МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики

Кафедра геоинформатики и информационной безопасности

**ОТЧЕТ ПО ПРАКТИКЕ**

|  |  |
| --- | --- |
| Вид практики: | производственная практика |
|  | (учебная, производственная) |
| Тип практики: | научно-исследовательская работа |

Сроки прохождения практики: с 10.02.2020 г. по 10.06.2020 г.

по направлению подготовки 10.05.03 Информационная безопасность автоматизированных систем

(уровень академического специалитета)

направленность (профиль) «Обеспечение информационной безопасности распределенных информационных систем»

|  |  |
| --- | --- |
| Студент группы | № 6411-100503D Цой Г.В. |
| Руководитель практики  от университета | Сергеев В.В., профессор, д.т.н. |

Дата сдачи 09.06.2020 г.

Дата защиты 10.06.2020 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Самарский национальный исследовательский университет имени академика С.П. Королева»

(Самарский университет)

Институт информатики, математики и электроники

Факультет информатики

Кафедра геоинформатики и информационной безопасности

**Индивидуальное задание на практику**

Студенту группы № 6411-100503D Г.В. Цою

|  |
| --- |
| Направление на практику оформлено приказом по университету от 20.01.2020 № 28-ПР на кафедру геоинформатики и информационной безопасности Самарского университета |
| (наименование профильной организации или структурного подразделения университета) |

|  |  |  |
| --- | --- | --- |
| **Планируемые результаты освоения образовательной программы (компетенции)** | **Планируемые результаты практики** | **Содержание задания** |
| ОК-1 | Знать:  основные направления, проблемы, теории и методы философии, содержание современных философских дискуссий по проблемам общественного развития  Уметь:  использовать положения и категории философии для оценивания и системного анализа различных социальных тенденций, фактов и явлений и моделирования процессов в научной  деятельности; Владеть: навыками анализа текстов, имеющих философское содержание | Использовать при анализе и обосновании своей исследовательской проблемы научную философскую литературу. |
| ОК-2 | Знать:  базовые экономические понятия, законы функционирования экономики и поведения экономических агентов, показатели макроэкономического уровня развития страны, экономические показатели, используемые для оценки производственно- хозяйственной деятельности промышленных предприятий Уметь:  использовать понятийный аппарат экономической науки для описания экономических и финансовых процессов  Владеть: навыками использования экономических знаний в сфере личных финансов и профессиональной деятельности. | Использовать основы экономических знаний в обосновании результативности собственной научной работы. |
| ОК-3 | Знать:  закономерности и этапы исторического процесса, основные исторические факты, даты, события и имена исторических деятелей России; основные события и процессы отечественной истории в контексте мировой истории  Уметь:  критически воспринимать, анализировать и оценивать историческую информацию, факторы и механизмы исторических изменений  Владеть: навыками анализа причинно-следственных связей в развитии российского государства и общества; места человека в историческом процессе и политической организации общества; навыками уважительного и бережного отношения к историческому наследию и культурным традициям. | Рассмотреть исторические этапы развития научных взглядов на проблему своего дипломного исследования. |
| ОК-5 | Знать:  основные закономерности взаимодействия человека и общества, специфику профессиональной деятельности; основы социологии, структуру общества и социальных институтов; основные этические понятия, историю этических учений, современное положение в сфере этического знания; основные понятия культурологии, типологию культур. Уметь:  определять место и роль профессии в социальной сфере, взаимосвязь с другими профессиями; создавать и поддерживать высокую мотивацию к выполнению профессиональной деятельности; ориентироваться в этической проблематике; выявлять основные черты и особенности культурно-исторических ценностей.  Владеть: методами выявления мотивов социального поведения; технологиями анализа и прогноза социокультурных процессов для решения практических профессиональных проблем. | Осознавать социальную значимость своей будущей профессии, при выполнении профессиональной деятельности в области обеспечения информационной безопасности соблюдать нормы профессиональной этики. |
| ОК-9 | Знать:  основные средства и методы физического воспитания Уметь:  выбирать и применять методы и средства физической культуры для совершенствования основных физических качеств Владеть: навыками использования методов и средств физической культуры для обеспечения полноценной социальной и профессиональной деятельности. | Применять методы и средства физической культуры для обеспечения собственной научной работы в необходимом режиме. |
| ОПК-6 | Знать: правила поиска и содержание основных нормативно-правовых документов регулирующих работу в области обеспечения информационной безопасности информационных систем Уметь:  разрабатывать локальные и объектовые нормативно-правовые документы для обеспечения нормативно-правового сопровождения работ по обеспечению информационной безопасности на предприятии. Владеть: навыками систематизации и выбора необходимой нормативно-правовой информации согласно поставленным задачам в области обеспечения информационной безопасности автоматизированных систем. | Осуществлять систематизацию и выбор необходимой нормативно-правовой информации согласно поставленным задачам в собственной научной работе |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дата выдачи задания: | 10.02.2020 г. | | | | |
| Срок предоставления на кафедру отчета о практике: | | | | 10.06.2020 г. | |
| Руководитель практики от | | |  | |  |
| университета, профессор, д.т.н. | | |  | | Сергеев В.В. |
|  | | | (подпись) | |  |
|  | | |  | |  |
| Задание принял к исполнению | | |  | |  |
| студент группы № 6411-100503D | |  | | | Г.В. Цой |
|  | | (подпись) | | |  |

**Рабочий график (план) проведения практики**

|  |  |  |
| --- | --- | --- |
| Дата (период) | Содержание задания | Результаты практики |
| 05.02.2020 – 15.03.2020 | Изучение существующих методов геометрического согласования серии изображений;  Изучить существующие методы нахождения ключевых точек на изображении. | Изучены существующие методы геометрического согласования серии изображений, а также изучены существующие методы нахождения ключевых точек на изображении. |
| 16.03.2020 – 26.05.2020 | Изучение существующих библиотек, необходимых для геометрического согласования и анализа данным методом; Программная реализация алгоритма геометрического согласования и анализа серии изображений;  Отладка и тестирование разработанной программы. | Изучены существующие программные библиотеки для работы с изображениями, а также осуществлена программная реализация алгоритма согласования серии изображений. Произведена отладка и тестирование программы. |
| 27.05.2020 –  10.06.2020 | Анализ результатов. Оценка наиболее качественного метода детектирования особых точек в терминах СКО | Проанализирован результат работы программы, оценен наиболее качественный метод детектирования ключевых точек на изображении. |

|  |  |  |
| --- | --- | --- |
| Руководитель практики от |  |  |
| университета, профессор, д.т.н. |  | Сергеев В.В. |
|  | (подпись) |  |

О Т Ч Е Т

о выполнении индивидуального задания

по научно-исследовательской работе

# ВВЕДЕНИЕ

При прохождении практики по научно-исследовательской работе, руководителем были поставлены следующие задачи:

* Изучить существующие методы геометрического согласования серии изображений;
* Изучить существующие методы согласования, основанные на нахождении особых точек;
* Изучить существующие библиотеки, необходимые для геометрического согласования и анализа данными методами;
* Программно реализовать алгоритм геометрического согласования, основанный на детектировании особых точек различными способами;
* Проанализировать результаты и оценить наиболее качественный метод детектирования особых точек.

Задания необходимо было выполнять последовательно в течение всего времени практики, предоставляя руководителю промежуточные отчеты.

# ВЫПОЛНЕНИЕ ЗАДАНИЯ

Одним из перспективных подходов к улучшению качества изображений является преобразование нескольких наблюдаемых изображений низкого разрешения (с помощью цифровой обработки сигналов) в одно, изображение с более высоким разрешением.

В последнее время такой подход к повышению разрешения является одним из наиболее активных направлений исследований, и в литературе он называется восстановлением изображений со сверхразрешением. [1]

Сверхразрешение – это способ преобразования нескольких изображений одного объекта для создания изображения большего разрешения. [2]

Данная работа посвящена геометрическому согласованию серии изображений.

Целью и задачей научной работы является определение наиболее эффективного в терминах СКО метода геометрического согласования, основанного на детектировании особых точек.

Предметом исследования стали особые точки изображения.

Темой исследования стало: «Анализ существующих способов реализации геометрического согласования серии изображений методом особых точек».

Структура исследования представляет собой следующие этапы:

* Изучение существующих методов геометрического согласования серии изображений;
* Изучение существующих методов согласования, основанных на нахождении особых точек;
* Изучение существующих программных библиотек, необходимых для геометрического согласования и анализа данными методами;
* Программная реализация алгоритма геометрического согласования на основе методов детектирования ключевых точек различными способами и анализ эффективности преобразований;
* Анализ результатов и оценка наиболее качественного метода детектирования особых точек.

## Изучение существующих методов геометрического согласования серии изображений

По применяемым методологиям алгоритмы геометрического согласования можно разделить на следующие группы [3]:

* Extrinsic Methods

В этом методе к наблюдаемому объекту прикрепляются дополнительные опознавательные точки, которые легко обнаруживаются. Они служат внешними ориентирами для сопоставления функций.

* Surface Methods

В этом методе в роли ориентиров, по которым будет происходить согласование, выступают границы поверхности объекта.

* Moments and Principle Axes Methods

Ортогональные оси, вокруг которых минимизированы моменты инерции, называются основными осями. Два идентичных объекта могут быть точно согласованы путем сопоставления их главных осей.

* Correlation Based Methods

Для регистрации изображений используются методы взаимной и фазовой корреляции.

* Mutual Information Based Methods

Оценивается совместная вероятность интенсивностей сопоставимых вокселей в рассматриваемых изображениях. Воксел представляет собой отдельный объект или точку на равномерно расположенной трехмерной плоскости. Эта точка данных может состоять из таких фрагментов данных как прозрачность или цвет.

* Wavelet Based Methods

В этом методе используется извлечение признаков на основе вейвлетов. Вейвлет – это математическая функция, позволяющая анализировать различные частотные компоненты данных.

* Soft Computing Based Methods

Эти методы производят согласование изображений с помощью нейронных сетей, нечетких множеств и эвристических алгоритмов.

* Feature-based method

Находит и фиксирует отличительные черты того или иного изображения. В качестве отличительных черт могут выступать линейные или точечные объекты, например рёбра или углы куба. Эти отличительные черты называются особыми или ключевыми точками (keypoints).

* Intensity-based method

Используется надежный коэффициент корреляции в качестве меры сходства. Относительно обычного коэффициента корреляции предложенная мера сходства уменьшает влияние шумов.

В рамках данной работы ограничимся более подробным рассмотрением fFeature-based метода и способов его реализации на серии изображений.

## Изучение существующих методов согласования, основанных на нахождении особых точек

Все известные алгоритмы согласования, основанные на feature-based методе, состоят из трёх шагов [4]:

1. Поиск ключевых точек (key-points), по которым в дальнейшем произойдёт согласование изображений;
2. Сопоставление найденных точек (feature-matching) эталонного изображения и обрабатываемого. Также на этом этапе можно произвести дополнительный отсев ошибочных сопоставлений;
3. Трансформация изображения (homography) в соответствии с найденными точками.

В моей работе мерой эффективности работы алгоритма было принято считать среднеквадратичное отклонение (СКО). Метрика вычисляется на основе эталонного изображения и результата работы алгоритма. Сравниваться будут значения метрики для различных детекторов особых точек.

### **Способы поиска ключевых точек**

Особые точки находятся на изображении с помощью детекторов и идентифицируются с помощью дескрипторов. [5]

Детектор – это метод извлечения особых точек из изображения. Детектор обеспечивает инвариантность нахождения одних и тех же особых точек относительно преобразований изображений.

Дескриптор – идентификатор особой точки, выделяющий её из остального множества особых точек. В свою очередь, дескрипторы должны обеспечивать инвариантность нахождения соответствия между особыми точками относительно преобразований изображений.

Существует несколько способов поиска ключевых точек [6]. Наиболее популярными являются:

* Алгоритм scale-invariant feature transform (SIFT);
* Алгоритм speeded up robust features (SURF);
* Алгоритм Binary Robust Independent Elementary Features + Star detector (BRIEF + Star)
* Алгоритм Oriented Features from Accelerated Segment Test and rotated BRIEF (ORB).

### **Сопоставление найденных точек. Отсев неподходящих точек**

Сопоставление точек происходит по дескрипторам. Наиболее распространенные алгоритмы сопоставления – полный перебор или брутфорс (brute-force) и быстрый поиск ближайших значений (FLANN Matcher). [7]

Brute-force алгоритм принимает дескриптор одного объекта из первого набора и сопоставляет его с объектами во втором наборе, находящимися на заданном расстоянии. Максимально похожий найденный дескриптор сопоставляется с исходным.

FLANN Matcher содержит набор алгоритмов, оптимизированных для быстрого поиска ближайших соседей в больших наборах данных. Для больших наборов данных этот метод работает быстрее, чем brute-force.

Не все найденные сопоставления подойдут для дальнейшего исследования. Часть из них будет ложной. Качественный отсев точек обеспечивает алгоритм RANdom SAmple Consensus (RANSAC). На рисунке 1 приведён пример работы алгоритма. Алгоритм должен вписать прямую в набор тестовых данных, при этом вокруг прямой разбросаны шумы. [8]



Рисунок 1 – Пример работы алгоритма RANSAC

### **Трансформация изображения**

После того, как были установлены связи между точками, изображение может быть трансформировано. Для этого сначала проводится оценка гомографии (homography) [9].

Гомография – это перспективное преобразование плоскости, то есть перепроецирование плоскости, с учетом изменения положения и ориентации объекта.

Возможные трансформации:

1. Перемещение (translation);
2. Поворот (rotation);
3. Масштабирование (scaling);
4. Сдвиг (shearing).

Все вместе данные трансформации называются аффинным преобразованием. Также существуют перспективное и нелинейное преобразования.

|  |  |
| --- | --- |
|  |  |
| а) | б) |
|  |  |
| в) | г) |
|  |  |
| д) | е) |

Рисунок 2 – Виды трансформаций. Перемещение (а), поворот (б), масштабирование (в), сдвиг(г), перспективное преобразование (д) и нелинейное преобразование (е)

## Изучение существующих программных средств, необходимых для геометрического согласования и анализа данными методами

Разработка будет происходить на языке Python.

Для работы с изображениями будет использоваться библиотека OpenCV2. Для выполнения математических операций будет использоваться библиотека NumPy.

Выбранная среда разработки – Visual Studio Code 1.45.1

Ввиду того, что библиотека OpenCV2 версии 4.2.0 не поддерживает работу с алгоритмами SIFT и SURF, эти алгоритмы будут реализовываться библиотекой OpenCV2 версии 3.4.2, которая в свою очередь не поддерживается языком Python 3.8, но поддерживается языком Pytrhon 2.7.

Необходимые функции описаны в источнике [10].

## Программная реализация алгоритма геометрического согласования и анализа серии изображений

### **Требования к программе**

Программная реализация алгоритма должна выполнять геометрическое преобразование серии изображений и считать среднеквадратическое отклонение СКО.

Геометрическое согласование должно выполняться feature-based методом алгоритмами SIFT, SURF, ORB и BRIEF + Star.

Работа алгоритмов зависит от количества ключевых точек.

Для методов SIFT и ORB максимального количества точек, которые можно детектировать и обработать нет. Для методов SURF и BRIEF это количество ограничено библиотекой. Максимально возможное количество точек для SURF – 25000, для BRIEF – 128.

Для дальнейшего анализа необходимо каждым методом произвести согласование и рассчитать СКО по количеству точек, приведённых в таблице 1. Для удобства сравнения результатов методы SIFT, SURF и ORB работают в одном диапазоне точек, однако SURF не обрабатывает последние 2 значения. Метод BRIEF возможно протестировать только в диапазоне до 128. Единственное значение, которое совпадает с остальными методами – это 100.

Таблица 1 – Количество ключевых точек для каждого метода

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод детектирования особых точек | Количество ключевых точек для исследуемого метода | | | | | | | | | |
| SIFT | 100 | 200 | 400 | 800 | 1600 | 3200 | 6400 | 12800 | 25600 | 51200 |
| SURF | 100 | 200 | 400 | 800 | 1600 | 3200 | 6400 | 12800 | - | - |
| ORB | 100 | 200 | 400 | 800 | 1600 | 3200 | 6400 | 12800 | 25600 | 51200 |
| BRIEF + Star | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

Разработанная программа на вход должна принимать серию изображений, полученных с помощью статически закреплённой камеры, а на выходе выдавать преобразованные изображения, а также рассчитывать СКО полученных изображений от эталона.

Расчёт СКО производится по формуле 1:

, (1)

где mse – СКО;

m – ширина изображения в пикселях;

n – длина изображения в пикселях;

I(i, j) – значение i,j-ого пикселя результирующего изображения;

K(i, j) – значение i,j-ого пикселя эталонного изображения.

### **Реализация программы**

1. Согласование изображений с использованием алгоритма SIFT использует сопоставление дескрипторов брутфорсом, отсев сопоставлений алгоритмом RANSAC и перспективную трансформацию.
2. Согласование изображений с использованием алгоритма SURF использует сопоставление дескрипторов брутфорсом, отсев сопоставлений алгоритмом RANSAC и перспективную трансформацию.
3. Согласование изображений с использованием алгоритма ORB использует сопоставление дескрипторов методом Brutforce + FLANN, отсев сопоставлений алгоритмом RANSAC и перспективную трансформацию.
4. Согласование изображений с использованием алгоритма BRIEF + Star использует сопоставление дескрипторов брутфорсом, отсев сопоставлений алгоритмом RANSAC и перспективную трансформацию.

Посчитанные СКО выводятся в консоль.

### **Тестирование программы и результат работы**

Разработанная программа обрабатывает тестовую последовательность изображений, состоящих из 50 кадров, снятых на статически неподвижную камеру.

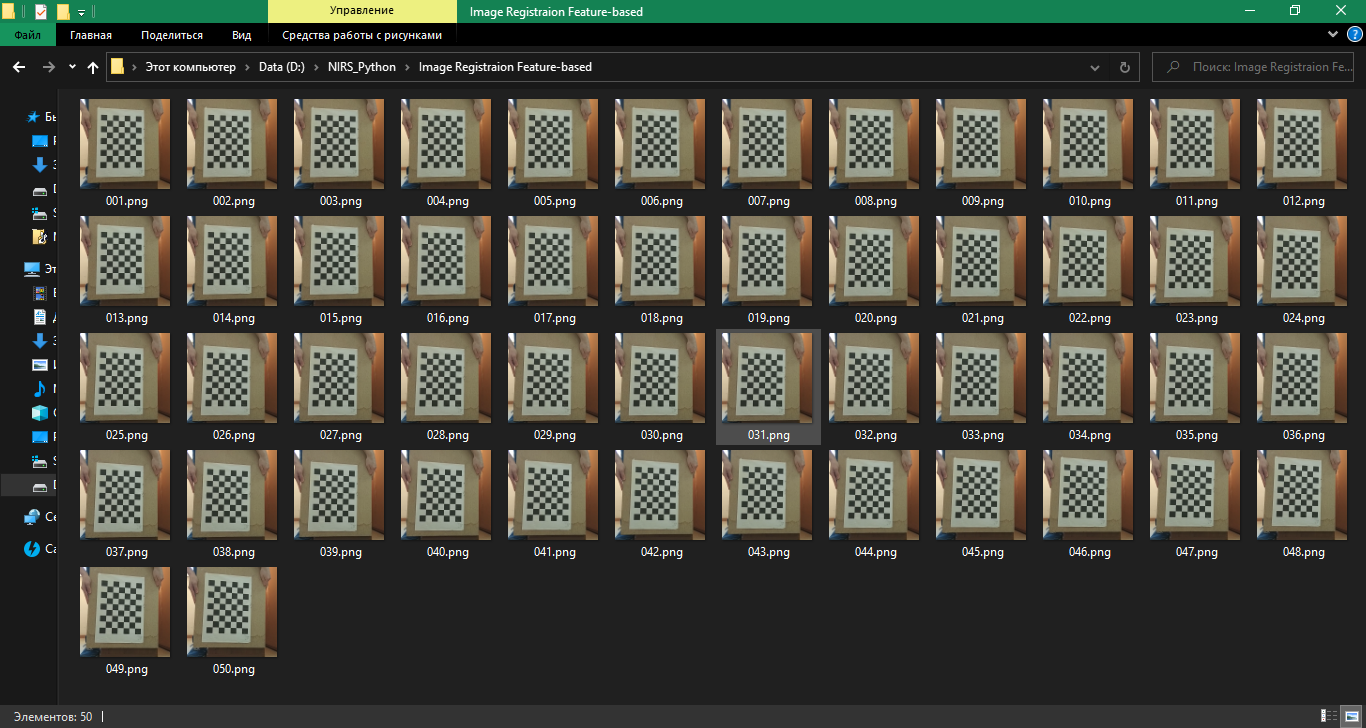


Рисунок 3 – Входная последовательность тестовой серии кадров

Во время работы программы в консоль выводятся посчитанные СКО.

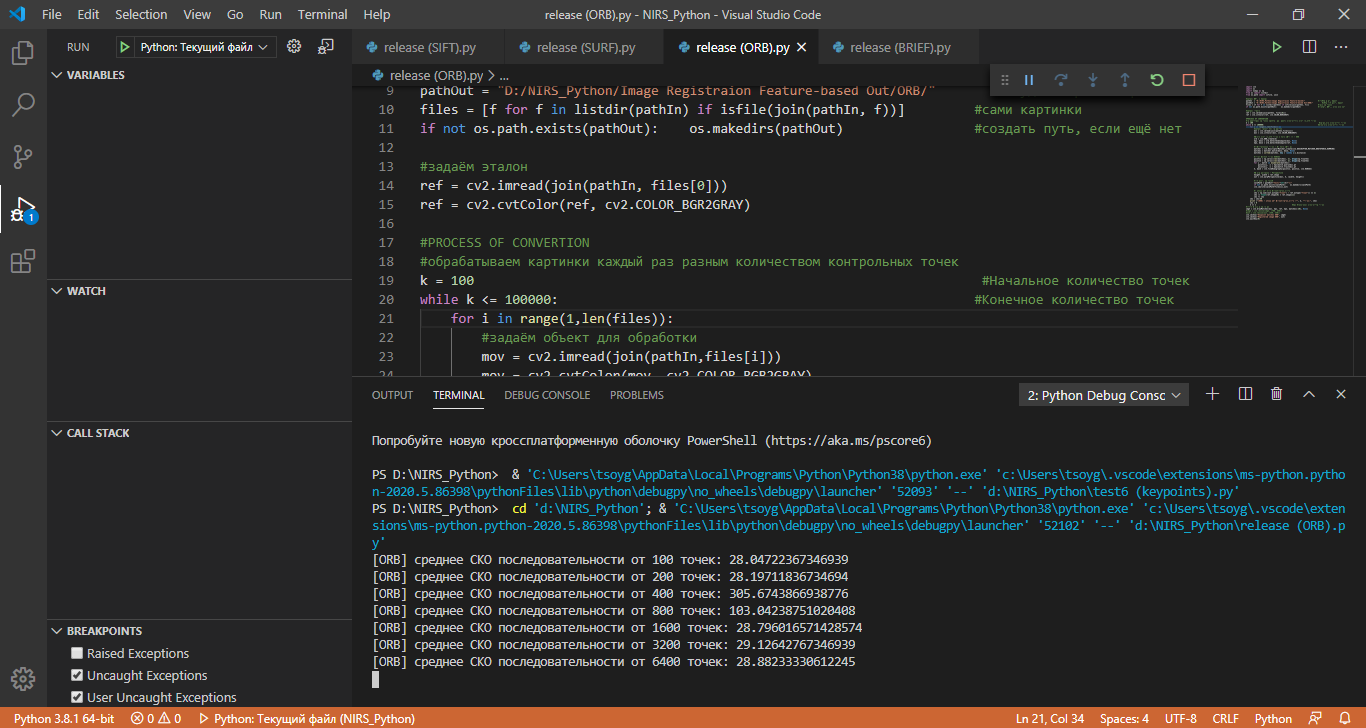


Рисунок 4 – Процесс выполнения программы

По завершению обработки одним алгоритмом, программа выводит пример найденных точек и сопоставление некоторых из них.



Рисунок 5 – Пример детектирования точек и их сопоставления

После завершения работы программы в консоли отображается посчитанные СКО для всех последовательностей, а на диске в указанном каталоге создаются результирующие наборы согласованных кадров. Одна из полученных последовательностей изображена на рисунке 6.

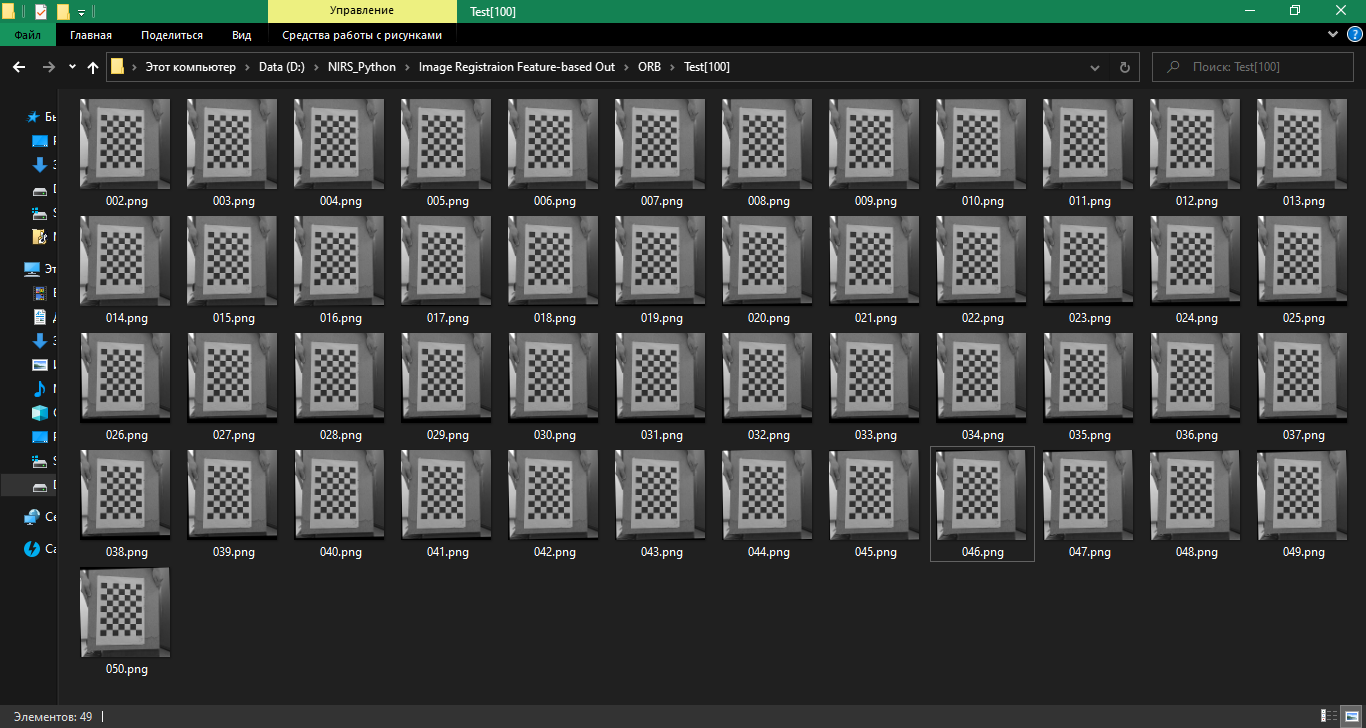


Рисунок 6 – Пример результата работы программы

## Экспериментальное исследование

По формуле (1) было рассчитано СКО для каждой последовательности с разным количеством ключевых точек. Полученные значения приведены в таблицах 2 и 3:

Таблица 2 – СКО для методов SIFT, SURF и ORB

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 200 | 400 | 800 | 1600 | 3200 | 6400 | 12800 | 25600 | 51200 |
| SIFT | 28.93 | 28.94 | 28.64 | 28.94 | 29.12 | 29.12 | 29.12 | 29.12 | 29.12 | 29.12 |
| SURF | 29.27 | 29.12 | 29.12 | 29.83 | 29.16 | 29.01 | 29.14 | 29.46 | - | - |
| ORB | 28.05 | 28.2 | 305.67 | 103.04 | 28.8 | 29.13 | 28.88 | 28.88 | 28.88 | 28.88 |

Таблица 3 – СКО для метода BRIEF + Star

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| BRIEF + Star | 28.89 | 28.95 | 29.15 | 29.65 | 29.42 | 29.42 | 29.16 | 29.16 | 29.16 | 29.9 |

По данным из таблиц были построены графики для каждого метода, отражающие зависимость СКО от количества детектируемых точек. Полученные графики представлены на рисунках 7, 8, 9, 10.

Рисунок 7 – График значений СКО для метода SIFT

Рисунок 8 – График значений СКО для метода SURF

Рисунок 9 – График значений СКО для метода ORB

Рисунок 10 – График значений СКО для метода BRIEF + Star

Рисунок 11 – График значений СКО для трёх методов

По полученным графикам видно, что среди всех алгоритмов, алгоритм ORB обладает наименьшим СКО, равным 28,05. Этот же алгоритм имеет наибольшее значение, равное 305,67. Следовательно данный алгоритм нельзя считать стабильным.

Алгоритм SIFT имеет второе по счёту минимальное значение СКО – 28,64, и в целом выдаёт стабильные результаты для разного количества детектируемых точек.

Для методов SIFT и ORB видно, что при достижении некоторого конкретного количества обнаруживаемых точек, результирующее СКО перестаёт изменяться. Для различных серий кадров, подаваемых на входе программы, это количество будет всегда разным.

## Выводы и результаты

В этом семестре я занимался геометрическим согласованием серии изображений на основе метода детектирования особых точек. Мной были изучены существующие методы нахождения особых точек и разработана программа, реализующая алгоритм согласования изображений несколькими методами. В результате работы программы были составлены графики зависимости среднеквадратических отклонений от количества детектируемых особых точек.

Из графиков видно, что ни один алгоритм не имеет функциональной зависимости СКО от количества точек. Это означает, что не всегда детектирование большего числа точек даст лучший результат.

Как и ожидалось, feature-based метод не даёт идеального согласования изображений, однако за короткий промежуток времени и небольших вычислительных затратах выдаёт достаточно приемлемый результат.

Метод, основанный на детектировании особых точек, не подходит для задач получения сверхразрешения, так как он даёт хоть и малую, но ощутимую погрешность в согласовании серии изображений.

Наилучшим в терминах СКО алгоритмом по детектированию ключевых точек является алгоритм SIFT.

# ЗАКЛЮЧЕНИЕ

В течение практики по научно-исследовательской работе успешно выполнены поставленные задачи: изучены существующие методы геометрического согласования серии изображений; изучены существующие методы нахождения особых точек на изображении; изучены существующие библиотеки, необходимые для геометрического согласования и анализа данными методами; программно реализован алгоритм геометрического согласования, основанный на детектировании особых точек различными способами; проанализированы результаты и произведена оценка наиболее качественного метода детектирования особых точек в терминах СКО.

# ПРИЛОЖЕНИЕ А

Полный листинг софта для автоматизации экспериментальных исследований

release (SIFT).py

# -\*- coding: utf-8 -\*-

import cv2

import os

import numpy as np

from os import listdir

from os.path import isfile, join

#задаём путь к файлам

pathIn  = "D:/NIRS\_Python/Image Registraion Feature-based/"             #откуда брать файлы

pathOut = "D:/NIRS\_Python/Image Registraion Feature-based Out/SIFT/"    #куда сохранять файлы

files = [f for f in listdir(pathIn) if isfile(join(pathIn, f))]         #сами картинки

if not os.path.exists(pathOut):    os.makedirs(pathOut)                 #создать путь, если ещё нет

#задаём эталон

ref = cv2.imread(join(pathIn, files[0]))

ref = cv2.cvtColor(ref, cv2.COLOR\_BGR2GRAY)

#PROCESS OF CONVERTION

#обрабатываем картинки каждый раз разным количеством контрольных точек

k = 100                                                                 #Начальное количество точек

while k <= 100000:                                                      #Конечное количество точек

    for i in range(1,len(files)):

        #задаём объект для обработки

        mov = cv2.imread(join(pathIn,files[i]))

        mov = cv2.cvtColor(mov, cv2.COLOR\_RGB2GRAY)

        #поиск контрольных точек и дескрипторов - SIFT

        sift = cv2.xfeatures2d.SIFT\_create(k)

        kp1, des1 = sift.detectAndCompute(mov, None)

        kp2, des2 = sift.detectAndCompute(ref, None)

        #сопоставление точек - по дескрипторам

        matcher = cv2.BFMatcher(cv2.NORM\_L2, crossCheck=False)

        matches = matcher.match(des1, des2)

        matches = sorted(matches, key = lambda x:x.distance)

        #отсев плохих точек RANSAC

        points1 = np.zeros((len(matches), 2), dtype=np.float32)

        points2 = np.zeros((len(matches), 2), dtype=np.float32)

        for j , match in enumerate(matches):

            points1[j, :] = kp1[match.queryIdx].pt

            points2[j, :] = kp2[match.trainIdx].pt

        h, mask = cv2.findHomography(points1, points2, cv2.RANSAC)

        #преобразование - Perspective

        height, width = ref.shape

        out = cv2.warpPerspective(mov, h, (width, height))

        #сохранить картикну

        savePath = pathOut+"Test["+str(k)+"]/"

        if not os.path.exists(savePath):    os.makedirs(savePath)

        cv2.imwrite(savePath+files[i],out)

        #среднее СКО k-ой последовательности

        sko = np.sum((out.astype("float") - ref.astype("float")) \*\* 2)

        sko /= float(ref.shape[0] \* ref.shape[1])

        sko += sko

    sko = sko/50

    print "[SIFT] average standard deviation of the sequence from ", k, "points:", sko

    sko = 0

    k \*= 2                                    #Шаг увеличения количества точек

#вывод примера на экран

img3 = cv2.drawMatches(mov, kp1, ref, kp2, matches[:20], None)

cv2.imshow("Keypoint matches SIFT", img3)

cv2.imshow("Registered image SIFT", out)

cv2.waitKey(0)

release (SURF).py

# -\*- coding: utf-8 -\*-

import cv2

import os

import numpy as np

from os import listdir

from os.path import isfile, join

#задаём путь к файлам

pathIn  = "D:/NIRS\_Python/Image Registraion Feature-based/"             #откуда брать файлы

pathOut = "D:/NIRS\_Python/Image Registraion Feature-based Out/SURF/"    #куда сохранять файлы

files = [f for f in listdir(pathIn) if isfile(join(pathIn, f))]         #сами картинки

if not os.path.exists(pathOut):    os.makedirs(pathOut)                 #создать путь, если ещё нет

#задаём эталон

ref = cv2.imread(join(pathIn, files[0]))

ref = cv2.cvtColor(ref, cv2.COLOR\_BGR2GRAY)

#PROCESS OF CONVERTION

#обрабатываем картинки каждый раз разным количеством контрольных точек

k = 100                                                                 #Начальное количество точек

while k <= 100000:                                                      #Конечное количество точек

    for i in range(1,len(files)):

        #задаём объект для обработки

        mov = cv2.imread(join(pathIn,files[i]))

        mov = cv2.cvtColor(mov, cv2.COLOR\_RGB2GRAY)

        #поиск контрольных точек и дескрипторов - SURF

        sift = cv2.xfeatures2d.SURF\_create(k)

        kp1, des1 = sift.detectAndCompute(mov, None)

        kp2, des2 = sift.detectAndCompute(ref, None)

        #сопоставление точек - по дескрипторам

        matcher = cv2.BFMatcher(cv2.NORM\_L2, crossCheck=False)

        matches = matcher.match(des1, des2)

        matches = sorted(matches, key = lambda x:x.distance)

        #отсев плохих точек RANSAC

        points1 = np.zeros((len(matches), 2), dtype=np.float32)

        points2 = np.zeros((len(matches), 2), dtype=np.float32)

        for j , match in enumerate(matches):

            points1[j, :] = kp1[match.queryIdx].pt

            points2[j, :] = kp2[match.trainIdx].pt

        h, mask = cv2.findHomography(points1, points2, cv2.RANSAC)

        #преобразование - Perspective

        height, width = ref.shape

        out = cv2.warpPerspective(mov, h, (width, height))

        #сохранить картикну

        savePath = pathOut+"Test["+str(k)+"]/"

        if not os.path.exists(savePath):    os.makedirs(savePath)

        cv2.imwrite(savePath+files[i],out)

        #среднее СКО k-ой последовательности

        sko = np.sum((out.astype("float") - ref.astype("float")) \*\* 2)

        sko /= float(ref.shape[0] \* ref.shape[1])

        sko += sko

    sko = sko/50

    print "[SURF] average standard deviation of the sequence from ", k, "points:", sko

    sko = 0

    k \*= 2                                    #Шаг увеличения количества точек

#вывод примера на экран

img3 = cv2.drawMatches(mov, kp1, ref, kp2, matches[:20], None)

cv2.imshow("Keypoint matches SURF", img3)

cv2.imshow("Registered image SURF", out)

cv2.waitKey(0)

release (ORB).py

import cv2

import os

import numpy as np

from os import listdir

from os.path import isfile, join

#задаём путь к файлам

pathIn  = "D:/NIRS\_Python/Image Registraion Feature-based/"             #откуда брать файлы

pathOut = "D:/NIRS\_Python/Image Registraion Feature-based Out/ORB/"         #куда сохранять файлы

files = [f for f in listdir(pathIn) if isfile(join(pathIn, f))]         #сами картинки

if not os.path.exists(pathOut):    os.makedirs(pathOut)                 #создать путь, если ещё нет

#задаём эталон

ref = cv2.imread(join(pathIn, files[0]))

ref = cv2.cvtColor(ref, cv2.COLOR\_BGR2GRAY)

#PROCESS OF CONVERTION

#обрабатываем картинки каждый раз разным количеством контрольных точек

k = 100                                                                  #Начальное количество точек

while k <= 100000:                                                      #Конечное количество точек

    for i in range(1,len(files)):

        #задаём объект для обработки

        mov = cv2.imread(join(pathIn,files[i]))

        mov = cv2.cvtColor(mov, cv2.COLOR\_RGB2GRAY)

        #поиск контрольных точек и дескрипторов - ORB

        orb = cv2.ORB\_create(k)

        kp1, des1 = orb.detectAndCompute(mov, None)

        kp2, des2 = orb.detectAndCompute(ref, None)

        #сопоставление точек - по дескрипторам

        matcher = cv2.DescriptorMatcher\_create(cv2.DESCRIPTOR\_MATCHER\_BRUTEFORCE\_HAMMING)

        matches = matcher.match(des1, des2, None)

        matches = sorted(matches, key = lambda x:x.distance)

        #отсев плохих точек RANSAC

        points1 = np.zeros((len(matches), 2), dtype=np.float32)

        points2 = np.zeros((len(matches), 2), dtype=np.float32)

        for j , match in enumerate(matches):

            points1[j, :] = kp1[match.queryIdx].pt

            points2[j, :] = kp2[match.trainIdx].pt

        h, mask = cv2.findHomography(points1, points2, cv2.RANSAC)

        #преобразование - Perspective

        height, width = ref.shape

        out = cv2.warpPerspective(mov, h, (width, height))

        #сохранить картикну

        savePath = pathOut+"Test["+str(k)+"]/"

        if not os.path.exists(savePath):    os.makedirs(savePath)

        cv2.imwrite(savePath+files[i],out)

        #среднее СКО k-ой последовательности

        sko = np.sum((out.astype("float") - ref.astype("float")) \*\* 2)

        sko /= float(ref.shape[0] \* ref.shape[1])

        sko += sko

    sko = sko/50

    print ("[ORB] среднее СКО последовательности от", k, "точек:", sko)

    sko = 0

    k \*= 2                                    #Шаг увеличения количества точек

#вывод примера на экран

img3 = cv2.drawMatches(mov, kp1, ref, kp2, matches[:20], None)

#img3 = cv2.resize(img3, (700, 350))

#out = cv2.resize(out, (350, 350))

cv2.imshow("Keypoint matches ORB", img3)

cv2.imshow("Registered image ORB", out)

cv2.waitKey(0)

release (BRIEF).py

# -\*- coding: utf-8 -\*-

import cv2

import os

import numpy as np

from os import listdir

from os.path import isfile, join

#задаём путь к файлам

pathIn  = "D:/NIRS\_Python/Image Registraion Feature-based/"             #откуда брать файлы

pathOut = "D:/NIRS\_Python/Image Registraion Feature-based Out/BRIEF + STAR/"   #куда сохранять файлы

files = [f for f in listdir(pathIn) if isfile(join(pathIn, f))]         #сами картинки

if not os.path.exists(pathOut):    os.makedirs(pathOut)                 #создать путь, если ещё нет

#задаём эталон

ref = cv2.imread(join(pathIn, files[0]))

ref = cv2.cvtColor(ref, cv2.COLOR\_BGR2GRAY)

#PROCESS OF CONVERTION

#обрабатываем картинки каждый раз разным количеством контрольных точек

k = 10                                                                  #Начальное количество точек

while k <= 100:                                                      #Конечное количество точек

    for i in range(1,len(files)):

        #задаём объект для обработки

        mov = cv2.imread(join(pathIn,files[i]))

        mov = cv2.cvtColor(mov, cv2.COLOR\_RGB2GRAY)

        #поиск контрольных точек и дескрипторов - STAR + BRIEF

        star1 = cv2.xfeatures2d.StarDetector\_create(k)                    #STAR

        star2 = cv2.xfeatures2d.StarDetector\_create(k)                    #STAR

        brief1 = cv2.xfeatures2d.BriefDescriptorExtractor\_create()        #BRIEF

        brief2 = cv2.xfeatures2d.BriefDescriptorExtractor\_create()        #BRIEF

        kp1 = star1.detect(mov,None)

        kp2 = star2.detect(ref,None)

        kp1, des1 = brief1.compute(mov, kp1)

        kp2, des2 = brief2.compute(ref, kp2)

        #сопоставление точек - по дескрипторам

        matcher = cv2.BFMatcher(cv2.NORM\_L2, crossCheck=False)

        matches = matcher.match(des1, des2)

        matches = sorted(matches, key = lambda x:x.distance)

        #отсев плохих точек RANSAC

        points1 = np.zeros((len(matches), 2), dtype=np.float32)

        points2 = np.zeros((len(matches), 2), dtype=np.float32)

        for j , match in enumerate(matches):

            points1[j, :] = kp1[match.queryIdx].pt

            points2[j, :] = kp2[match.trainIdx].pt

        h, mask = cv2.findHomography(points1, points2, cv2.RANSAC)

        #преобразование - Perspective

        height, width = ref.shape

        out = cv2.warpPerspective(mov, h, (width, height))

        #сохранить картикну

        savePath = pathOut+"Test["+str(k)+"]/"

        if not os.path.exists(savePath):    os.makedirs(savePath)

        cv2.imwrite(savePath+files[i],out)

        #среднее СКО k-ой последовательности

        sko = np.sum((out.astype("float") - ref.astype("float")) \*\* 2)

        sko /= float(ref.shape[0] \* ref.shape[1])

        sko += sko

    sko = sko/50

    print ("[BRIEF] average standard deviation of the sequence from ", k, "points:", sko)

    sko = 0

    k += 10                                    #Шаг увеличения количества точек

#вывод примера на экран

img3 = cv2.drawMatches(mov, kp1, ref, kp2, matches[:20], None)

cv2.imshow("Keypoint matches BRIEF", img3)

cv2.imshow("Registered image BRIEF", out)

cv2.waitKey(0)

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Sung Cheol Park, Min Kyu Park, Moon Gi Kang Super-resolution image reconstruction: a technical overview [Текст] / Sung Cheol Park, Min Kyu Park, Moon Gi Kang // IEEE Signal Processing Magazine. 2003 г. №20 – 21-36 с.

2 Сверхразрешение [Электронный ресурс]/ URL: <http://www.hi-def.ru/sverkhrazreshenie>

3 Image Registration Techniques: A Survey [Текст] / Sayan Nag [Электронный ресурс] URL: <https://arxiv.org/ftp/arxiv/papers/1712/1712.07540.pdf>

4 Zhang Z, Han D, Dezert J, Yang Y. A New Image Registration Algorithm Based on Evidential Reasoning [Текст] / Zhang Z, Han D, Dezert J, Yang Y. // Sensors (Basel). 2019 г. doi:10.3390/s19051091

5 David G. Lowe Distinctive Image Features from Scale-Invariant Keypoints [Текст] / David G. Lowe // Computer Science Department University of British Columbia Vancouver, B.C., Canada. 2004 г.

6 Karami E. Prasad S. Shehata M. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. [Текст] // Conference: 2015 Newfoundland Electrical and Computer Engineering Conference, At St. John's, Canada, Ноябрь 2015 г. 6 с.

7 Feature Matching [Электронный ресурс] // URL: <https://docs.opencv.org/master/dc/dc3/tutorial_py_matcher.html>

8 O. Chum, J. Matas Randomized RANSAC with Td,d test [Текст] / O. Chum, J. Matas // Center for Machine Perception, Dept. of Cybernetics, CTU Prague, Karlovo n´am 13, CZ 121 35 2CVSSP, University of Surrey, Guildford GU2 7XH, UK

9 Basic concepts of the homography explained with code [Электронный ресурс] // URL: <https://docs.opencv.org/master/d9/dab/tutorial_homography.html>

10 OpenCV modules [Электронный ресурс] // URL: <https://docs.opencv.org/master/>

**ОТЗЫВ О ПРОХОЖДЕНИИ ПРАКТИКИ**

Сроки прохождения практики: с 10.02.2020 г. по 10.06.2020 г.

по направлению подготовки 10.05.03 Информационная безопасность автоматизированных систем

(уровень академического специалитета)

направленность (профиль) «Обеспечение информационной безопасности распределенных информационных систем»

студентом группы № 6411-100503D Г.В. Цоем

|  |  |  |
| --- | --- | --- |
| № п/п | Критерии оценки | Оценка  (по 5-балльной шкале) |
| 1. | Общая систематичность и ответственность работы в ходе практики |  |
| 2. | Достижение планируемых результатов практики |  |
| 3. | Корректность в сборе, анализе и интерпретации представляемых данных |  |
| 4. | Степень личного участия и самостоятельности практиканта в представляемом отчете о практике |  |
| 5. | Качество оформления отчетной документации |  |
|  | **ИТОГОВАЯ ОЦЕНКА[[1]](#footnote-1)\*** |  |

1. \* Итоговая оценка выставляется как средняя арифметическая оценок по пяти критериям оценки [↑](#footnote-ref-1)