

Monoceros

Полярный Николай

Помогают:

Никита Офицеров (обсуждаем архитектуру)

Роман Силаков (участвует в разработке рендеринга)

**КРОНШТАДТ
ТЕХНОЛОГИИ**



PolarNick239@gmail.com

Задача

Вход: фотографии с БПЛА

Задача

Вход: фотографии с БПЛА

Выход:

1. Карта
2. Оцифрованная 3D местность

Задача

Вход: фотографии с БПЛА

Выход:

1. Карта
2. Оцифрованная 3D местность

Ограничения:

1. Реальное время
2. Отказоустойчивость

Вход



Выход



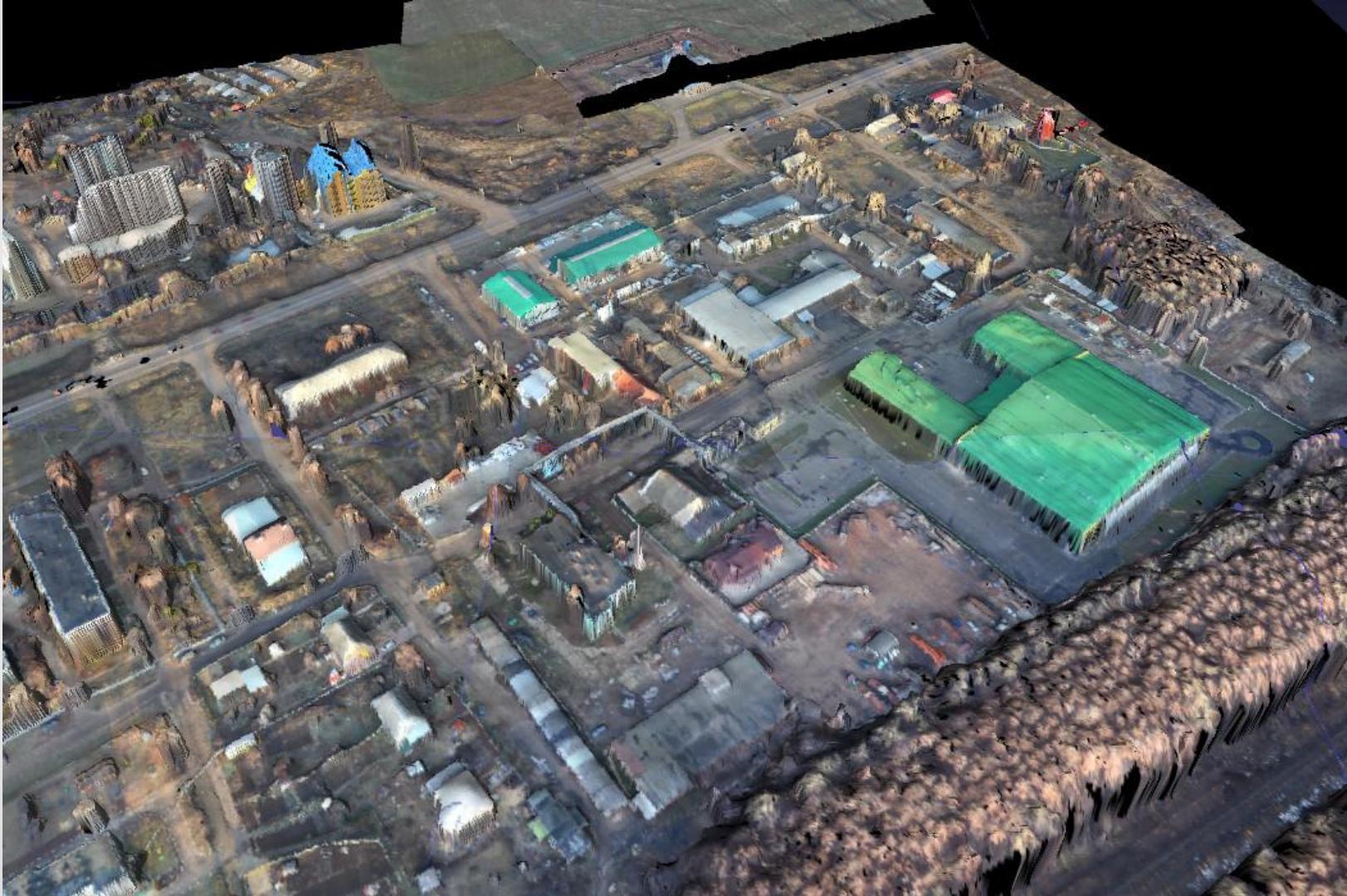
Выход (цвет - кадр)



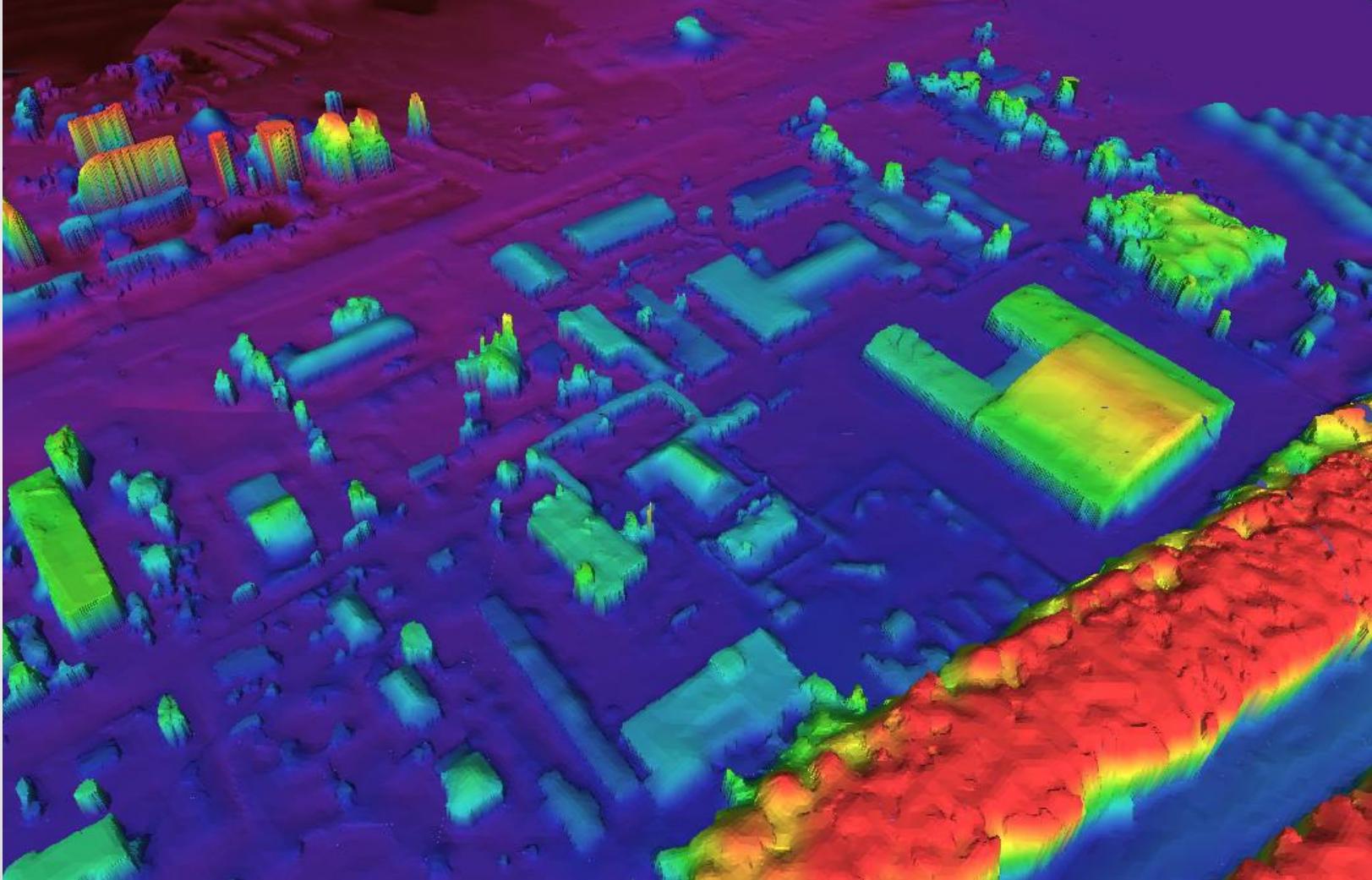
Спутник



3D модель (цвет)



3D модель (высота)



К прошлому отчету было сделано

Реализован сервер с базовой функциональностью:

1. Online обработка потока фотографий
2. Offline пост-обработка данных
3. Отказоустойчивость (перезапуск сервера без потери данных в любой момент)
4. Параллельная обработка слоев
5. HTTP API для толстого клиента

Но были проблемы

1. ☹ Потребление памяти росло со временем линейно, а слоев много
2. ☹ Неэффективная модель данных (ради отказоустойчивости)
3. ☹ Медленная отрисовка карты (online-задача)
4. ☹ Ошибки во внешних библиотеках и IDE (OpenCV, PyOpenCL, PyCharm!)

~~Но были проблемы~~

1. ☺ Память ограничена константой
(много слоев по 20 Гб данных)
2. ☺ Оптимальная модель данных
(атомарные записи, lock-free методы)
3. ☺ Быстрый рендеринг
(предпросмотр 20 Гб за 10 секунд)
4. ☺ Один SegFault в OpenCV победил,
отправил несколько pull-request-ов

Что нового?

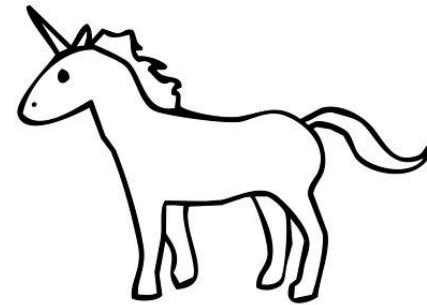
- геопинирование (привязка к спутнику)
- возможность отрисовки на клиентской стороне
- запуск с заранее подготовленными состояниями
(реализовано с помощью hardlinks)
- очень быстрый предпросмотр
(20 Гб за 10 секунд)

Что нового? (еще!)

- многоракурсность
- своя визуализация
- часть тестов теперь детерминированы
(по-хорошему надо синтетические тесты)
- оптимизирован Cython код
- сделаны шаги в сторону много-
контекстного рендеринга (несколько GPU)

Спасибо!
Вопросы?

Being a human is
too complicated.



Time to be
a unicorn.

Полярный Николай
PolarNick239@gmail.com
2015 г.