

Регистрация облаков точек

Фотограмметрия. Лекция 7

Александрина Стрельцова
aleksandrina.strelczova@gmail.com

7 марта 2023 г.

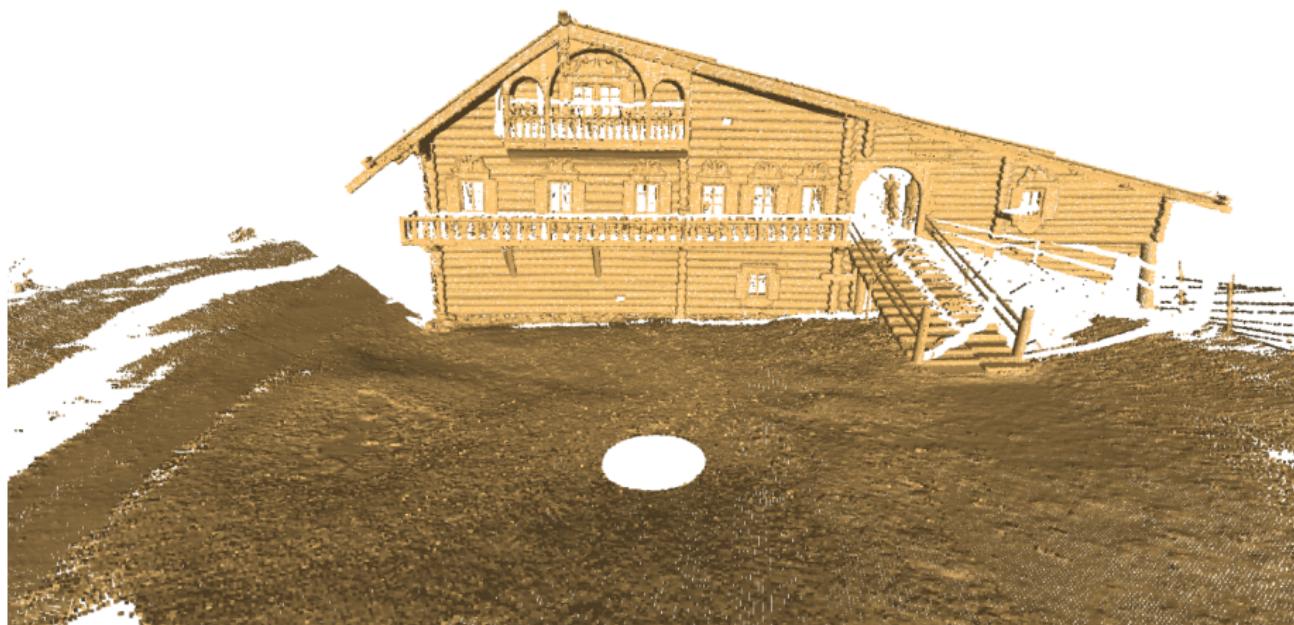
Облако точек. RGB-D камера



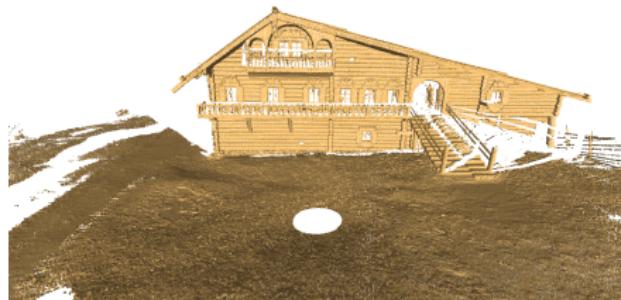
<https://www.eth3d.net>

Облако точек. LiDAR

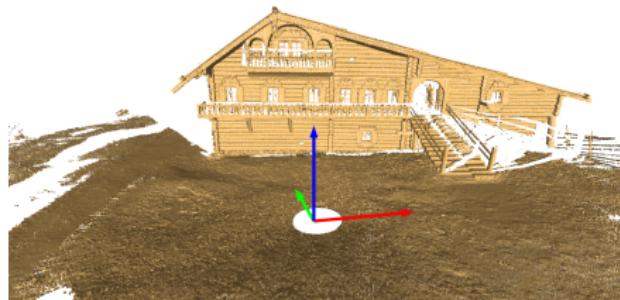
LiDAR - Light Detection and Ranging



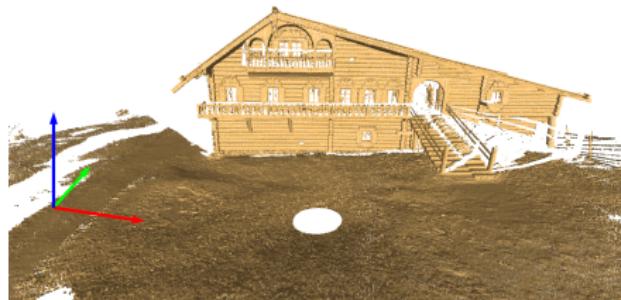
Постановка задачи



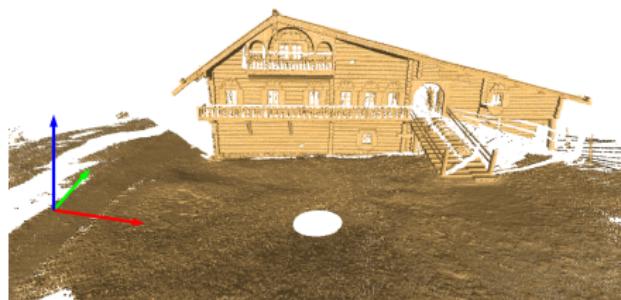
Постановка задачи



Постановка задачи

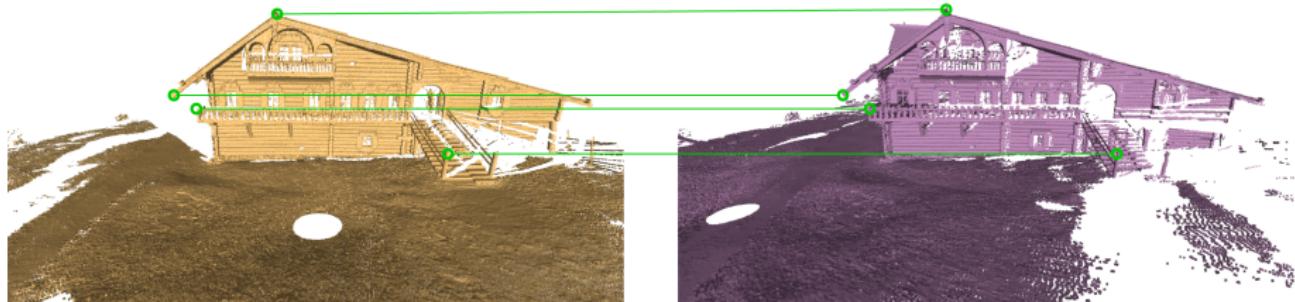


Постановка задачи



Постановка задачи

- Облако точек: $X = \{x_i\}, X \subset \mathbb{R}^3$
- Параметры преобразования: $R \in SO(3)$ и $t \in \mathbb{R}^3$
- $T \in SE(3)$



Как вычислить преобразование, если известны соответствия?

$$e_i = x_{t,i} - Rx_{s,i} - t$$

$$F(R, t) = \sum_{i=1}^n \|e_i\|^2$$

Вычисление параметров преобразования

- $F(R, t) = \sum_{i=1}^n \|x_{t,i} - Rx_{s,i} - t\|^2 \rightarrow \min$
 $\nabla_t F = \sum_{i=1}^n -2(x_{t,i} - Rx_{s,i} - t)^T = 0$
 $n t = \sum_{i=1}^n (x_{t,i} - Rx_{s,i})$
 $t = \frac{1}{n} \sum_{i=1}^n x_{t,i} - R \frac{1}{n} \sum_{i=1}^n x_{s,i}$

Вычисление параметров преобразования

- $F(R, t) = \sum_{i=1}^n \|x_{t,i} - Rx_{s,i} - t\|^2 \rightarrow \min$

- $t = \bar{x}_t - R\bar{x}_s$

- $x'_{t,i} = x_{t,i} - \bar{x}_t, x'_{s,i} = x_{s,i} - \bar{x}_s$

$$\bar{x}_t = \frac{1}{n} \sum_{i=1}^n x_{t,i} \quad \bar{x}_s = \frac{1}{n} \sum_{i=1}^n x_{s,i}$$
$$F(R) = \sum_{i=1}^n \|x_{t,i} - Rx_{s,i} - \bar{x}_t - R\bar{x}_s\|^2 =$$
$$= \sum_{i=1}^n \|x'_{t,i} - Rx'_{s,i}\|^2$$

Вычисление параметров преобразования

- $F(R, t) = \sum_{i=1}^n \|x_{t,i} - Rx_{s,i} - t\|^2 \rightarrow \min$

- $t = \bar{x}_t - R\bar{x}_s$

- $x'_{t,i} = x_{t,i} - \bar{x}_t, x'_{s,i} = x_{s,i} - \bar{x}_s$

- $F(R) = \sum_{i=1}^n \|x'_{t,i} - Rx'_{s,i}\|^2 \rightarrow \min$

$$x'_t = \underbrace{(0 \ 0 \ \dots \ 0)}_n \ 3$$

$$\sum_{i=1}^n x'^T_{t,i} R x'_{s,i} = \text{tr}(x'^T_t R x'_s) = \text{tr}(x'^T_s x'^T_t R) = \text{tr}(UDV^T R) = \text{tr}(D \ V^T R)$$
$$R = VU^T$$

$$\sum_{i=1}^n \|x'_{t,i}\|^2 - 2 \sum_{i=1}^n x'^T_{t,i} R x'_{s,i} + \sum_{i=1}^n \|x'_{s,i}\|^2 \rightarrow \min$$

const min const

Вычисление параметров преобразования

- $F(R, t) = \sum_{i=1}^n \|x_{t,i} - Rx_{s,i} - t\|^2 \rightarrow \min$
- $t = \bar{x}_t - R\bar{x}_s$
- $x'_{t,i} = x_{t,i} - \bar{x}_t, x'_{s,i} = x_{s,i} - \bar{x}_s$
- $F(R) = \sum_{i=1}^n \|x'_{t,i} - Rx'_{s,i}\|^2 \rightarrow \min$
- $X'_s X'^T_t = UDV^T$

Вычисление параметров преобразования

- $F(R, t) = \sum_{i=1}^n \|x_{t,i} - Rx_{s,i} - t\|^2 \rightarrow \min$
- $t = \bar{x}_t - R\bar{x}_s$
- $x'_{t,i} = x_{t,i} - \bar{x}_t, x'_{s,i} = x_{s,i} - \bar{x}_s$
- $F(R) = \sum_{i=1}^n \|x'_{t,i} - Rx'_{s,i}\|^2 \rightarrow \min$
- $X'_s X'^T_t = UDV^T$
- $R = VU^T$

Вычисление параметров преобразования

- ① Вычислить центроиды облаков \bar{x}_s и \bar{x}_t
- ② Сместить облака на \bar{x}_s и \bar{x}_t соответственно
- ③ Вычислить сингулярное разложение матрицы $X'_s X'^T_t$
- ④ Вычислить параметры преобразования $R = VU^T$ и
 $t = \bar{x}_t - R\bar{x}_s$

Декриптор. Локальная система координат

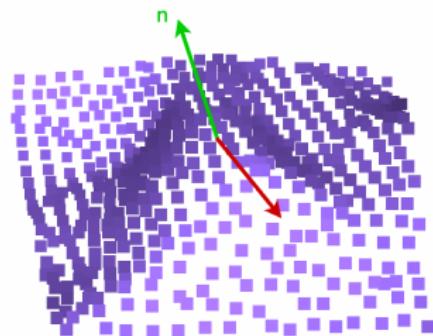
- ① Инвариантность к повороту
- ② Инвариантность к масштабу

Декриптор. Локальная система координат

- ① Инвариантность к повороту
- ② Инвариантность к масштабу

Дескриптор. Локальная система координат

- ① Инвариантность к повороту
- ② Инвариантность к масштабу

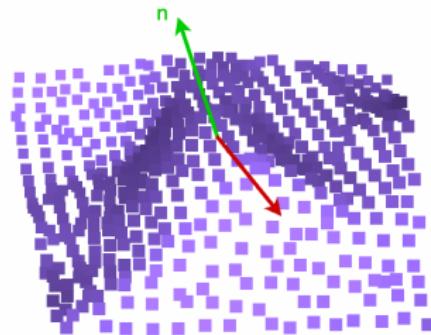


нормаль + кривизна

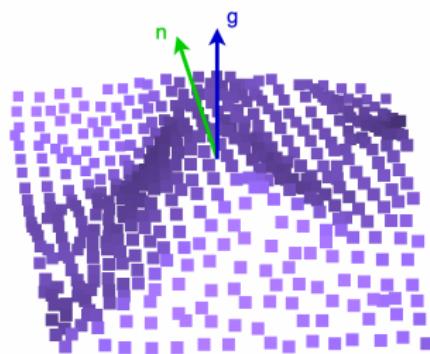
Дескриптор. Локальная система координат

- ① Инвариантность к повороту
- ② Инвариантность к масштабу

SIFT:



нормаль + кривизна

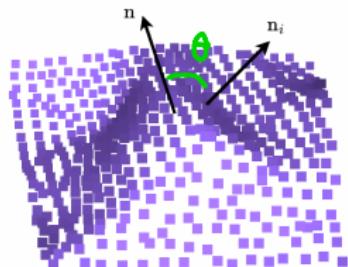


нормаль + гравитация

Дескриптор. SHOT

- θ - угол между нормалями n и n_i
- $\cos \theta = n^T n_i$

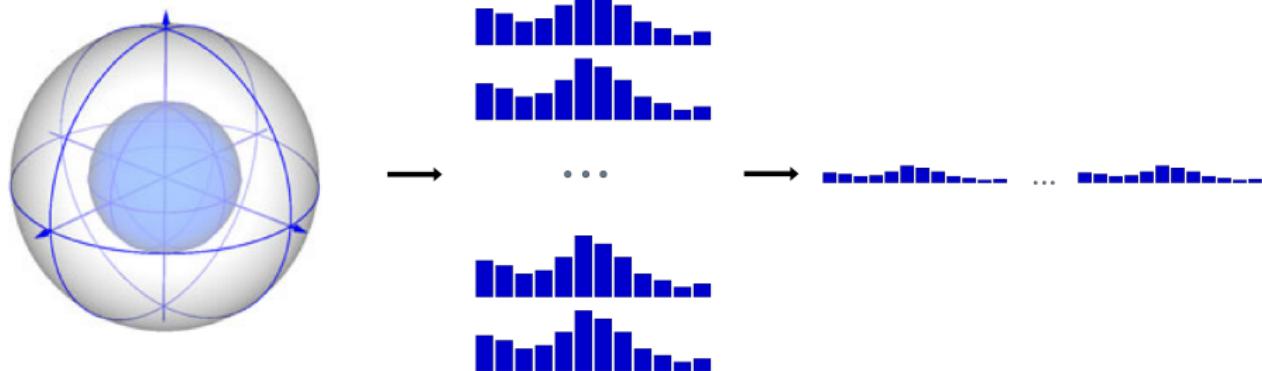
В качестве значений гистограммы используется именно $\cos \theta$, а не θ .



статья: Unique Signatures of Histograms for Local Surface Description, 2010

Декриптор. SHOT

- 1 Разделим окрестность на области
- 2 Для каждой области построим гистограмму значений $\cos \theta$



статья: Unique Signatures of Histograms for Local Surface Description, 2010
дескрипторы с примерами использования: <https://robotica.unileon.es>

Детектор ключевых точек

Какие точки нас интересуют?

- Радиус кривизны в точке не должен быть большим

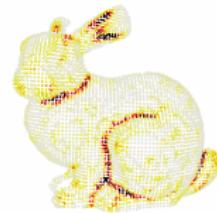
$$r = \infty$$



Детектор ключевых точек

Какие точки нас интересуют?

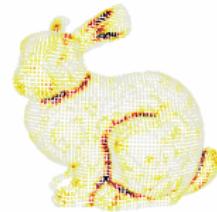
- Радиус кривизны в точке не должен быть большим
- Направления кривизны должны задаваться однозначно
- Точка не должна "скользить" по направлению кривизны



Детектор ключевых точек

Какие точки нас интересуют?

- Радиус кривизны в точке не должен быть большим
- Направления кривизны должны задаваться однозначно
- Точка не должна "скользить" по направлению кривизны



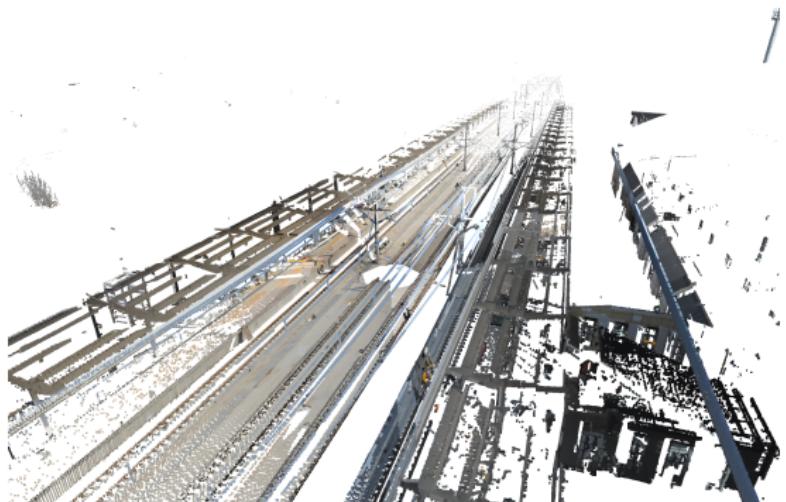
статья: Intrinsic Shape Signatures: A Shape Descriptor for 3D Object Recognition, 2009

Матчинг дескрипторов

$$\text{dist}(A, B) = \sqrt{\sum_{i=1}^{352} (A_i - B_i)^2}$$

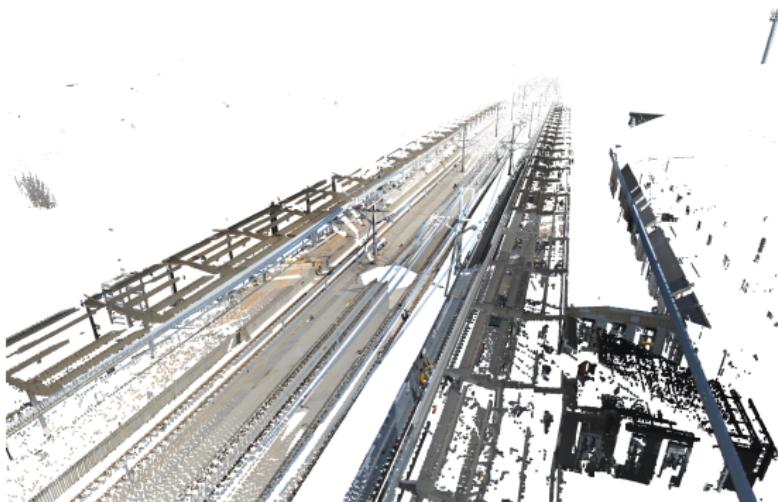
- K-d дерево
- Полный перебор

Фильтрация сопоставлений

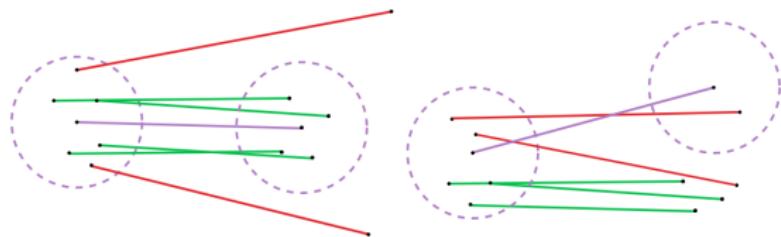


Фильтрация сопоставлений

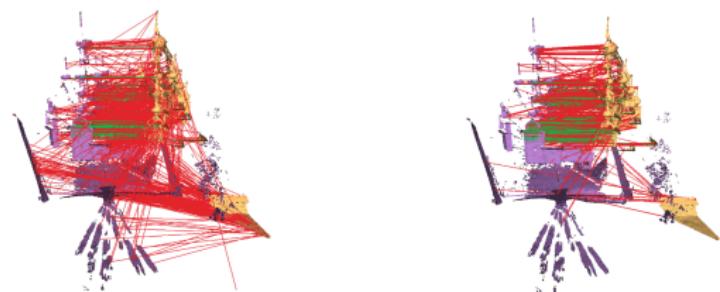
- K-ratio test
- Left-right check



Фильтрация сопоставлений



- Cluster filtering



- N - количество соответствий
- M - количество итераций
- for _ in range(M):
 samples = `sample(correspondences, 3)`
 T = `estimateTransformation(samples)`
 inlierRate = `countInliers(correspondences, T) / N`

- N - количество соответствий
- M - **максимальное** количество итераций
- for _ in range(**min(M, estimatedM)**):
 samples = **sample(correspondences, 3)**
 T = **estimateTransformation(samples)**
 inlierRate = **countInliers(correspondences, T) / N**
 estimatedM = **estimateIterations(inlierRate, M)**

- N - количество соответствий
- M - **максимальное** количество итераций
- for _ in range(**min(M, estimatedM)**):
 samples = **sample(correspondences, 3)**
 T = **estimateTransformation(samples)**
 inlierRate = **countInliers(correspondences, T) / N**
 estimatedM = **estimateIterations(inlierRate, M)**

- Количество итераций должно быть не меньше $\frac{\log 1-p}{\log 1-w^3}$

w - доля шумов

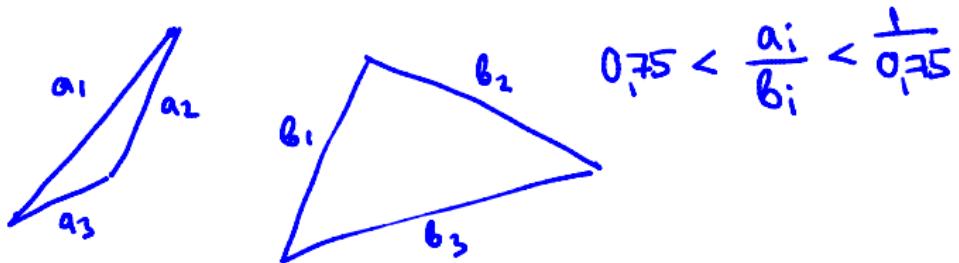
$$P[\text{все выборки содержат аутлиеров}] = (1-w^3)^M$$

$$(1-w^3)^M < 1-p \leftarrow \begin{array}{l} \text{заранее заданный} \\ \text{параметр уверенность} \\ n-p, 0,999 \end{array}$$

$$M > \frac{\log(1-p)}{\log(1-w^3)}$$

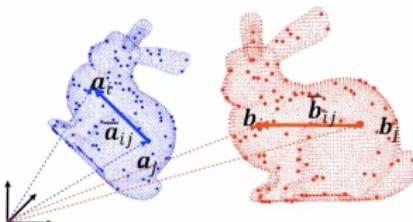
- Количество итераций должно быть не меньше $\frac{\log 1-p}{\log 1-w^3}$
- Если выборка не удовлетворяет геометрическим ограничениям, количество инлаеров для неё можно не считать

- Количество итераций должно быть не меньше $\frac{\log 1-p}{\log 1-w^3}$
- Если выборка не удовлетворяет геометрическим ограничениям, количество инлаеров для неё можно не считать
- Полученные параметры можно уточнить, используя аналитическое решение для инлаеров

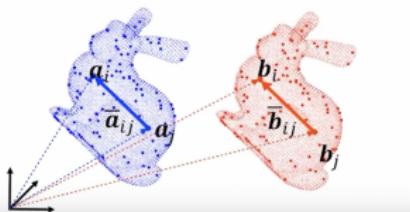


<https://github.com/MIT-SPARK/TEASER-plusplus>

Insight 1: Relative distances are invariant to translation and rotation



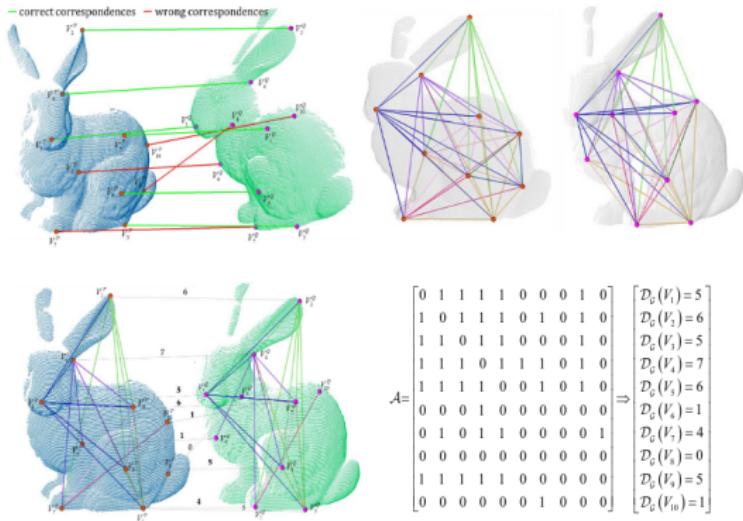
Insight 2: Relative positions between points are invariant to translation



$$\min_{s>0, R \in \text{SO}(3), t \in \mathbb{R}^3} \sum_{i=1}^N \min \left(\frac{1}{\beta_i^2} \|b_i - sRa_i - t\|^2, \bar{C}^2 \right)$$

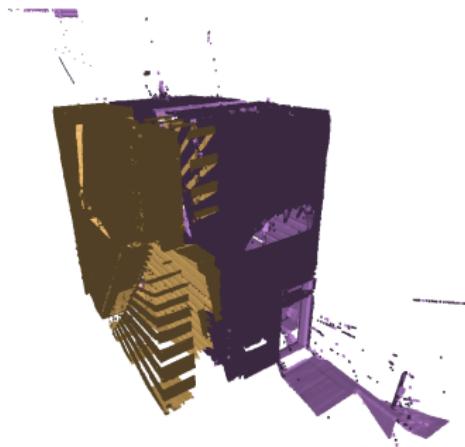
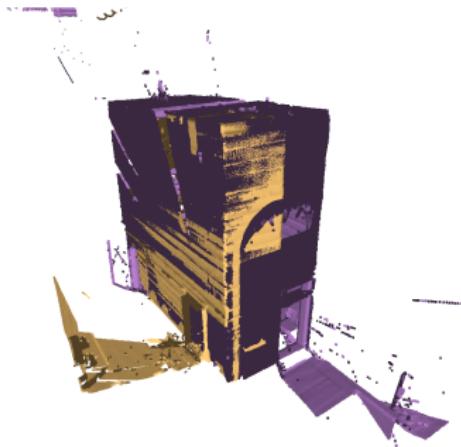
статья: TEASER: Fast and Certifiable Point Cloud Registration, 2020

<https://github.com/WPC-WHU/GROR>



статья: A New Outlier Removal Strategy Based on Reliability of Correspondence Graph for Fast Point Cloud Registration, 2022

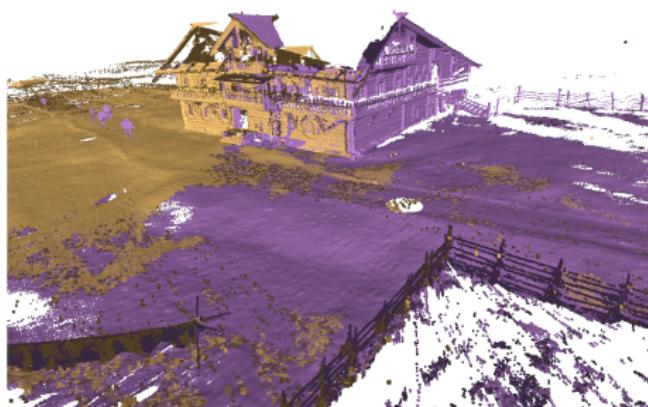
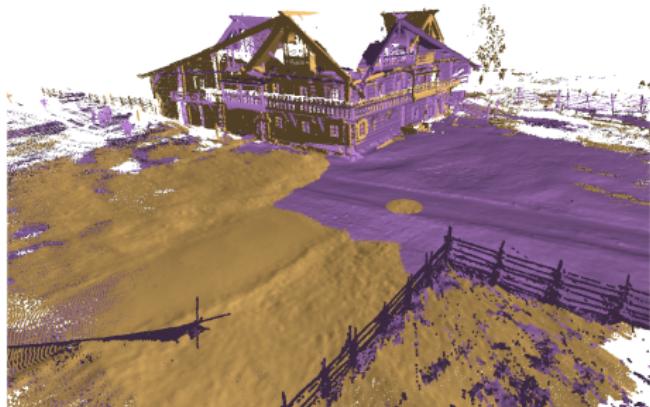
Метрика RANSAC



Метрика RANSAC



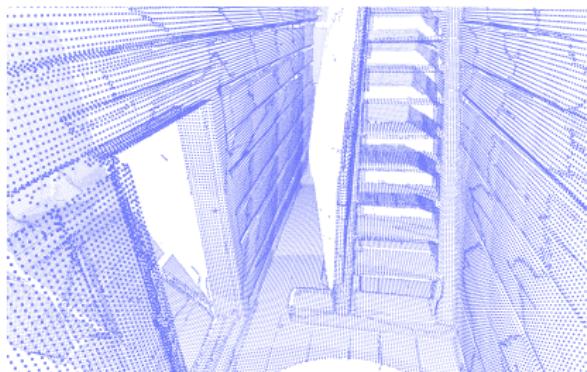
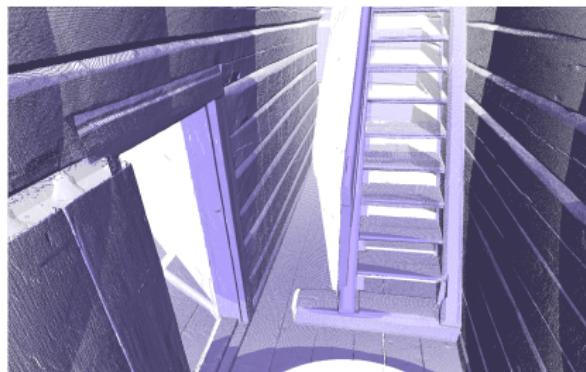
Метрика RANSAC



- Количество инлаеров
- Размер перекрытия
- Энтропия инлаеров

Разрешение облака точек

- Облака точек содержат несколько миллионов точек
- Работать с такими облаками напрямую – долго



Iterative Closest Point (ICP)

На каждой итерации:

- 1 Для каждой точки исходного облака находим ближайшую в целевом

$$k_i = \operatorname{argmin}_j \|x_i - T \cdot x_j\|$$

$$e_i = x_i - T \cdot x_{k_i}$$

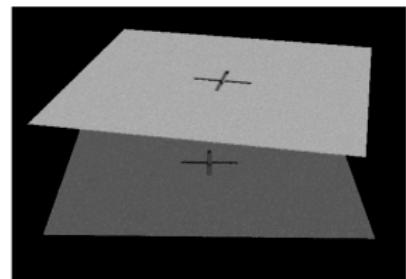
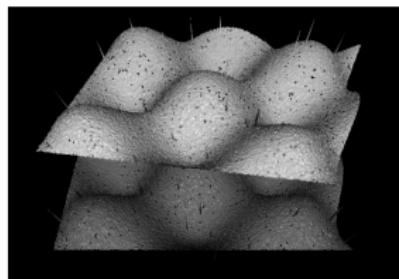
- 2 Вычисляем параметры преобразования, используя получившиеся соответствия

$$T^* = \operatorname{argmin}_{T \in SE(3)} \sum_i \|t_i(T)\|^2$$

статья: A Method for Registration of 3-D Shapes, 1992

Iterative Closest Point (ICP)

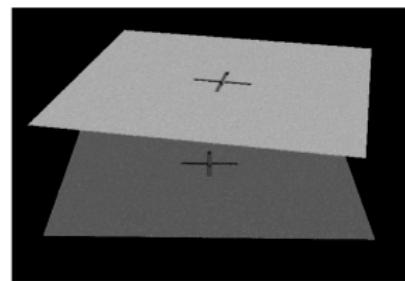
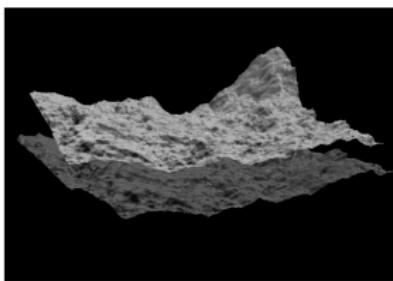
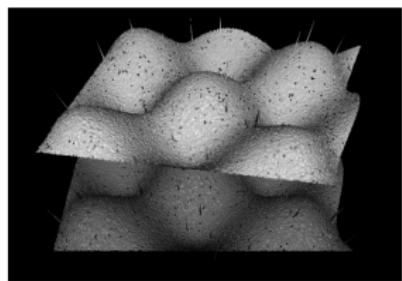
- Вместо того чтобы брать все точки, можем их сэмплировать:
 - случайным образом
 - равномерно в пространстве нормалей



статья: Efficient Variants of the ICP Algorithm, 2001

Iterative Closest Point (ICP)

- Вместо того чтобы брать все точки, можем их сэмплировать:
 - случайным образом
 - равномерно в пространстве нормалей



- Каждая точка может вносить вклад с разным весом:
 - $w = 1 - \frac{dist(p_1, p_2)}{dist_{max}}$
 - $w = n_1^T n_2$

статья: Efficient Variants of the ICP Algorithm, 2001

Регистрация облаков точек



Какие ещё остались проблемы?

- Разрешения сопоставляемых облаков могут отличаться в области перекрытия
- Какой использовать радиус дескриптора?

Какие ещё остались проблемы?

- Разрешения сопоставляемых облаков могут отличаться в области перекрытия
- Какой использовать радиус дескриптора?
- Как сопоставлять более двух облаков точек?

Какие ещё остались проблемы?

- Разрешения сопоставляемых облаков могут отличаться в области перекрытия
- Какой использовать радиус дескриптора?
- Как сопоставлять более двух облаков точек?

