

Agisoft



Metashape

Полярный Николай
polarnick@agisoft.com

Metashape

Основная задача:

По множеству фотографий восстановить трехмерную модель.



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

Metashape

Основная задача:

По множеству фотографий восстановить трехмерную модель.



Данные предоставил Stéphane Prodent

Предложенные задачи

1. Распознавание размытия на фотографиях (**out-of-focus blur**)
2. Улучшение сопоставления ключевых точек (**SURF**) в случае terrestrial laser scanning (360-градусного **лидара с RGB** фотографией)

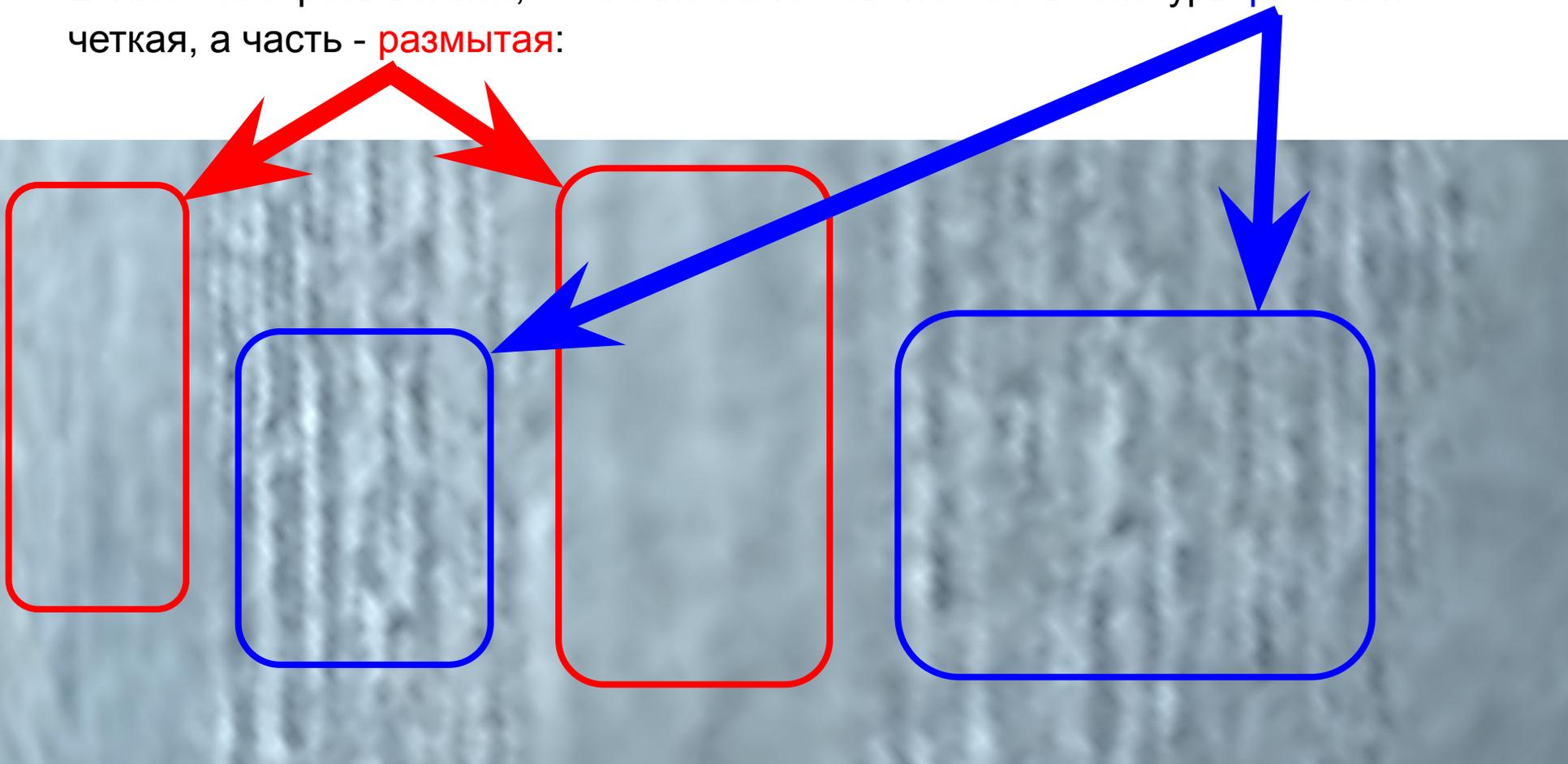
1. Распознавание размытия на фотографиях

Пусть был фотосканирован такой тапочек (построена геометрия и текстура):



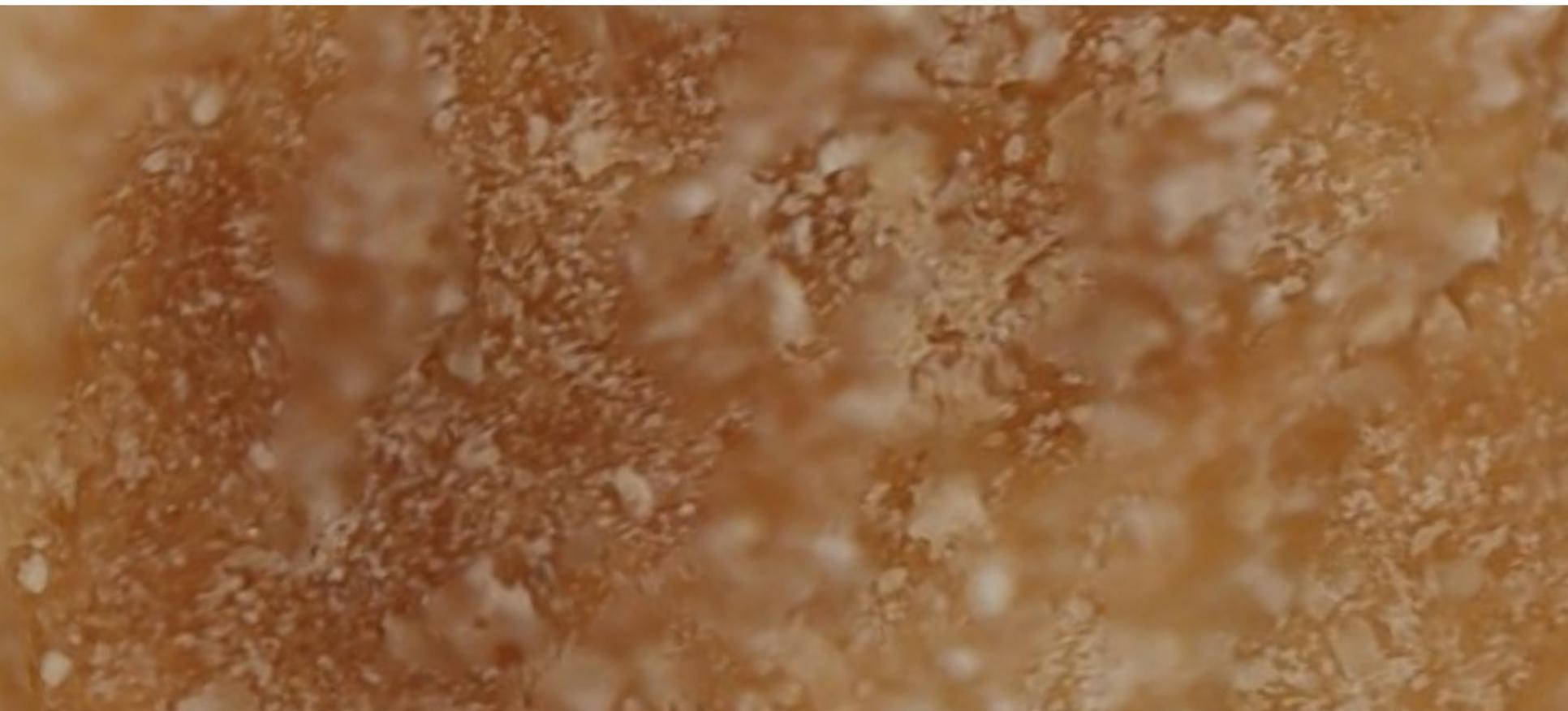
1. Распознавание размытия на фотографиях

Если посмотреть вблизи, то можно заметить что часть текстуры **резкая** и четкая, а часть - **размытая**:



1. Распознавание размытия на фотографиях

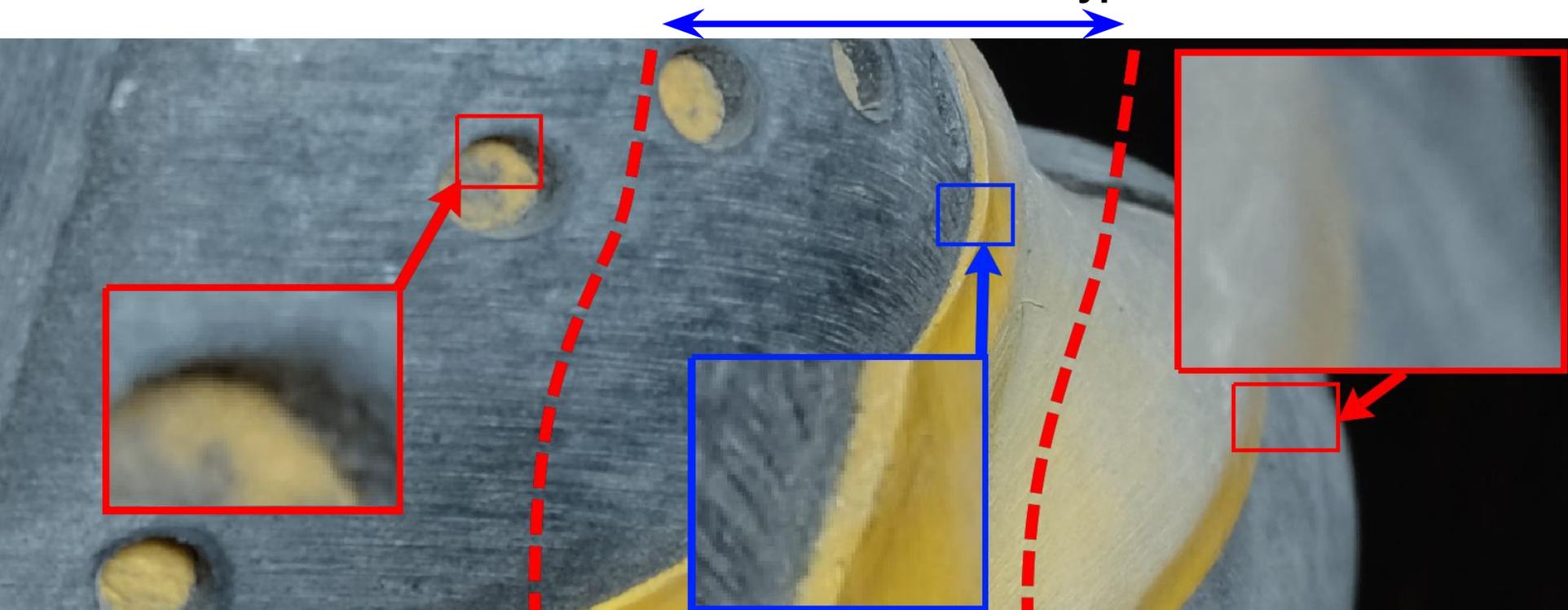
Еще пример - реконструированная текстура маффина:



1. Распознавание размытия на фотографиях

Причина - в фотографии четко получается лишь то что лежит в диапазоне глубины фокуса (все что ближе или дальше - размыто - out of focus blur):

in focus - четкая текстура



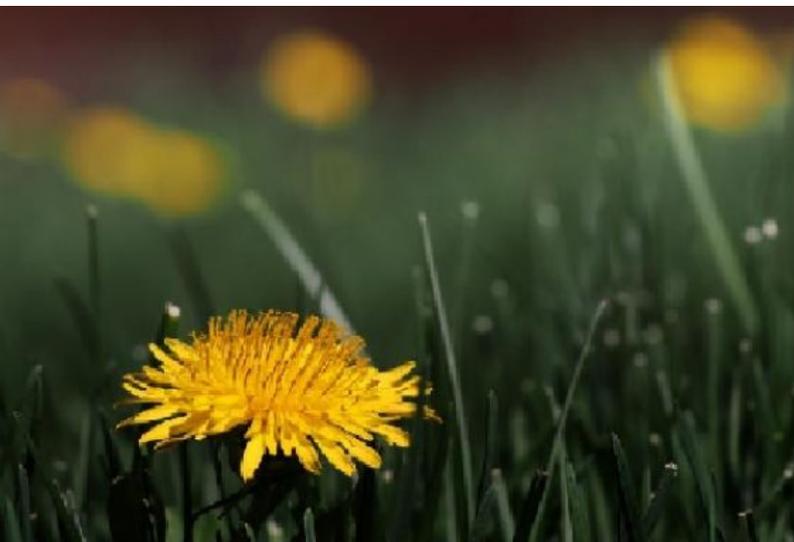
out of focus blur

т.к. **ближе** чем фокус

out of focus blur

т.к. **дальше** чем фокус

1. Распознавание размытия на фотографиях



Хочется для каждого пикселя фотографии **определить степень размытия.**

Тогда на этапе комбинирования фотографий в текстуру можно будет это учесть.

Т.е. станет возможно смешивать фотографии взвешенно:
на каких фотографиях какая часть поверхности в фокусе - туда и будет взята четкая текстура из этой фотографии.

1. Распознавание размытия на фотографиях

Предлагается реализовать [1]:



1. Распознавание размытия на фотографиях

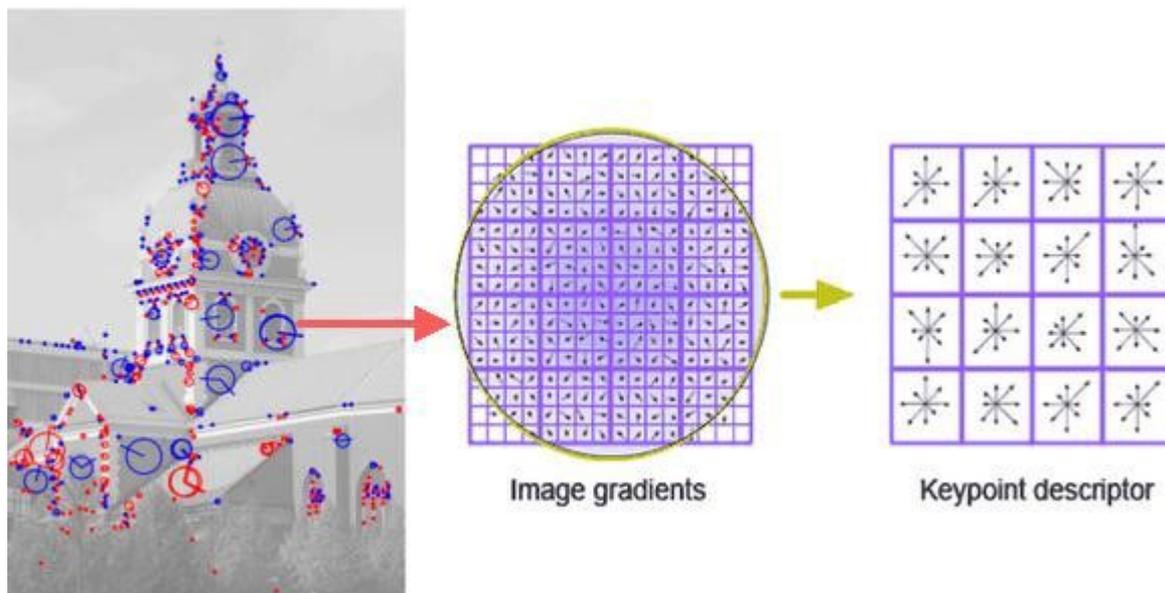
[1] [Defocus map detection using a single image, Juan Andrade, 2016](#)

[2] [Focusing on out-of-focus: assessing defocus estimation algorithms for the benefit of automated image masking, Geert J. Verhoeven, 2018](#)

2. Улучшение сопоставления ключевых точек

Введение: как вообще находится взаимное положение фотографий?

1) На фотографиях находятся “хорошие” точки (feature points detection)



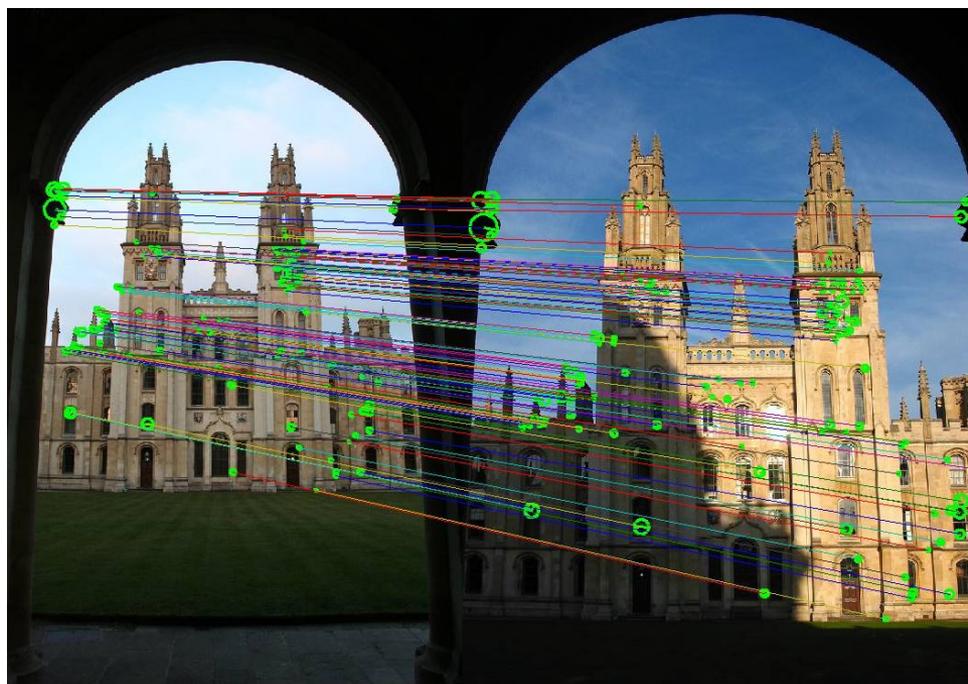
Источник иллюстрации:

<https://medium.com/machine-learning-world/feature-extraction-and-similar-image-search-with-opencv-for-newbies-3c59796bf774>

2. Улучшение сопоставления ключевых точек

Введение: как вообще находится взаимное положение фотографий?

- 1) На фотографиях находятся “хорошие” точки (feature points detection)
- 2) Сопоставляются эти ключевые точки между парами кадров (matching)



Источник иллюстрации:

<https://blog.ekbana.com/extracting-invariant-features-from-images-using-sift-for-key-point-matching-675f818ce199>

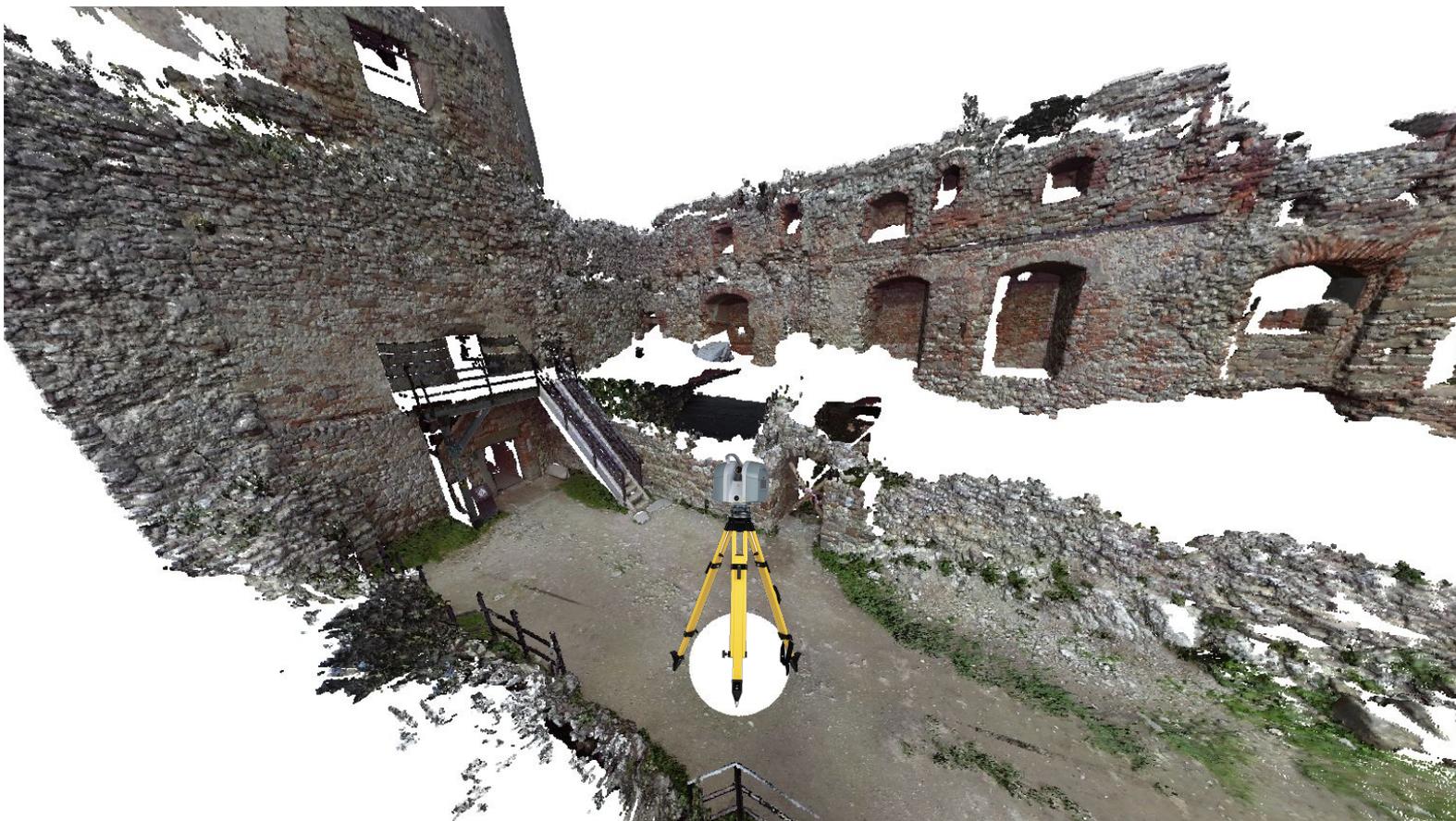
2. Улучшение сопоставления ключевых точек

Не только на базе фотографий осуществляется реконструкция.
В некоторых случаях используется **LIDAR** (часто с фотограмметрией):



2. Улучшение сопоставления ключевых точек

У таких LIDAR-ов точки на самом деле **цветные**:



2. Улучшение сопоставления ключевых точек

Более того - облако хранится в структурированном виде.

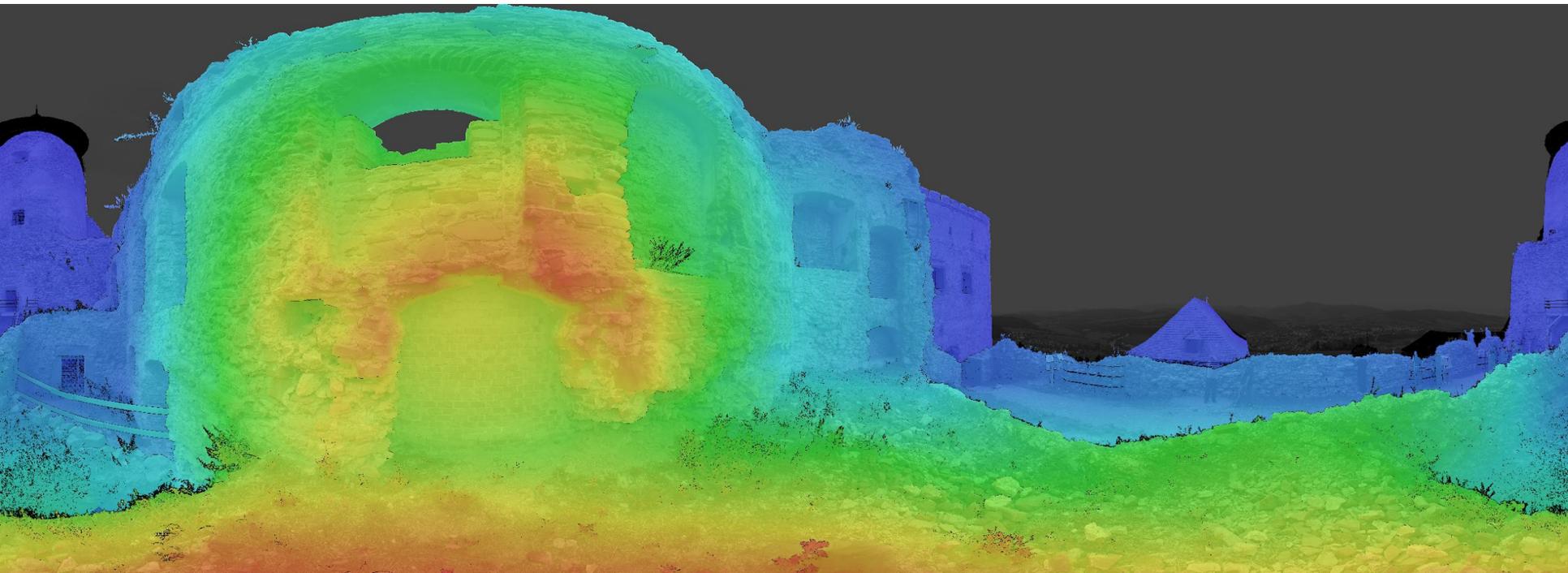
В виде **360-картинки с глубиной**:



2. Улучшение сопоставления ключевых точек

Более того - облако хранится в структурированном виде.

В виде **360-картинки с глубиной:**



2. Улучшение сопоставления ключевых точек

Умеем сопоставлять:

- фотографию с фотографией

Хочется научиться сопоставлять:

- LIDAR облако с LIDAR облаком
- LIDAR облако с фотографией

Чтобы было просто обрабатывать комбинированную LIDAR + фото съемку.

Хорошая идея - вспомнить что LIDAR облако - почти что картинка.

Поэтому почему бы не детектировать на ней ключевые точки как будто бы это 360-картинка?

2. Улучшение сопоставления ключевых точек



И это работает.

Но не всегда.

Т.к.

сильные искажения

и

LIDAR кадр от LIDAR кадра
обычно **находится далеко**

(т.к. долгое сканирование,
а значит мало кадров
можно успеть снять).

2. Улучшение сопоставления ключевых точек

Хочется поглотить сильные искажения (устранить аффинное искажение):

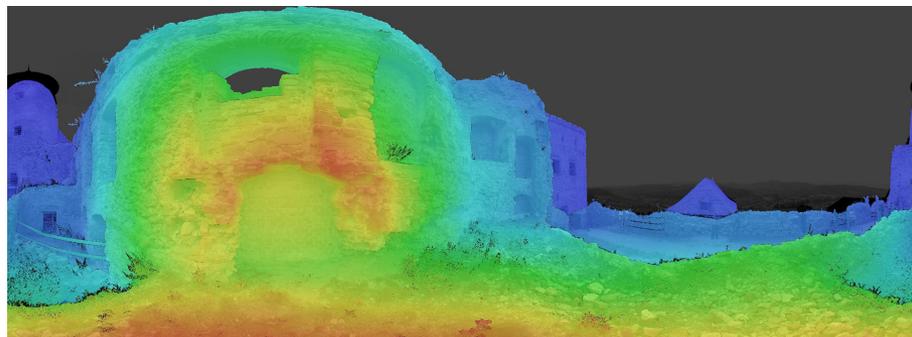


2. Улучшение сопоставления ключевых точек

Хочется поправить сильные искажения (устранить аффинное искажение).

Для этого нам нужно знать в какой пространственной плоскости лежит окрестность данной точки.

В фотографиях такой информации нет, но в LIDAR наравне с RGB есть глубина, а значит можно построить плоскость окрестности по соседним точкам переводя их в 3D (ведь известна глубина).



2. Улучшение сопоставления ключевых точек

Итого план:

- 1) Для каждой точки по соседним точкам **построить плоскость**.
- 2) **Подменить в расчете дескриптора** обращения к цветам картинки в окрестности вокруг точки, чтобы оно было с учетом компенсации локального искажения. (дескриптор - вероятно **SURF в OpenCV**)

Попутно:

- Пробовать разные дескрипторы
- Визуализировать все промежуточные результаты для отладки (плоскости в 3D, устранение искажения в окрестности и т.п.)
- Читать статьи

Организационные детали

- Язык: **C++**
- Адрес офиса: [Дегтярный Переулок, 11 лит. Б](#)
- С любыми вопросами можно писать на polarnick@agisoft.com или <http://t.me/PolarNick239>
- К предложенным темам есть тестовые задания
- Во всех практиках нужно будет читать статьи

Вопросы?



Agisoft

Полярный Николай

polarnick@agisoft.com