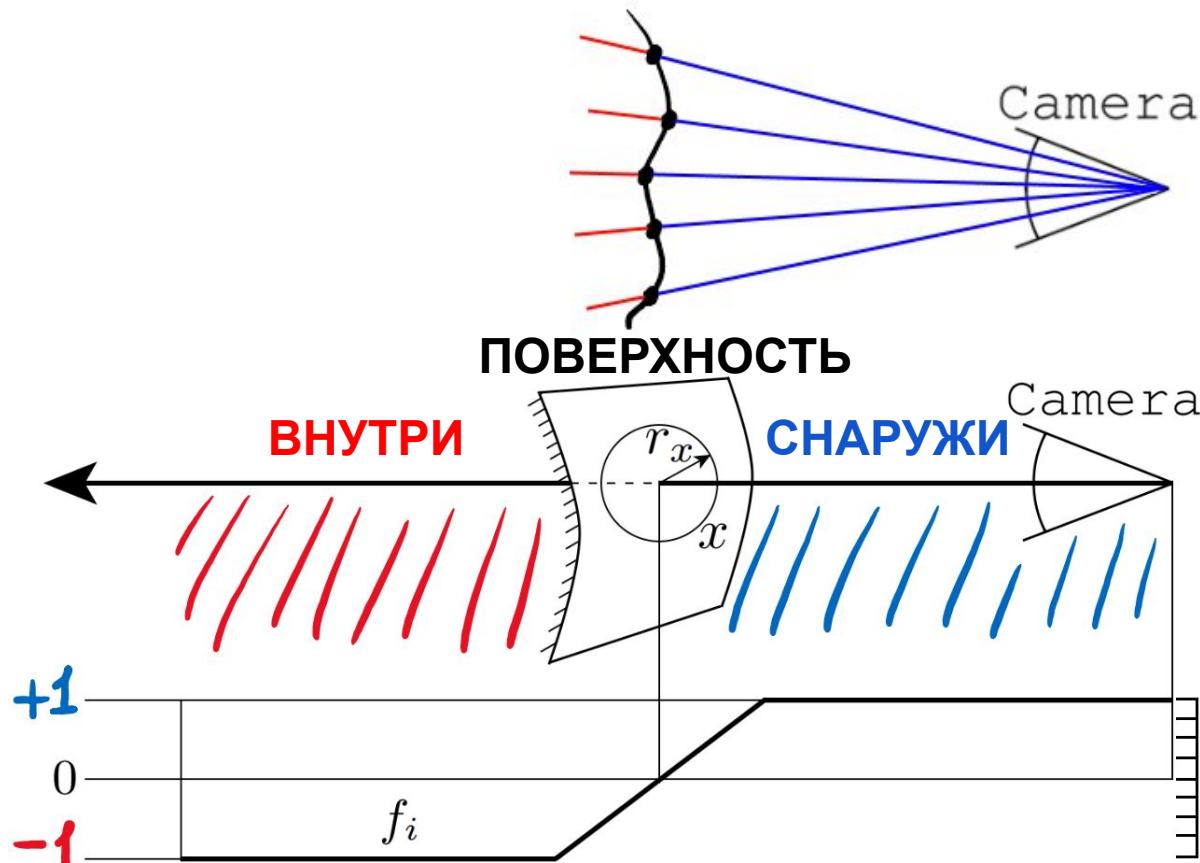
Карта глубины в виде индикаторного поля



Карта глубины в виде индикаторного поля



Карта глубины в виде индикаторного поля



f_i — индикаторное поле
соответствующее
карте глубины i ,
хранится в вокселях
гистограммами

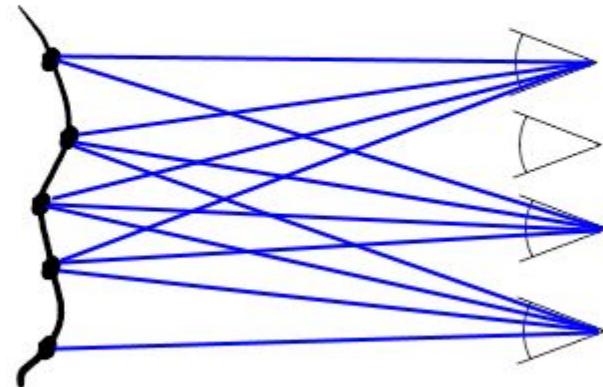
Численный метод на регулярной решетке [Zach2007]

- 1) Карты глубины преобразуем в гистограммы-голоса в каждом вокселе
- 2) Ищем оптимальное индикаторное поле (численными методами):
 - минимизируя полную вариацию
 - минимизируя отличия от индикаторных полей карт глубины
- 3) Извлекаем поверхность из индикаторного поля маршировкой кубов

Численный метод на регулярной решетке [Zach2007]

- 1) Карты глубины преобразуем в гистограммы-голоса в каждом вокселе
- 2) Ищем оптимальное индикаторное поле (численными методами):
 - минимизируя полную вариацию
 - минимизируя отличия от индикаторных полей карт глубины
- 3) Извлекаем поверхность из индикаторного поля маршировкой кубов

Очень сильное свойство - фильтрация шумов:

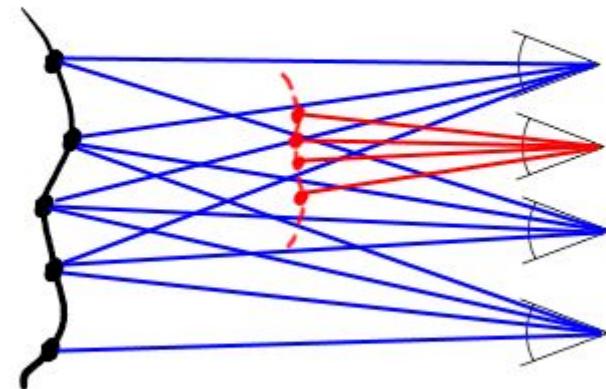


Численный метод на регулярной решетке [Zach2007]

- 1) Карты глубины преобразуем в гистограммы-голоса в каждом вокселе
- 2) Ищем оптимальное индикаторное поле (численными методами):
 - минимизируя полную вариацию
 - минимизируя отличия от индикаторных полей карт глубины
- 3) Извлекаем поверхность из индикаторного поля маршировкой кубов

Очень сильное свойство - фильтрация шумов:

$$E = \int_{\Omega} \left\{ |\nabla u| + \lambda \sum_{i \in \mathcal{I}(\vec{x})} w_i(\vec{x}) |u - f_i| \right\} d\vec{x}$$

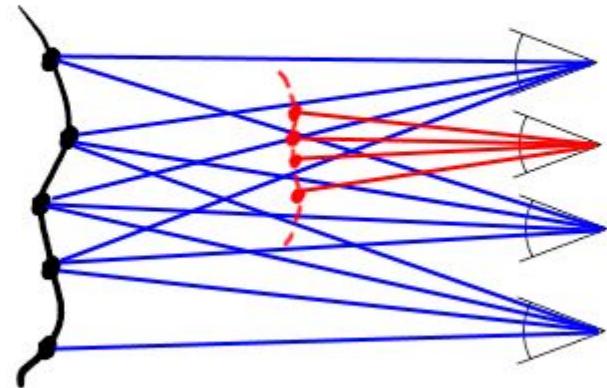


Численный метод на **регулярной решетке** [Zach2007]

- 1) Карты глубины преобразуем в гистограммы-голоса в каждом вокселе
- 2) Ищем оптимальное индикаторное поле (численными методами):
 - минимизируя полную вариацию
 - минимизируя отличия от индикаторных полей карт глубины
- 3) Извлекаем поверхность из индикаторного поля маршировкой кубов

Очень сильное свойство - фильтрация шумов:

$$E = \int_{\Omega} \left\{ |\nabla u| + \lambda \sum_{i \in \mathcal{I}(\vec{x})} w_i(\vec{x}) |u - f_i| \right\} d\vec{x}$$

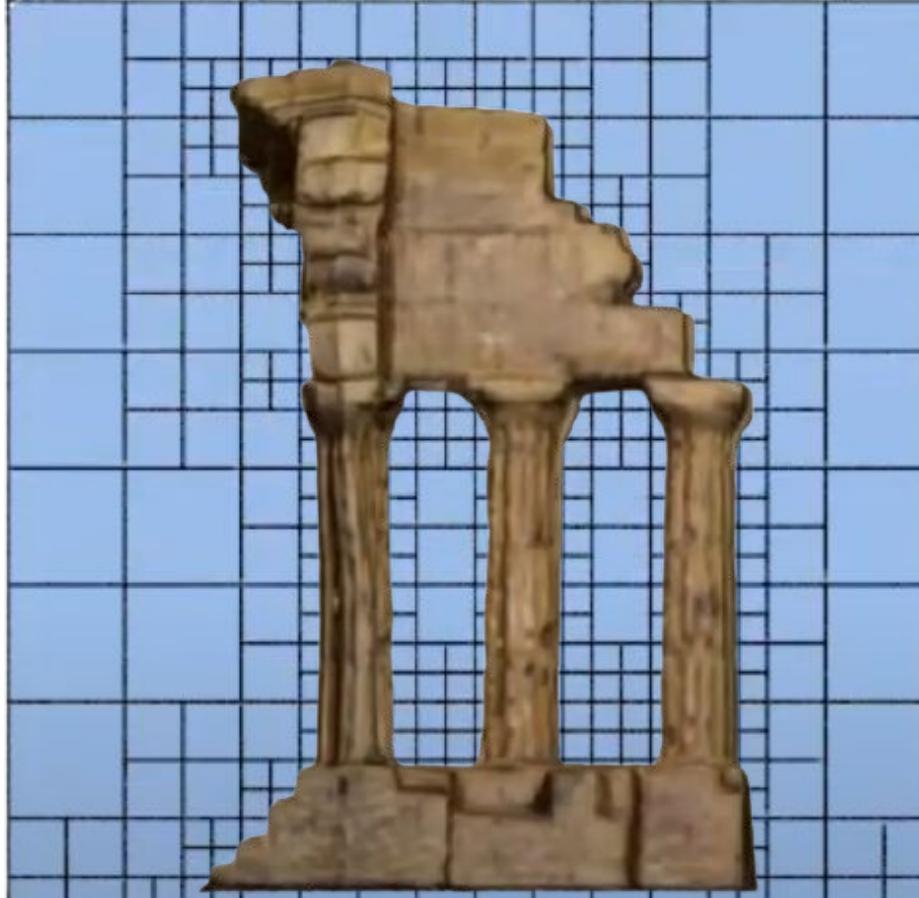


Численный метод на адаптивном октодереве [Ummenhofer2017]



Global, Dense Multiscale Reconstruction for a Billion Points, Ummenhofer et al., 2017

Численный метод на адаптивном октодереве [Ummenhofer2017]



Global, Dense Multiscale Reconstruction for a Billion Points, Ummenhofer et al., 2017

Численный метод на адаптивном октодереве [Ummenhofer2017]

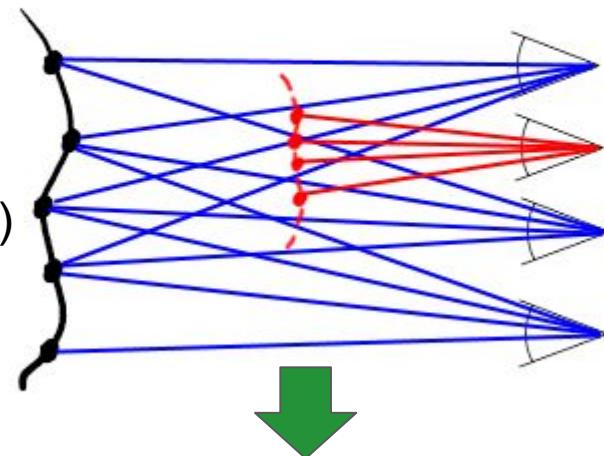
Отличия **GDMR**:

- Адаптивное октодерево (**сбалансированное**)
- Точки с нормалями вместо лучей видимости

Плюс: меньше оперативной памяти требуется

Минус: все еще **in-core**, на Breisach - **64 GB RAM**

Минус: потеряли свойство сильной фильтрации



Численный метод на адаптивном октодереве [Ummenhofer2017]

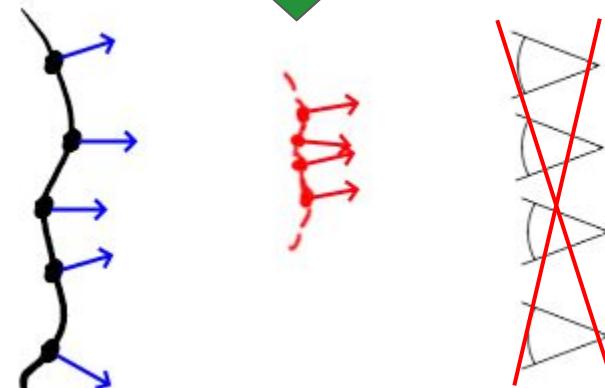
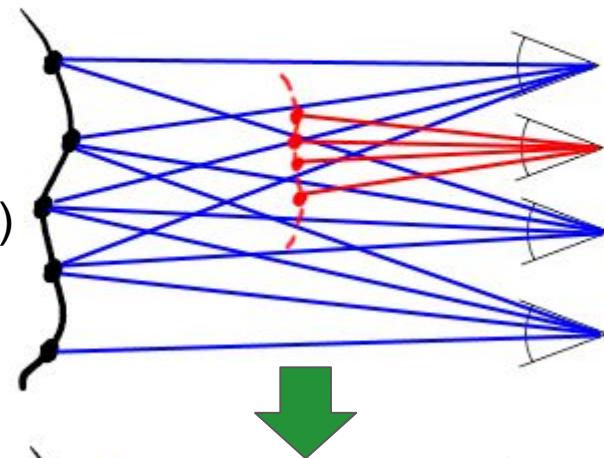
Отличия GDMR:

- Адаптивное октодерево (**сбалансированное**)
- Точки с нормалями вместо лучей видимости

Плюс: меньше оперативной памяти требуется

Минус: все еще **in-core**, на Breisach - **64 GB RAM**

Минус: потеряли свойство сильной фильтрации

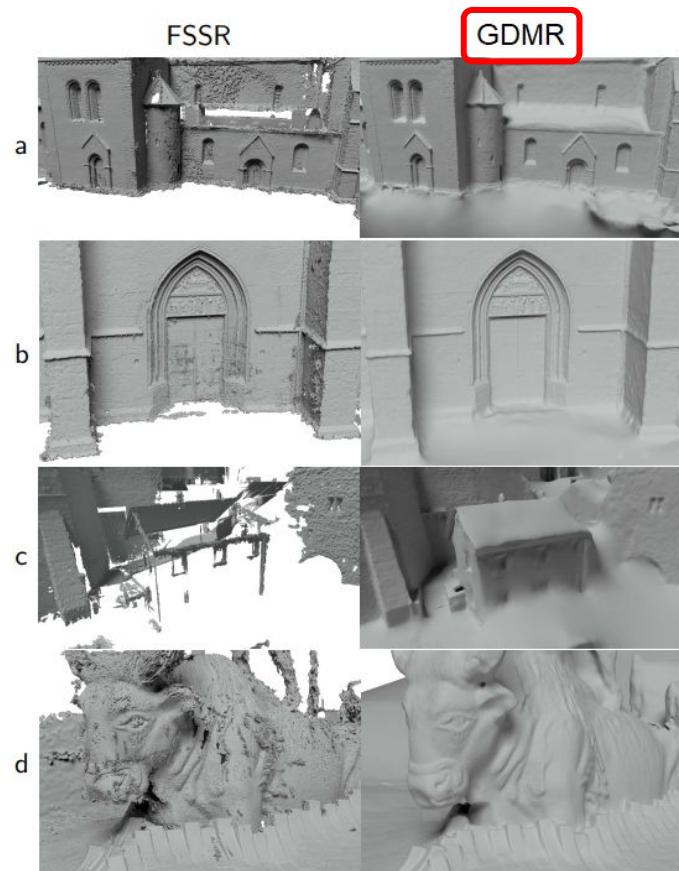


Численный метод на адаптивном октодереве [Ummenhofer2017]

Breisach: 2111 фотографий, 1.5 миллиарда точек

	Duration	17%
Octree generation	Density estimation	74.6 min
	Balancing	7.9 min
	Histograms	782.4 min
Surface comp.	Dual grid generation	19.9 min
	Energy minimization	3678.0 min
	Dual contouring	16.3 min
Other		23.5 min
Total		4602.9 min

Table 1. Runtime breakdown for the Breisach data set.



Мечта

На компьютере с 16 GB RAM реконструировать объект любой сложности.

Мечта

На компьютере с 16 GB RAM реконструировать объект любой сложности.

- **Out-of-Core** - в любой момент в памяти ограниченный объем данных

Мечта

На компьютере с 16 GB RAM реконструировать объект любой сложности.

- **Out-of-Core** - в любой момент в памяти ограниченный объем данных
- Преобразование карт глубины в гистограммы (голосование) на 

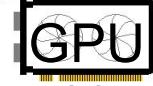
Мечта

На компьютере с 16 GB RAM реконструировать объект любой сложности.

- **Out-of-Core** - в любой момент в памяти ограниченный объем данных
- Преобразование карт глубины в гистограммы (голосование) на 
- Минимизация численным методом на 

Мечта

На компьютере с 16 GB RAM реконструировать объект любой сложности.

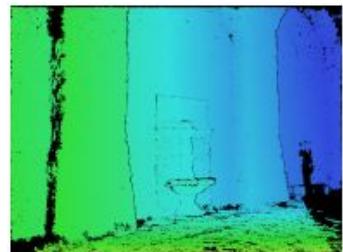
- **Out-of-Core** - в любой момент в памяти ограниченный объем данных
- Преобразование карт глубины в гистограммы (голосование) на 
- Минимизация численным методом на 
- Фильтрация шумов на базе лучей видимости (просто точки - не подходят)

Мечта

На компьютере с 16 GB RAM реконструировать объект любой сложности.

- **Out-of-Core** - в любой момент в памяти ограниченный объем данных
- Преобразование карт глубины в гистограммы (голосование) на 
- Минимизация численным методом на 
- Фильтрация шумов на базе лучей видимости (просто точки - не подходят)
- Возможность ускорять за счет вычисления на кластере

1) Out-of-Core построение октодерева

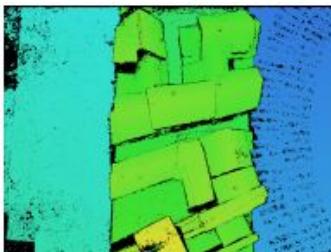


Octree #1



96-bit

Morton Codes



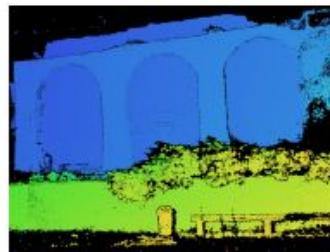
Octree #2



96-bit

Morton Codes

...



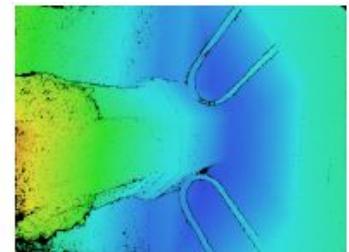
Octree #N-1



96-bit

Morton Codes

...



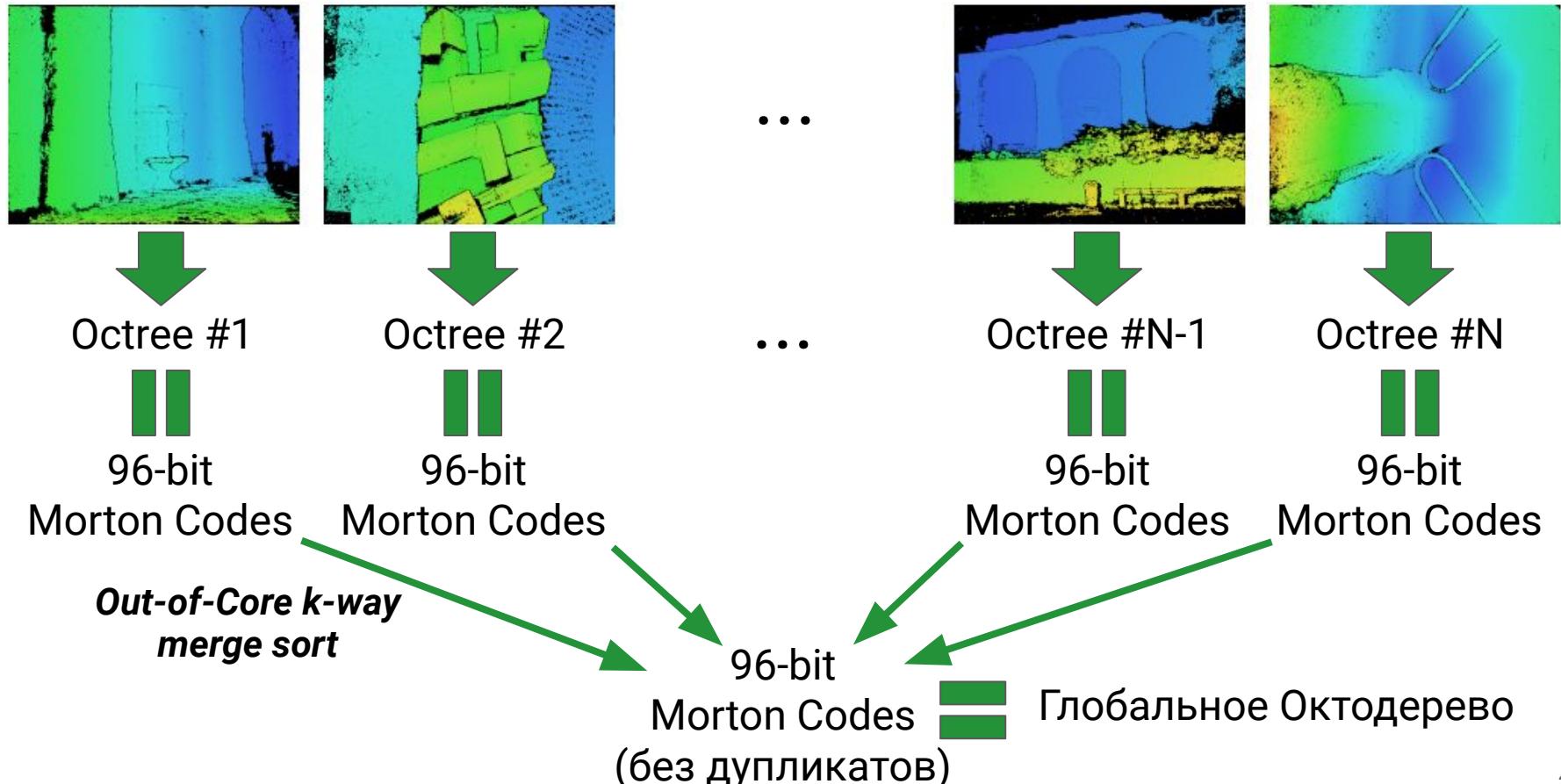
Octree #N



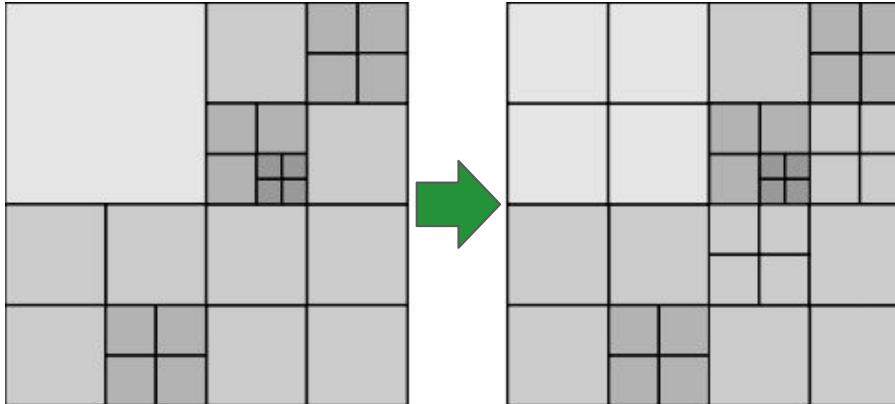
96-bit

Morton Codes

1) Out-of-Core построение октодерева



2) 2:1 балансировка октодерева

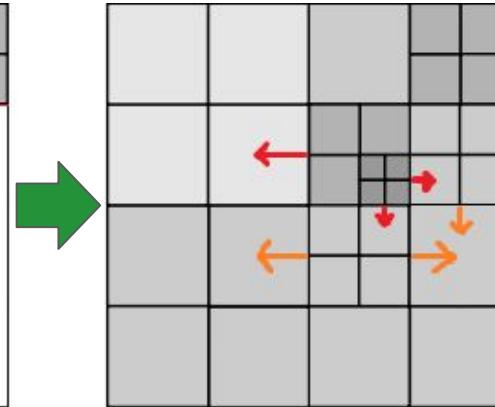
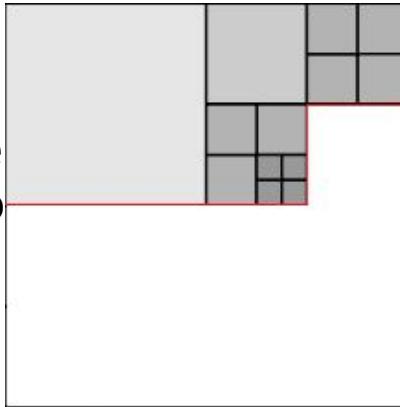


Глобальное
Октодерево:
один огромный файл
с отсортированными
Morton Codes

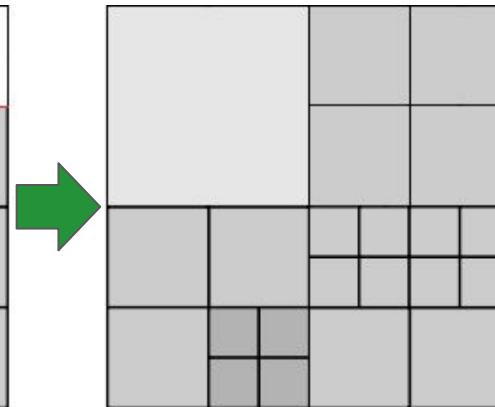
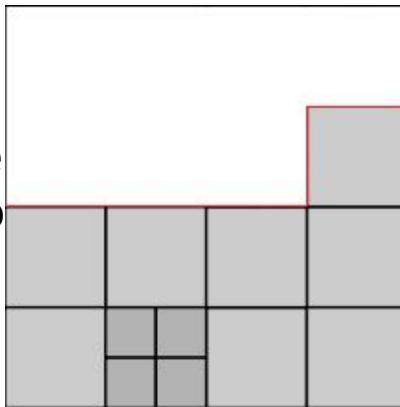
Балансированное
Глобальное
Октодерево

2) Out-of-Core 2:1 балансировка октодерева

Глобальное
Октодерево
(part #1)



Глобальное
Октодерево
(part #2)

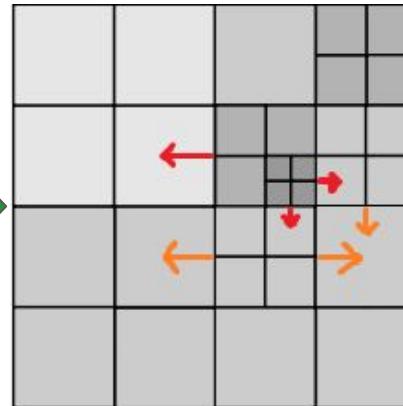
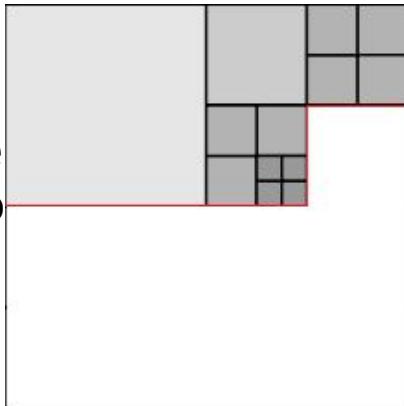


IO-friendly:

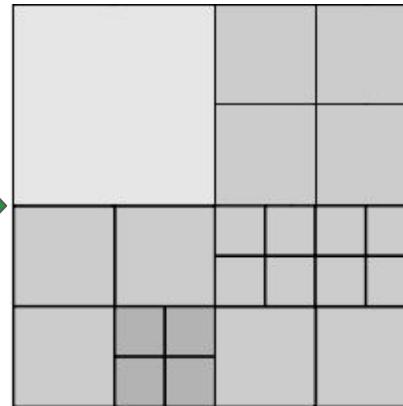
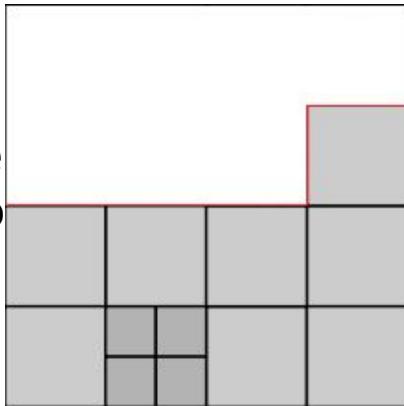
Исходные кубы на диске
последовательны
благодаря сортировке
Morton Codes

2) Out-of-Core 2:1 балансировка октодерева

Глобальное
Октодерево
(part #1)

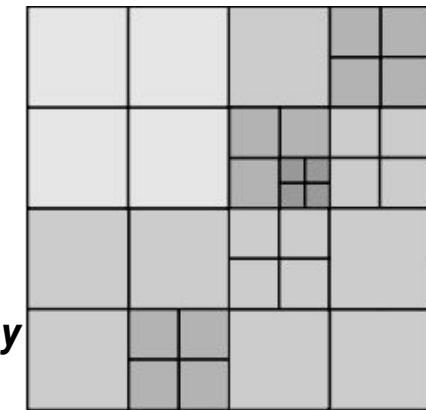


Глобальное
Октодерево
(part #2)

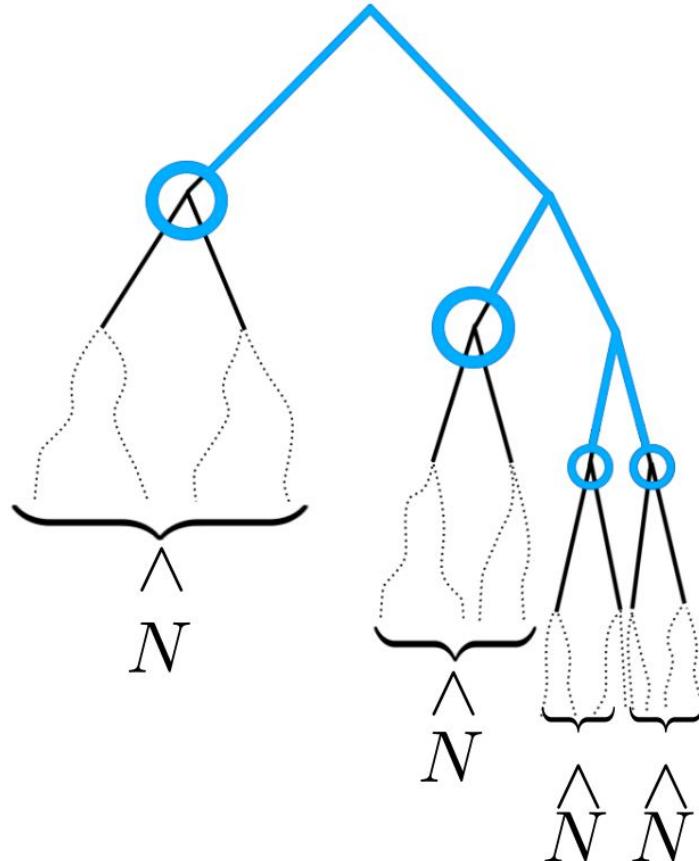


*Out-of-Core k-way
merge sort*

Балансированное
Глобальное
Октодерево

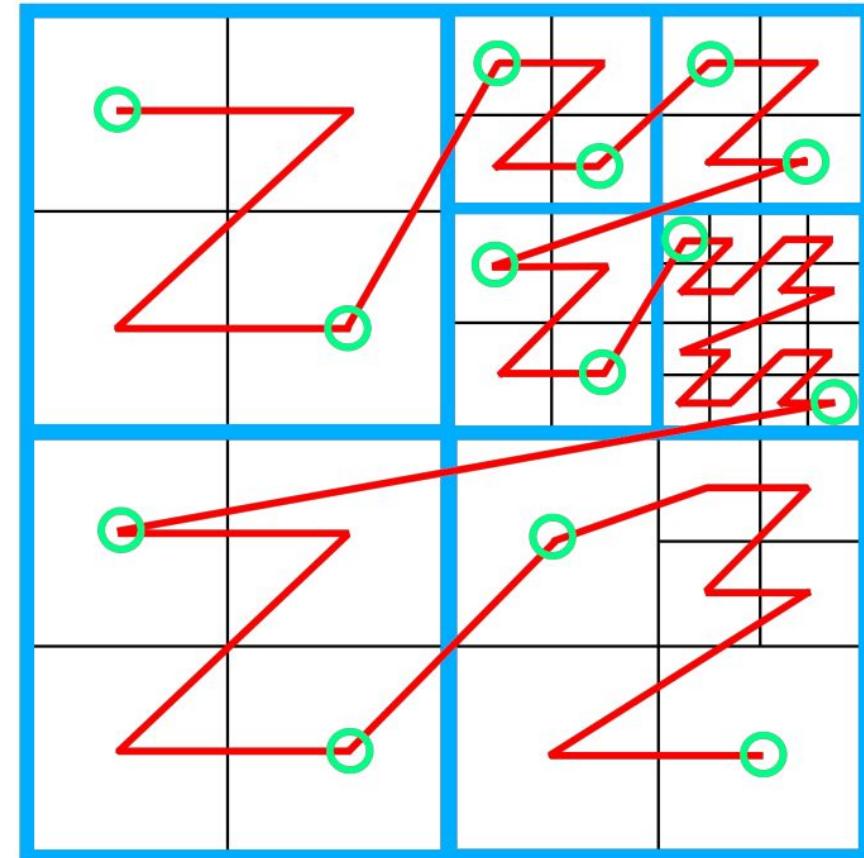
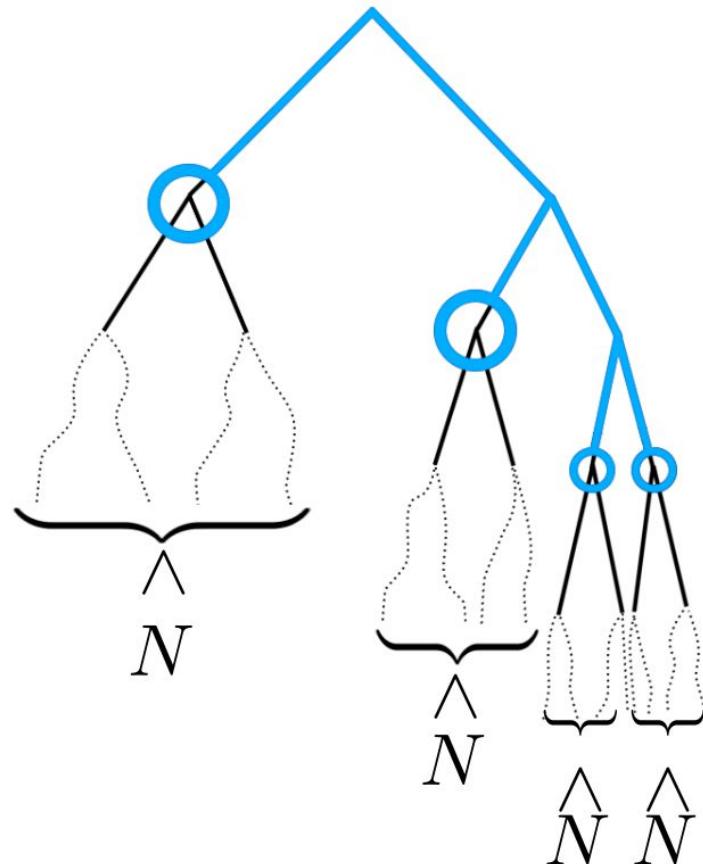


3) Построение верхушки дерева

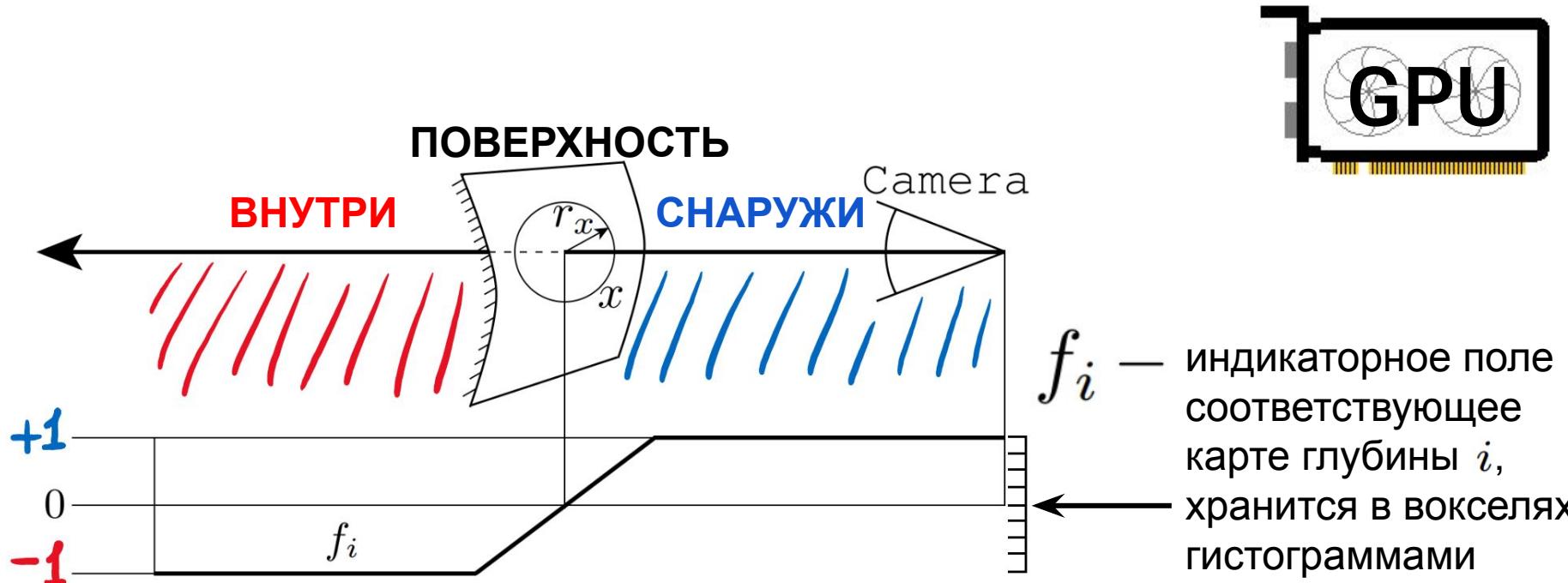


IO-friendly:
Исходные кубы на диске
последовательны
благодаря сортировке
Morton Codes

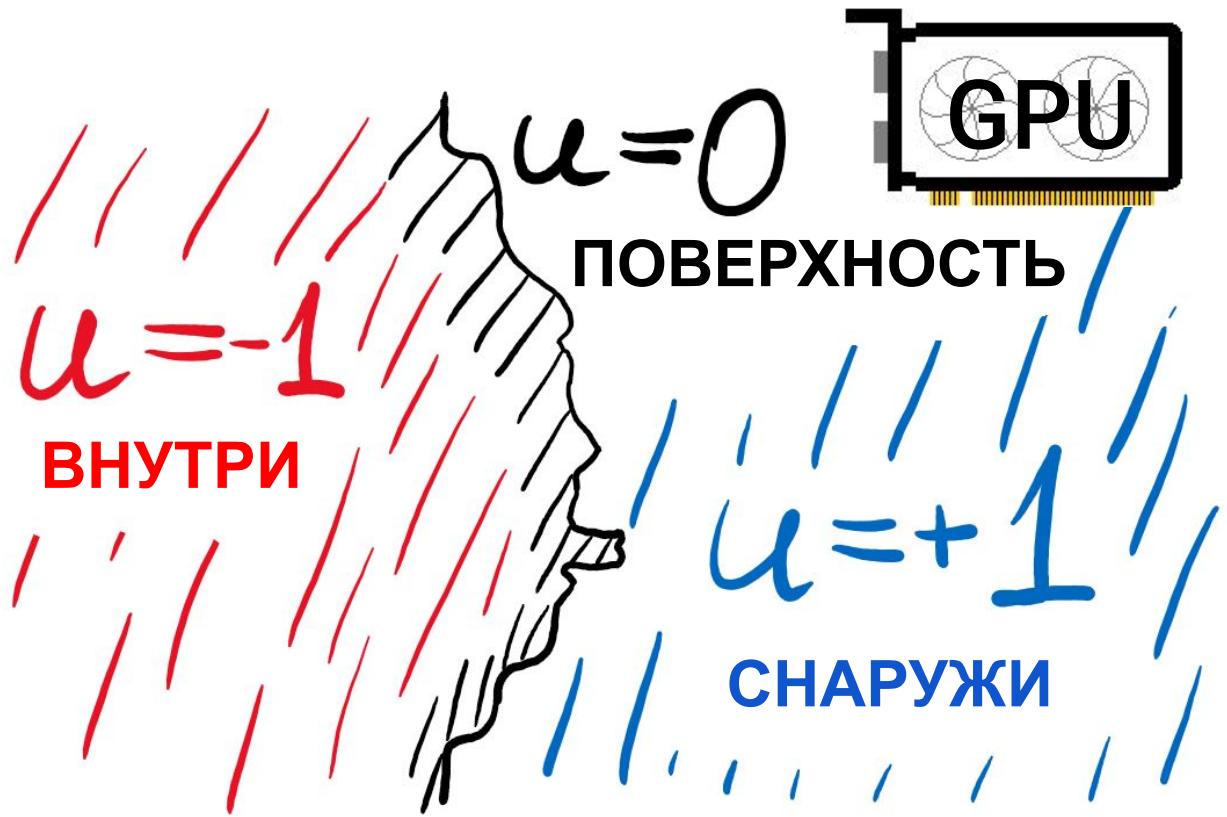
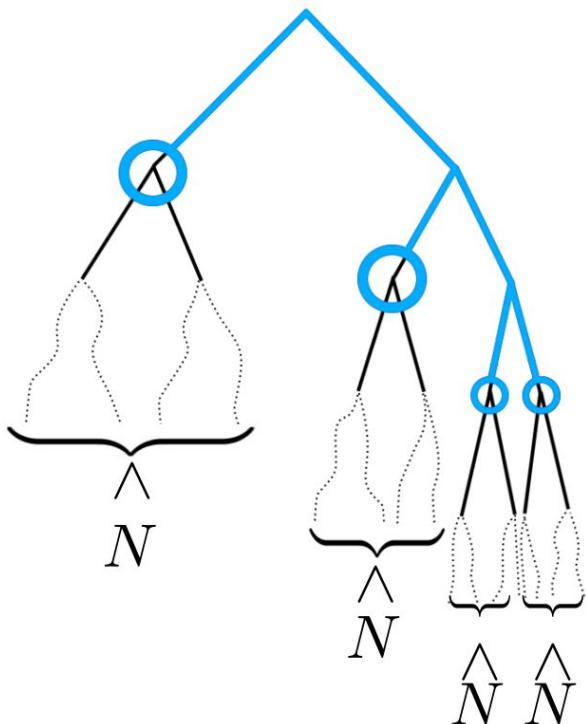
3) Построение верхушки дерева



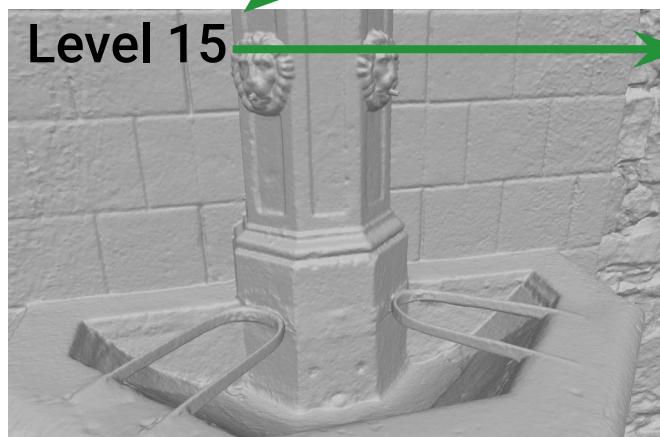
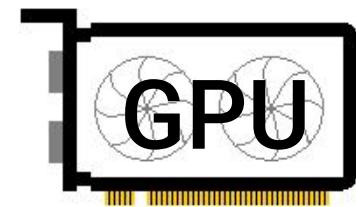
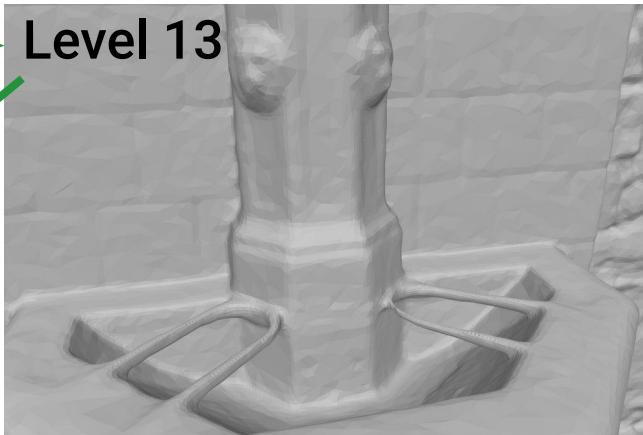
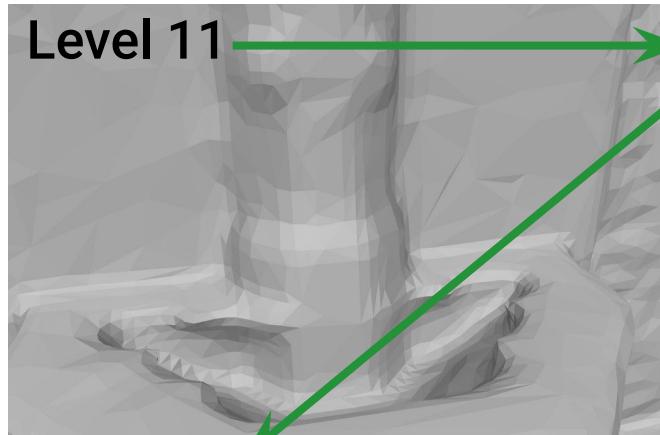
4) Преобразование карт глубины в гистограммы вокселей



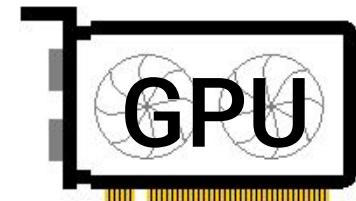
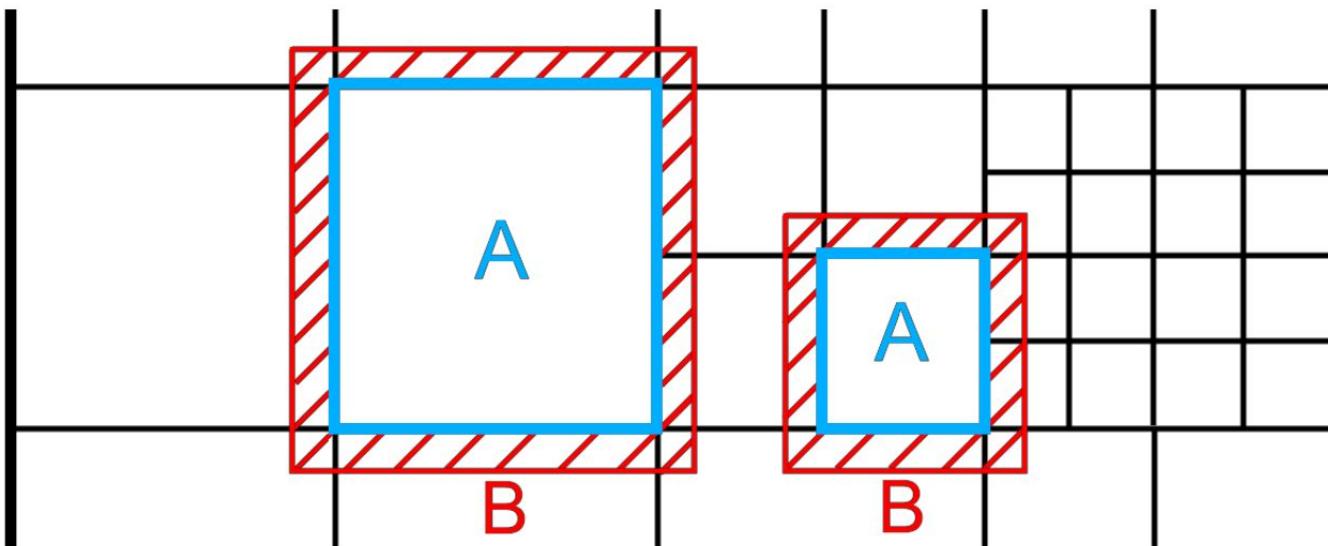
5) Оптимизация индикаторного поля, минимизация TGV



5) Оптимизация индикаторного поля: Coarse-to-Fine



5) Оптимизация индикаторного поля: заморозка границ



6) Извлечение поверхности

Индикаторное поле



Marching cubes



Полигональная модель



Свойства алгоритма

Качественные свойства:

- **Scale-diverse** (адаптивное разрешение)
- Сильные свойства фильтрации шумов (благодаря лучам видимости)
- Бесшовная поверхность

Свойства алгоритма

Качественные свойства:

- **Scale-diverse** (адаптивное разрешение)
- Сильные свойства фильтрации шумов (благодаря лучам видимости)
- Бесшовная поверхность

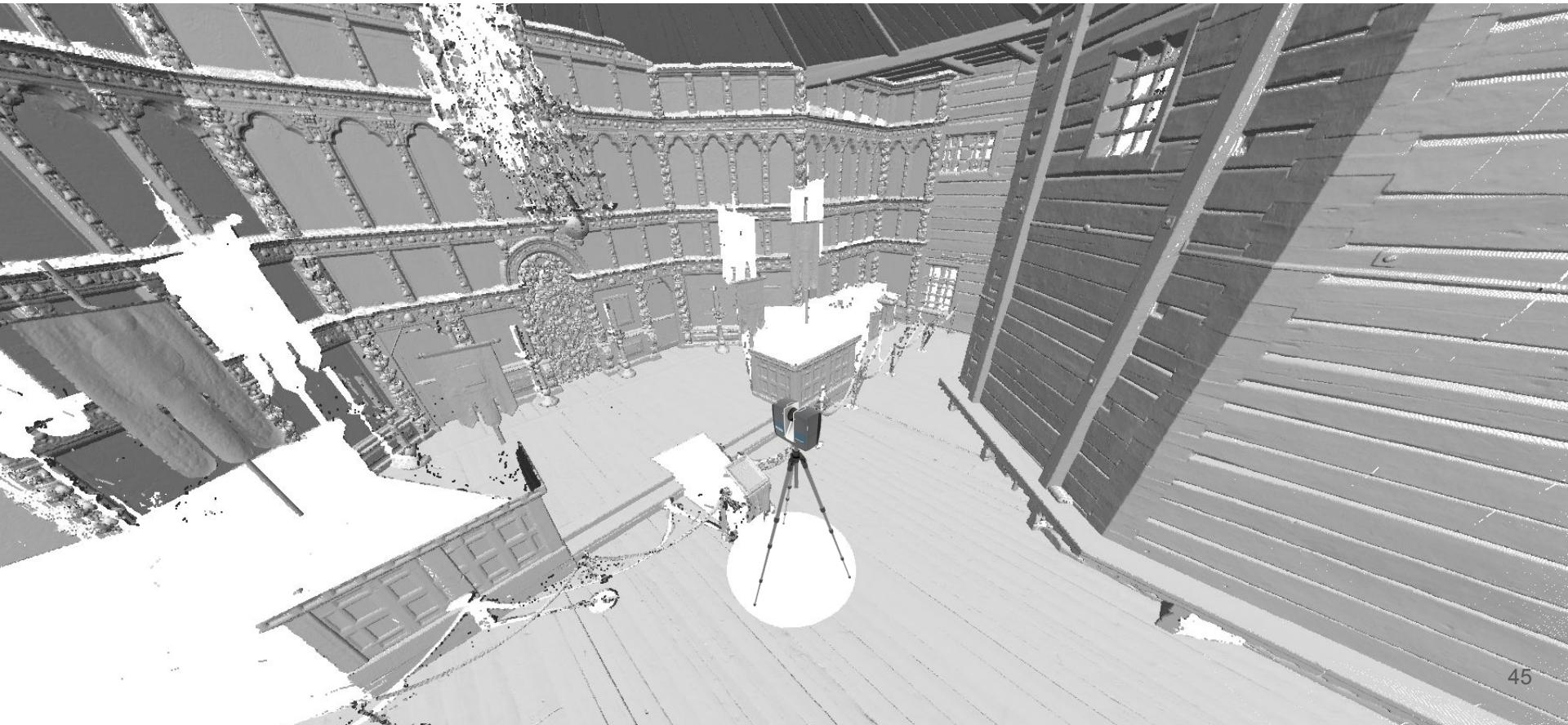
Практические свойства:

- **Out-of-Core** (ограниченные требования к RAM)
- Быстрый (ускорение на **GPU** & **IO-friendly**)
- Можно запускать на **клUSTERе**
- Поддерживает **LIDAR** сканы и спутниковые снимки

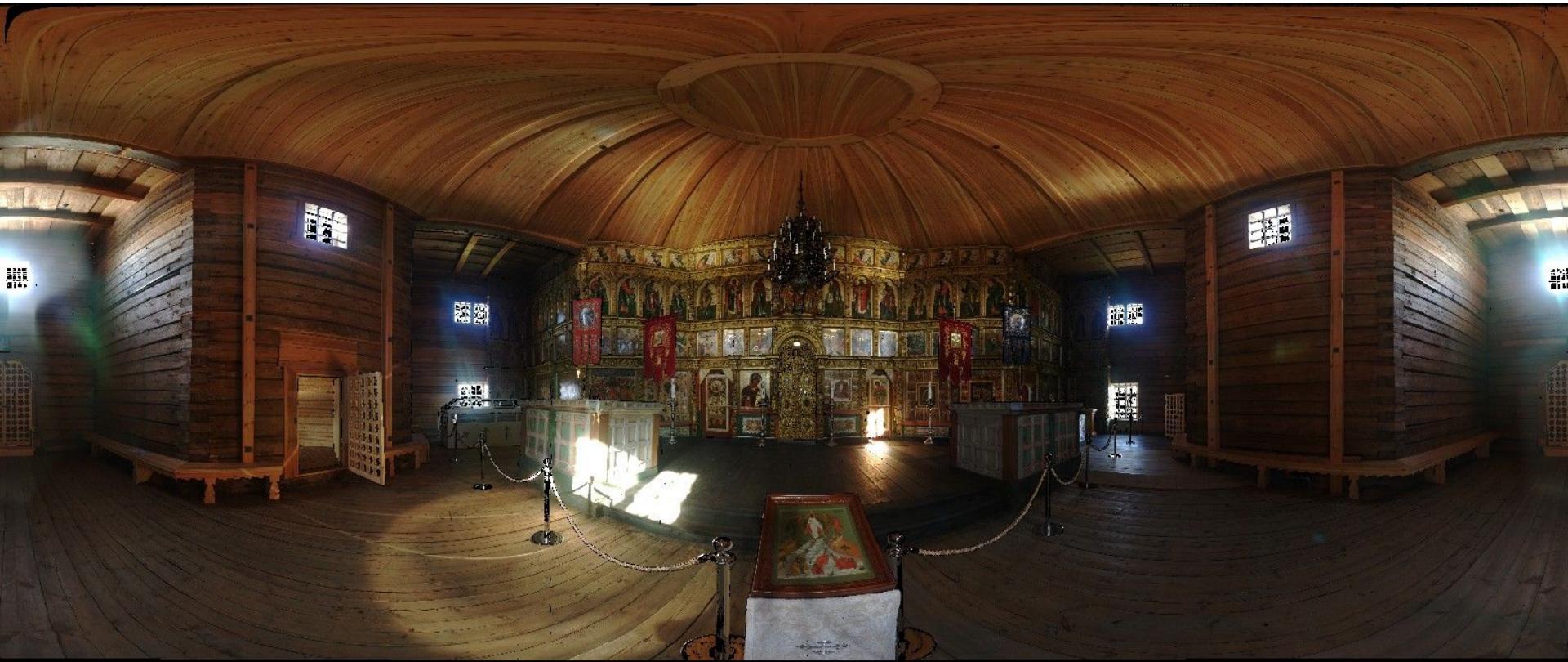
Входные данные: Terrestrial LIDAR



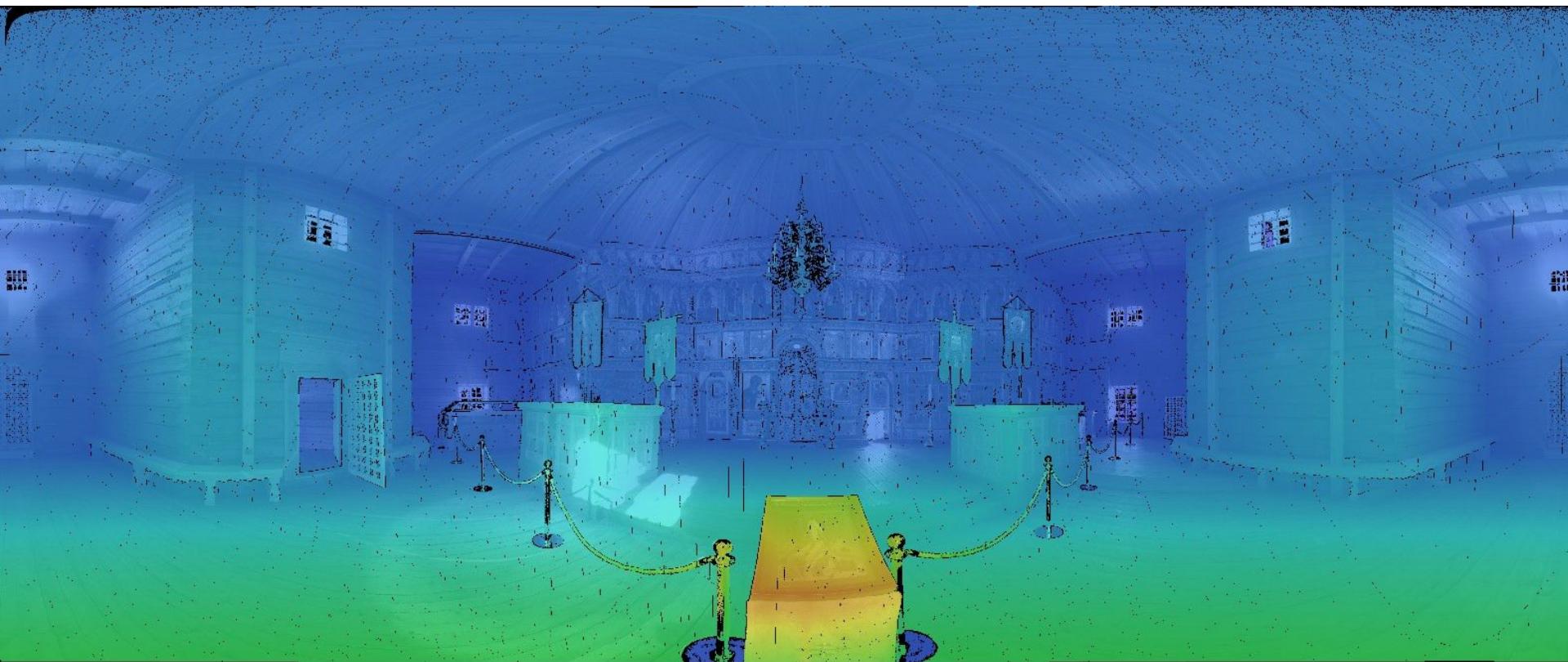
Входные данные: Terrestrial LIDAR

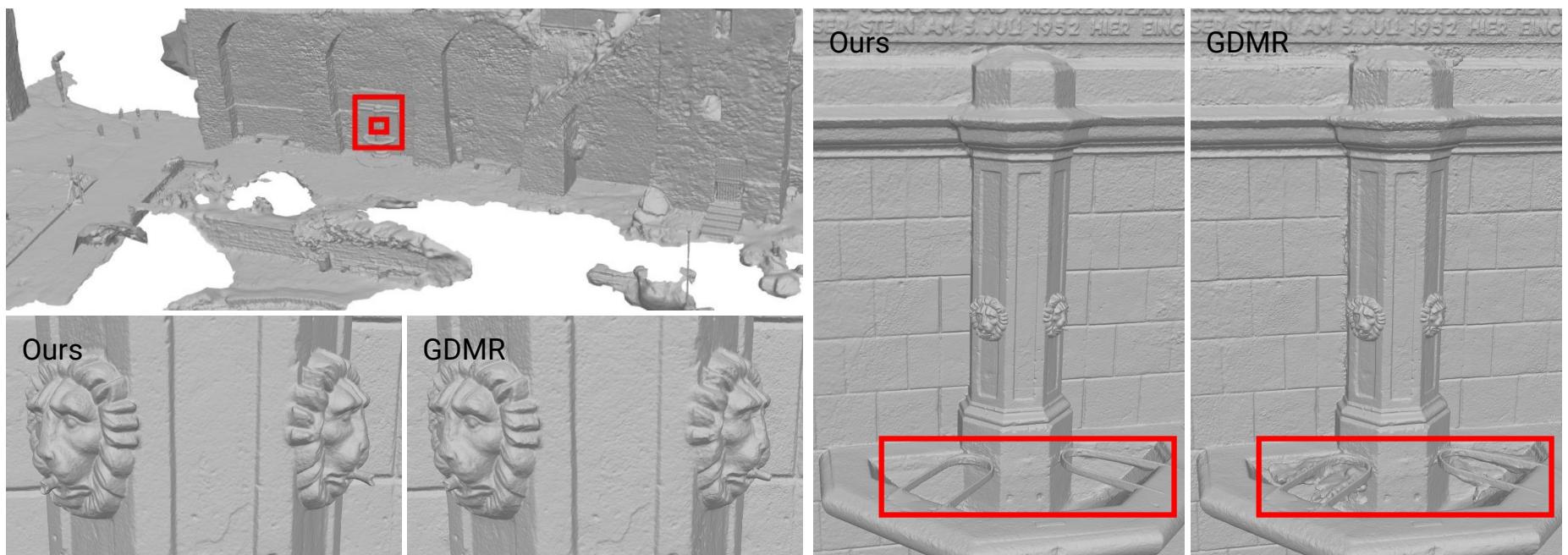


Terrestrial LIDAR = карта глубины сферической камеры



Terrestrial LIDAR = карта глубины сферической камеры

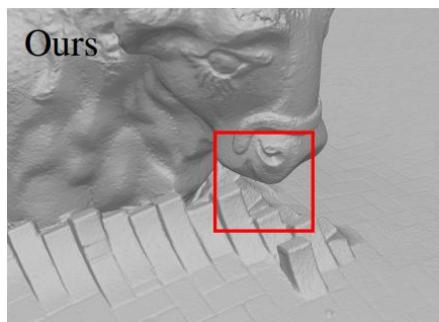




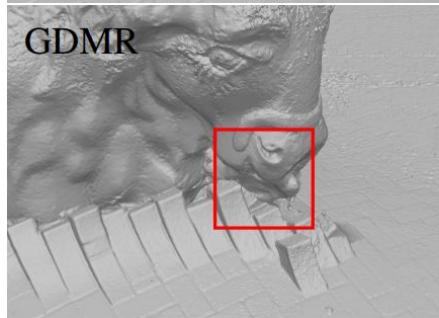
Dataset name	Input data	GDMR Peak RAM	GDMR time	Our Peak RAM	Our time	SSR Peak RAM	SSR time
Citywall	564 depth maps	75 GB	19 h	13.17 GB	63 min	32*8.9 GB	58 h

x19 faster

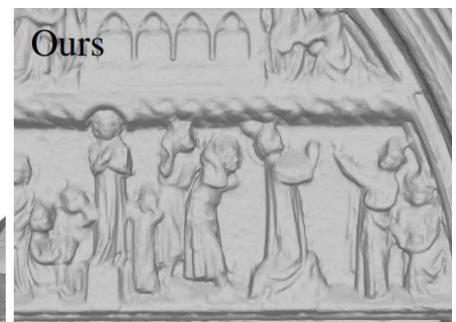
Ours



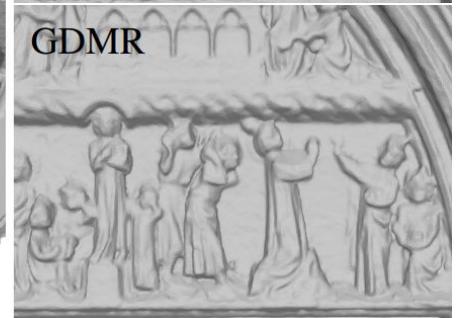
GDMR



Ours

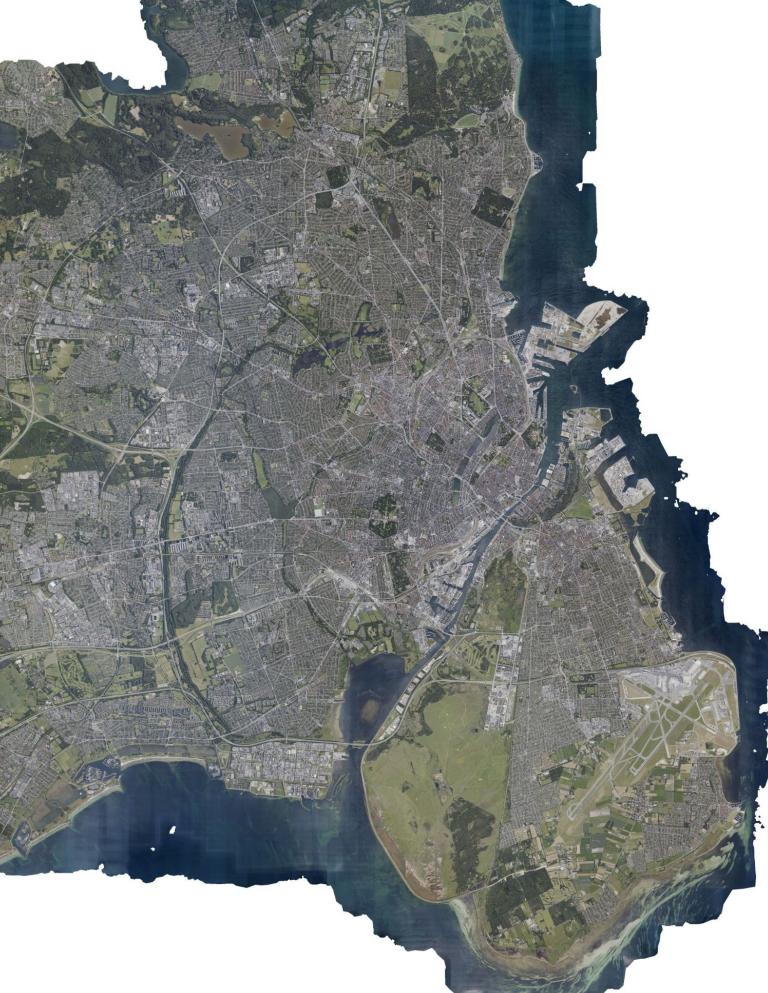


GDMR



Dataset name	Input data	GDMR Peak RAM	GDMR time	Our Peak RAM	Our time	SSR Peak RAM	SSR time
Citywall	564 depth maps	75 GB	19 h	13.17 GB	63 min	32*8.9 GB	58 h
Breisach	2111 depth maps	64 GB	76 h	10.07 GB	260 min	N/A	N/A

x17 faster



Building Copenhagen in a Day*

- 425 km²
- 27472 фотографии (566 GB, jpeg)
- 28 миллиардов “точек”
- 7.5 миллиардов треугольников
- пиковая RAM: 14 GB
- *на кластере за 29 часов
(7 компьютеров по 8 ядер + GTX 1080)



Figure 4. Frederiksberg Forsyning A/S closeup. Note that both pipes were reconstructed well.



Figure 7. Airport closeup.

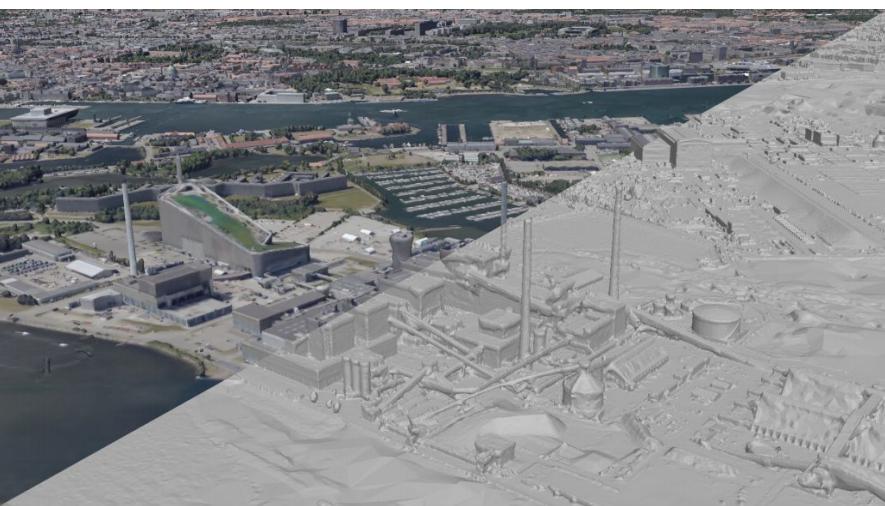


Figure 8. Høfor Amagerverket closeup.

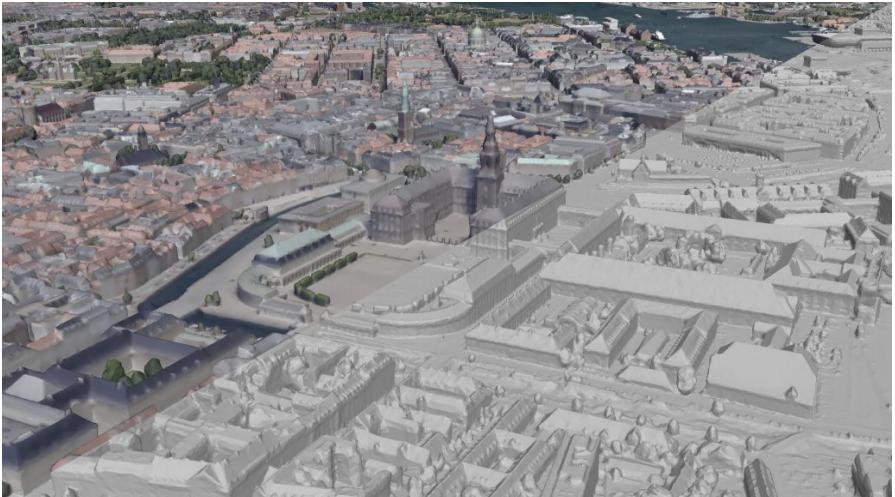


Figure 9. Christiansborg Palace closeup.

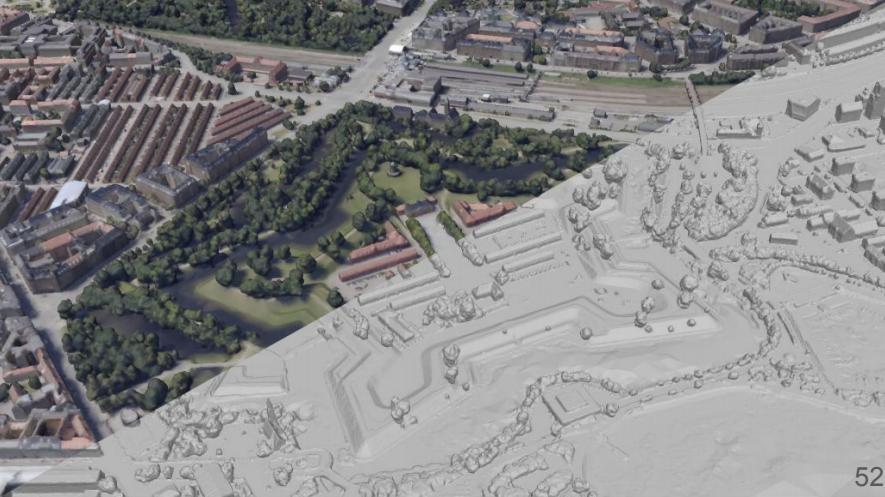


Figure 10. Kastellet closeup.

Dataset name	Input data	Initial cubes	Faces after marching cubes	Peak RAM (GB)	Processing time
Citywall [10]	564 depth maps	1205 mil	135 mil	13.17	63 min
Breisach [27]	2111 depth maps	2642 mil	558 mil	10.07	260 min
Tomb of Tu Duc (LIDAR) [6]	42 LIDAR scans	661 mil	672 mil	10.05	160 min
Palacio Tschudi [7]	13703 depth maps	16 billion	3159 mil	16.75	1213 min
Copenhagen city [8]	27472 depth maps	28 billion	7490 mil	13.35	1758 min

Вопросы?



Agisoft

polarnick@agisoft.com

Николай Полярный