

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:

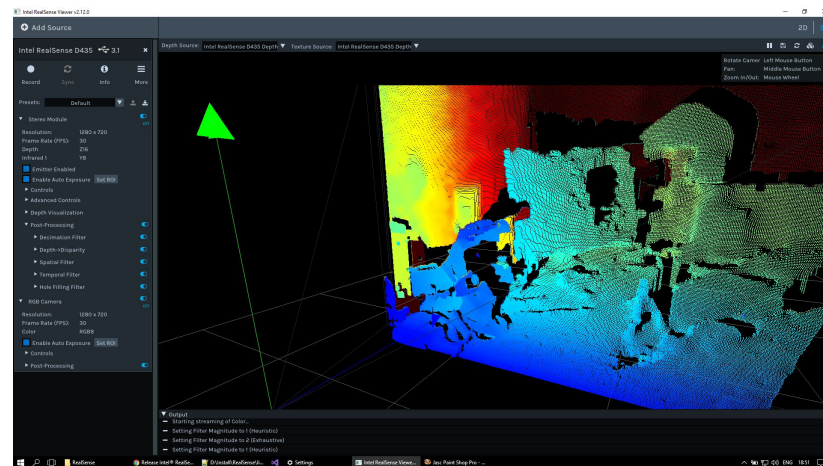
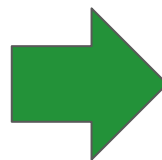
По множеству фотографий восстановить трехмерную модель.



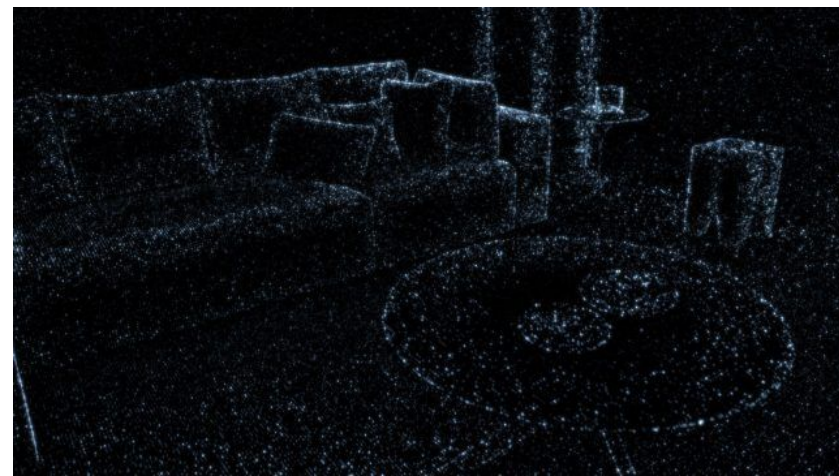
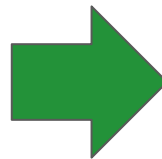
Данные предоставил Stéphane Prodent

1. Поддержка RGB-D сенсоров

Intel® RealSense D435



Apple iPhone/iPad LiDAR



1. Поддержка RGB-D сенсоров. Задача на семестр

В Metashape уже есть поддержка полноценных LIDAR-ов, но нет поддержки карт глубины полученных с разнородных телефонов/Intel RealSense.

Зато есть Python API через который можно эту поддержку добавить.

Хочется на **Python** преобразовать данные сначала с Intel RealSense D435, затем с iPhone/iPad в подходящий формат (двустраничный .tiff-файл состоящий из цветной картинки и карты глубины) и [опубликовать](#) скрипт для пользователей.

1. Поддержка RGB-D сенсоров. Тестовое.

Черпая информацию из поста <http://ngghiaho.com/?p=2629> преобразовать каждую RGB-D картинку из его [данных](#) в хоть какое-то правдоподобное цветное облако точек (в виде бинарной версии формата [.ply](#), должен открываться в **MeshLab**, система координат у каждой картинки может быть своя, т.е. их взаимное выравнивание делать не нужно).

Результатом тестового задания является исходный код на питоне (преобразующий все эти картинки в облака точек), а так же результирующие облака точек в zip-архиве.

Отправьте и то и то мне на почту - polarnick@agisoft.com

Дедлайн: 15 февраля включительно

1. Поддержка RGB-D сенсоров. Мелочи

- Язык: **Python**
- Адрес офиса: [Дегтярный Переулок, 11 лит. Б](#)
(как минимум потребуется заехать чтобы передать RealSense камеру)
- С любыми вопросами можно писать на polarnick@agisoft.com или <http://t.me/PolarNick239>
- Нужно будет разбираться в разных форматах файлов

2. Подсчет деревьев (машинным обучением)



Иллюстрация: <https://blog.droneDeploy.com/product-release-wrap-up-april-2019-9a67f8bbfd0f>

2. Подсчет деревьев.

Машинное обучение в виде “обучили на куче датасетов, отдали обученную модель пользователям для запуска на их данных” - для нас не работает, т.к. наши пользователи рано или поздно запустят модель на синих деревьях посреди лавового леса где бегают очень мало единорогов, что нетипично для датасетов в обучающей выборке и поэтому работать не будет.

Иначе говоря - обучить на “достаточном” числе датасетов почти невозможно, т.к. мир очень разный.

2. Подсчет деревьев. Transfer learning

Но можно обучить модель так чтобы она умела выделять ключевые особенности из того что видит, а что из всего этого пользователю нужно детектировать и считать - можно спросить у него на каждом датасете отдельно.

Т.е. пользователь выделяет маленький кусочек датасета, размечает его, предобученная нейронная сеть дообучается на этом маленьком наборе данных, и затем запускается на всем датасете. Получается умный обучаемый помощник.

Бонус: детектирование и подсчет чего-нибудь странного, например подсчет машин, морских котиков, полиэтиленовых шариков для лопания, чего угодно.

2. Подсчет деревьев. Задача на семестр

Найти примеры (jupyter notebook, google colab исходников) как делается transfer learning и отталкиваясь от этого:

- 1) Взять обученную хорошую нейронную сеть
- 2) Убрать у нее последние слои (оставив только те что занимаются выделением признаков и высокоуровневых фичей)
- 3) Дообучить ее на малой разметке (часть датасета)
- 4) Прогнать по всему датасету

Обратите внимание что решаться будет задача детекции, а не задача сегментации. В тестовом задании предлагается задача сегментации из соображений упрощения.

2. Подсчет деревьев. Тестовое

- 1) Скачать размеченные данные - [trees_segmentation_data.zip](#)
- 2) Обучить любую нейронную сеть (без transfer learning) задаче сегментации по разметке из папки 'train'
- 3) Оценить точность на разметке папки 'test' (precision и recall)
- 4) Обучить модель на разметке сразу из обеих папок
- 5) Построить маски сегментации для картинок из 'final'
- 6) Найти исходники (jupyter notebook/google colab) от которых вы бы отталкивались для попытки решения этой же задачи через transfer learning
- 7) Отправить построенные маски для 'test' и 'final', значения precision и recall для test, исходники (если ноутбук, то и в виде pdf, и в виде .ipynb), выбранную ссылку из пункта 6 мне на почту - polarnick@agisoft.com



Дедлайн: 15 февраля включительно

2. Подсчет деревьев. Мелочи

- Язык: **Python**
- Адрес офиса: [Дегтярный Переулок, 11 лит. Б](#)
(предположительно будет полностью удаленно)
- С любыми вопросами можно писать на polarnick@agisoft.com или <http://t.me/PolarNick239>
- Нужно быть готовым пробовать много предобученных моделей, критически смотреть на результаты и пытаться зайти как можно дальше

Вопросы?



Agisoft

Полярный Николай

polarnick@agisoft.com

[Telegram/PolarNick239](https://t.me/PolarNick239)