

# Реализация кросс-платформенной библиотеки нахождения глубины изображения по стереопаре

Пушкин Тимофей Дмитриевич, группа 20.Б11-мм

17 Май 2022 г. 24

***Научный руководитель:***

старший преподаватель каф. СП Я.А. Кириленко

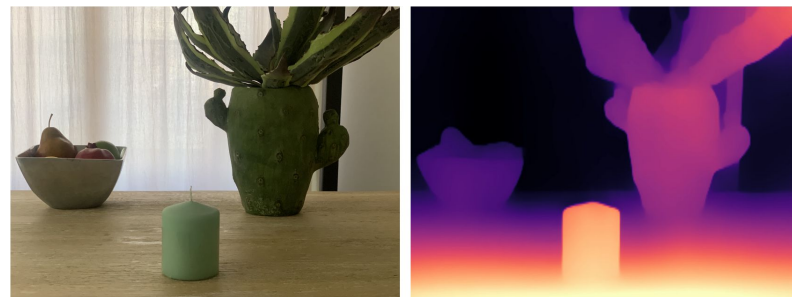
***Консультанты:***

инженер-исследователь, Сколтех, А.В. Корнилова,

Android-разработчик, VK, Д.С. Ярош

# Карты глубины

- Содержат информацию об удаленности точек сцены от камеры
- Используются для 3D-реконструкции, оцифровки объектов, в виртуальной и дополненной реальностях
- Традиционно вычисляются при помощи специального оборудования
- Можно использовать смартфоны, которые широко распространены



**Рис.** Пример карты глубины

[Источник](#)

# Смартфоны в качестве стереокамеры

**StereoSGBM**<sup>1</sup> — алгоритм в составе библиотеки OpenCV<sup>2</sup>, который может быть применен для вычисления карт глубины по двум изображениям (стереопаре)



Рис. Пример установки из двух смартфонов

[Источник](#)

1 — [https://docs.opencv.org/4.5.5/d2/d85/classcv\\_1\\_1StereoSGBM.html](https://docs.opencv.org/4.5.5/d2/d85/classcv_1_1StereoSGBM.html)

2 — <https://opencv.org>

# Цель и задачи практики

**Цель:** создание кросс-платформенной библиотеки для нахождения карты глубины по стереопаре с применением алгоритма StereoSGBM из библиотеки OpenCV

## Задачи:

- Выполнить обзор механизмов использования OpenCV на различных платформах
- Реализовать кросс-платформенную библиотеку на основе демонстрационной версии подхода
- Провести качественное сравнение полученной реализации с оригинальной
- Провести апробацию полученной библиотеки, реализовав на её основе мобильное приложение для нахождения карты глубины по стереопаре
- Провести проверку возможности применения полученной реализации для вычисления карт глубины в мягком реальном времени на мобильных устройствах

# Использованный подход

**Входные данные** — два изображения одной сцены (стереопара), калибровочные параметры

## **Этапы вычисления карты глубины:**

1. Выравнивание изображений
2. Масштабирование изображений
3. Вычисление взаимного смещения точек сцены
4. Построение карты глубины по найденному смещению

# Подключение OpenCV

<b>Мобильные SDK</b>	<b>Сборка из исходного кода</b>
<ul style="list-style-type: none"><li>+ Поставляются в виде готовых сборок</li><li>+ Содержат готовые языковые интерфейсы</li><li>- Препятствуют кросс-платформенности</li><li>- Возможны проблемы с производительностью</li></ul>	<ul style="list-style-type: none"><li>+ Возможна гибкая конфигурации сборки</li><li>- Необходима самостоятельная сборка</li><li>- Нет языковых интерфейсов для мобильных платформ</li></ul>

# Генерация языковых интерфейсов

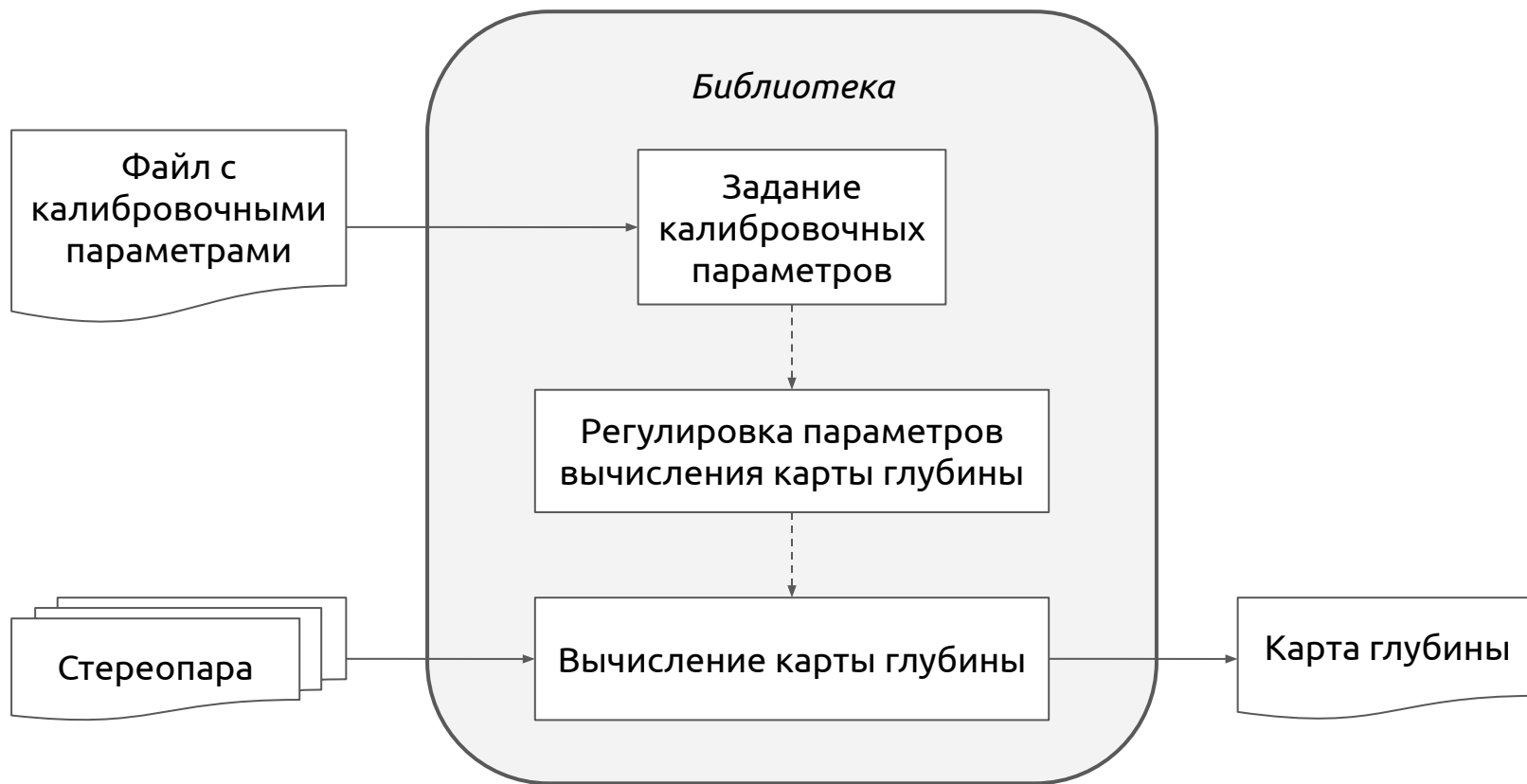
Djinni <sup>1</sup>	SWIG <sup>2</sup>	Scapix <sup>3</sup>
<ul style="list-style-type: none"><li>+ Активное сообщество</li><li>- Необходим файл-интерфейс</li><li>- Официальная поддержка прекращена</li></ul>	<ul style="list-style-type: none"><li>+ Активное сообщество</li><li>+ Расширенная поддержка в CMake</li><li>- Необходим файл-интерфейс</li></ul>	<ul style="list-style-type: none"><li>+ Не требуется файл-интерфейс</li><li>- Необходимо точечное изменение заголовочных файлов проекта</li></ul>

1 — <https://github.com/dropbox/djinni>

2 — <https://www.swig.org>

3 — <https://github.com/scapix-com/scapix>

# Взаимодействие с реализованной библиотекой





# Сборка библиотеки

Автоматическая загрузка и  
сборка OpenCV через  
отдельный CMake-скрипт

или

Использование OpenCV,  
установленного в системе

↓  
Подключение OpenCV

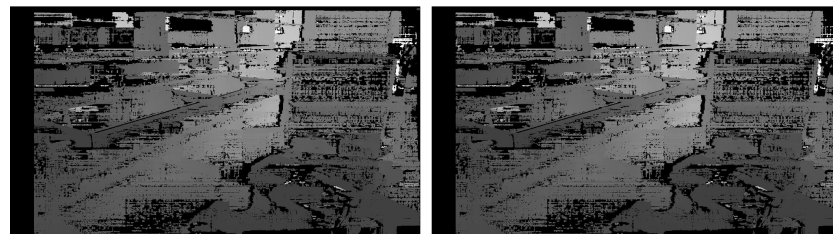
↓  
Сборка проекта из исходного кода

↓  
(Опционально) Генерация языкового  
интерфейса через Scarix

# Качественная оценка: полученные значения

Стереопара	Абс. разность	Отн. разность
1	$0.04 \pm 0.08$ мм	$1.4 \cdot 10^{-6} \pm 8 \cdot 10^{-7}$
2	$0.09 \pm 0.10$ мм	$1.3 \cdot 10^{-6} \pm 8 \cdot 10^{-7}$
3	$0.04 \pm 0.06$ мм	$0.8 \cdot 10^{-6} \pm 6 \cdot 10^{-7}$
4	$0.06 \pm 0.07$ мм	$0.6 \cdot 10^{-6} \pm 5 \cdot 10^{-7}$

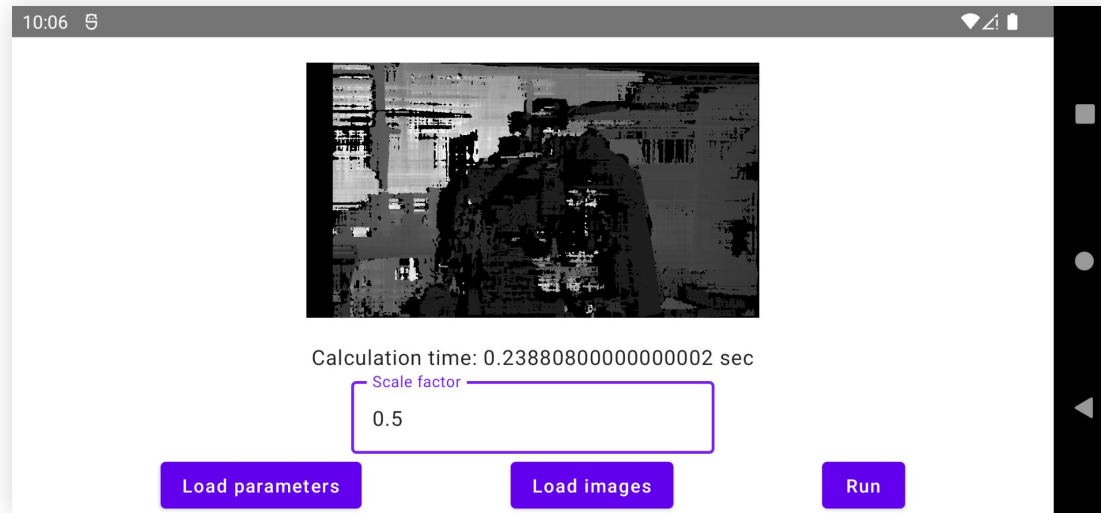
**Таблица.** Разности значений в точках карт глубины, подсчитанных при помощи полученной и оригинальной реализаций (среднее  $\pm$  стандартное отклонение)



**Рис.** Изображение из стереопары (сверху), карта глубины от оригинальной реализации (слева), карта глубины от полученной реализации (справа)

# Апробация на Android

На основе реализованной библиотеки было создано Android-приложение



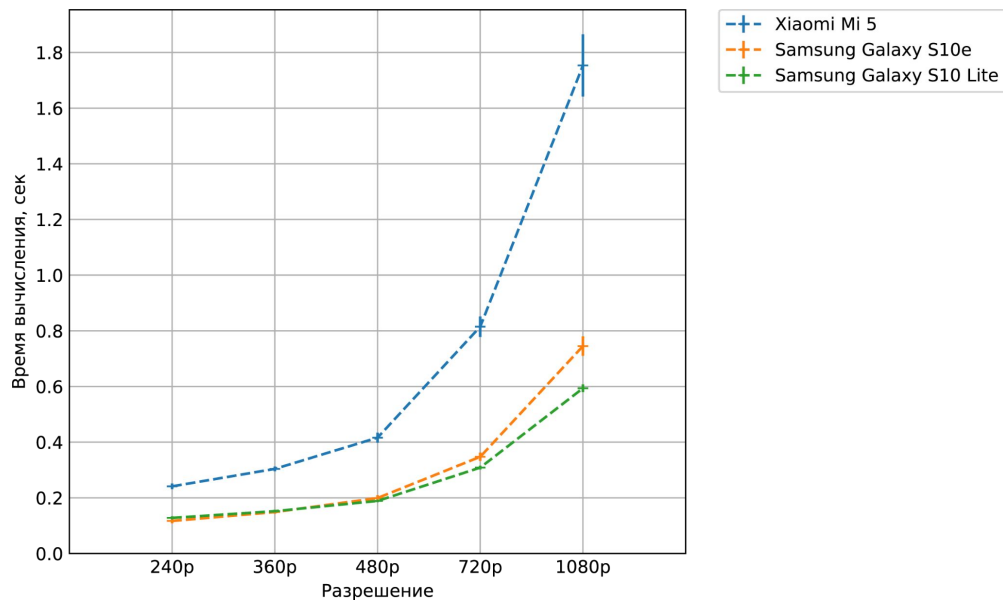
**Рис.** Графический интерфейс реализованного Android-приложения

# Оценка производительности: устройства

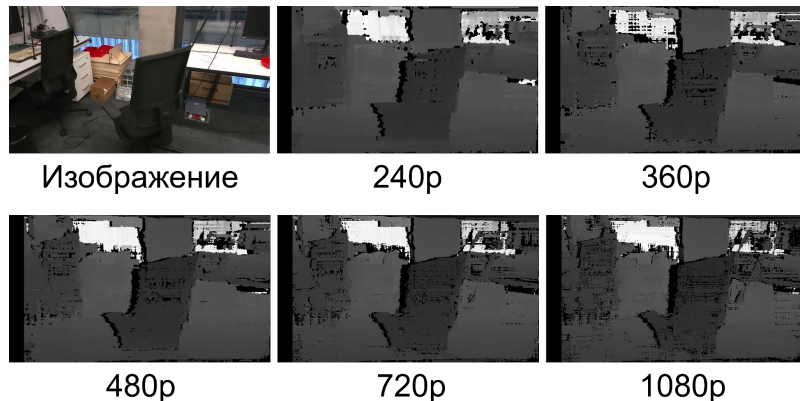
Замеры проводились на следующих устройствах:

- Xiaomi Mi 5 (Android API 30, LineageOS 18.1)
  - Год выпуска: 2016
  - Процессор: Qualcomm Snapdragon 820
- Samsung Galaxy S10e SM-G970F/DS (Android API 31, One UI 4.0)
  - Год выпуска: 2019
  - Процессор: Samsung Exynos 9820
- Samsung Galaxy S10 Lite SM-G770F/DS (Android API 31, One UI 4.0)
  - Год выпуска: 2020
  - Процессор: Qualcomm Snapdragon 855

# Оценка производительности: результаты



**Рис.** График зависимости времени вычисления карты глубины от разрешения стереопары; изображены средние значения со стандартным отклонением за 50 замеров; меньше — лучше



**Рис.** Карты глубины, вычисленные при различном предварительном масштабировании; все изображения приведены к разрешению 1080p средствами стороннего программного обеспечения

# Результаты практики

Были выполнены следующие задачи:

- Выполнен обзор механизмов использования OpenCV на различных платформах
- Реализована кросс-платформенная библиотека на основе демонстрационной версии подхода к нахождению карты глубины по стереопаре
- Проведено качественное сравнение полученной реализации с оригинальной
- Проведена апробация полученной библиотеки через реализацию на её основе мобильного приложения для нахождения карты глубины по стереопаре
- Проведена проверка возможности применения полученной реализации для вычисления карт глубины в мягком реальном времени на мобильных устройствах

Результаты работы доступны в [данном репозитории](#)