

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ «МИСИС»

Институт компьютерных наук  
Кафедра инженерной кибернетики

### **Курсовая работа**

по дисциплине «Методы и средства обработки изображений» на тему  
«Алгоритм определения состояния окна в помещении МИСИС»

Выполнил:

студент гр. БПМ-21-1 Усков Н. А.

Проверил:

доцент кафедры ИК, к.т.н. Полевой Д.В.

Москва, 2024

## Содержание

<b>1. Описание задачи .....</b>	<b>3</b>
<b>1.1. Описание контекста задачи.....</b>	<b>3</b>
<b>1.2. Цель исследования.....</b>	<b>3</b>
<b>1.3. Задачи исследования .....</b>	<b>3</b>
<b>1.4. Допущения в задаче .....</b>	<b>3</b>
<b>2. Описание решения .....</b>	<b>4</b>
<b>2.1. Общее описание подхода к решению .....</b>	<b>4</b>
<b>2.2. Описание метода сравнения absdiff .....</b>	<b>6</b>
<b>2.3. Описание метода сравнения SIFT .....</b>	<b>9</b>
<b>2.4. Настройка параметров.....</b>	<b>10</b>
<b>3. Описание измерения качества.....</b>	<b>11</b>
<b>3.1. Описание выбранных метрик качества .....</b>	<b>11</b>
<b>3.2. Обоснование выбора метрик.....</b>	<b>12</b>
<b>4. Описание результатов.....</b>	<b>12</b>
<b>4.1. Сравнение эффективности методов .....</b>	<b>12</b>
<b>5. Инструкция по использованию кода .....</b>	<b>13</b>
<b>5.1. Требования для установки и тестирования .....</b>	<b>13</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>13</b>
<b>ССЫЛКИ НА РЕСУРСЫ.....</b>	<b>14</b>

## **1. Описание задачи**

### **1.1. Описание контекста задачи**

Задача появилась ввиду потребности сотрудников университета в отделе технической поддержки в понимании состояния окна. Если окно в аудитории открыто, то им необходимо пойти и закрыть его. В настоящий момент эта задача решается простым анализом видео с камеры (сам человек проводит анализ). В рамках данной работы будут рассмотрены несколько подходов к реализации алгоритма определения состояния окна. Окно считается открытым только при наличии расстояния между рамой и ручки окна, также хочется уточнить, что смена состояния ручки не влияет на открытость окна.

### **1.2. Цель исследования**

Целью исследования является создание программного инструмента, способного определять состояние окна на основе анализа поступающих фотографий с наличие «эталонного» вида закрытого окна.

### **1.3. Задачи исследования**

- Разработать алгоритмы для обработки изображений и сравнения их путем абсолютного сравнения и методом SIFT.
- Провести эксперименты на тестовых данных для проверки точности и эффективности разработанных методов.
- Оценить результаты работы каждого метода с использованием соответствующих метрик каждого алгоритма.
- Сравнить эффективность двух методов и выбрать наиболее подходящий для задачи определения состояния окна.
- Написать инструкцию по использованию разработанного кода.

### **1.4. Допущения в задаче**

Для упрощения поставленного задания, а также приближения условий близко к условиям реальной ситуации были сделаны следующие допущения:

- Камера и рамка окна статичны
- Камера снимает в одном ракурсе и в поле зрения находится единственное окно
- Свет в помещении всегда горит
- Жалюзи всегда подняты
- Мы знаем как выглядит эталонно закрытое окно

## **2. Описание решения**

### **2.1. Общее описание подхода к решению**

Для решения задачи определения состояния окна используется подход, основанный на сравнении текущего изображения с "эталонным" изображением закрытого окна. Ниже приведено подробное описание этого подхода:

#### **1. Использование эталонного изображения закрытого окна:**

Имеется "эталонное" изображение, которое представляет собой снимок закрытого окна в стандартных условиях освещения и ракурса съемки (см. рис. 1)



Рис. 1. «Закрытое окно»

#### **2. Выделение рамки окна на эталонном и текущем изображениях:**

На эталонном изображении известно положение и форма рамки окна. Это позволяет выделить область, которая соответствует рамке окна (см. рис.2.1).



Рис.2.1 «Вырезанное окно эталон»

То же самое выполняется и для текущего изображения: по координатам окна мы обрезаем изображение, чтобы отсечь все лишнее. (см рис.2.2)



Рис.2.2 «Вырезанное окно тест»

### 3. Сравнение изображений:

Используются два метода сравнения: метод абсолютной разности (absdiff) и метод SIFT. Метод абсолютной разности вычисляет абсолютную разницу между пикселями в соответствующих точках изображений. Метод SIFT (Scale-Invariant Feature Transform) используется для извлечения ключевых точек и их описаний, после чего выполняется сопоставление и нахождение преобразования (гомографии) между ключевыми точками.

## **2.2. Описание метода сравнения absdiff**

Метод абсолютной разности (absdiff) выполняет следующие шаги:

### **1. Преобразование изображений в монохромное пространство:**

Сначала исходные цветные изображения преобразуются в монохромные изображения (оттенки серого) с использованием функции `cv::cvtColor()` (см. Рис.3).



Рис.3 «Ч/б закрытое окно»

### **2. Размытие изображений:**

На полученных монохромных изображениях выполняется размытие для сглаживания шумов (см Рис.4) с помощью функции `cv::GaussianBlur()`. Это уменьшает высокочастотные компоненты изображений, что помогает сделать абсолютную разность более устойчивой к шумам.



Рис.4 «Сглаженное окно»

### 3. **Вычисление абсолютной разности:**

Для каждой соответствующей пары пикселей изображений `blurredImg1` и `blurredImg2` вычисляется абсолютная разность (см. Рис.5) с использованием функции `cv::absdiff()`.



Рис.5 «Разность эталона и теста»

При вычислении используется следующая формула:

$$D(x, y) = | G1(x, y) - G2(x, y) |$$

Где

G1, G2 – монохромные изображения

#### 4. **Нормализация и оценка сходства:**

Полученная матрица diff содержит абсолютные разности между пикселями. Затем эти значения нормализуются, разделяя каждое значение на максимально возможную разность (255 для монохромного изображения в 8-битном пространстве). Нормализованные значения используются для оценки степени сходства изображений.

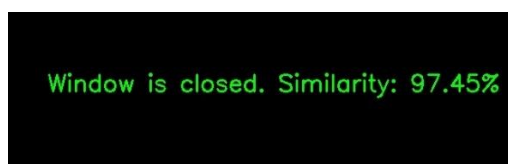


Рис.6 «Результат сравнения»

Чем значение ближе к 1, тем больше сходство между изображениями. Нормализация происходит по следующему принципу:

$$S = 1 - \frac{\sum D(x, y)}{N \times 255}$$

где

N - общее количество пикселей на изображении (ширина x высота), а 255 - максимальное значение яркости пикселя в монохромном изображении.



### 2.3. Описание метода сравнения SIFT

Метод SIFT (Scale-Invariant Feature Transform) выполняет следующие шаги:

#### 1. Извлечение ключевых точек и описаний:

На монохромных изображениях с использованием метода SIFT извлекаются ключевые точки и их локальные описания. Ключевые точки представляют собой особенные точки на изображении, которые выделяются благодаря своей уникальности и стабильности относительно изменений масштаба, поворотов и освещения.

#### 2. Сопоставление ключевых точек:

Производится сопоставление ключевых точек между эталонным (закрытым) и текущим (анализируемым) изображением (см. Рис.7). Это делается с использованием метода cv::BFMatcher, который находит наилучшие соответствия между описаниями ключевых точек двух изображений.



Рис.7 «Сопоставление ключевых точек»

#### 3. Оценка сходства с помощью RANSAC:

Применяется метод RANSAC (Random Sample Consensus) для нахождения гомографического преобразования между наборами ключевых точек. RANSAC позволяет отфильтровать ложные сопоставления и оценить наилучшее

гомографическое преобразование, которое наиболее точно объясняет набор сопоставлений.

#### 4. Вычисление среднего смещения:

После получения набора "инлайнеров" (сопоставлений, которые были признаны как корректные после применения RANSAC), вычисляется среднее смещение между соответствующими ключевыми точками (см. Рис.8).



Рис.8 «Оценка среднего смещения»

Это смещение используется для оценки степени сходства между изображениями и рассчитывается по следующей формуле:

$$d_{\text{avg}} = \frac{1}{N} \sum_{i=1}^N \|p_i' - Hp_i\|$$

Где

$N$  - количество инлайнеров,

$p_i$  и  $p_i'$  - координаты ключевых точек на эталонном и текущем изображениях,

$H$  - гомографическое преобразование между изображениями.

#### 2.4. Настройка параметров

- Параметры размера окна для вырезания области с изображением окна следующие:

Координаты левого угла  $(x, y) = (2400, 2500)$ , ширина 1600 пик., высота 2680 пик.

- Параметры размытия для сглаживания изображений перед вычислением абсолютной разности: Размер ядра (5, 5), среднее значение 1.5

- Пороговые значения для сравнения результатов методов абсолютной разности и SIFT:

Для абсолютной разности порог threshold равен 0,95, а для сравнения через SIFT среднее расстояние пикселей не должно превышать 10.

### 3. Описание измерения качества

#### 3.1. Описание выбранных метрик качества

Для оценки качества методов определения состояния окна на основе абсолютной разности и метода SIFT были выбраны следующие метрики:

1. Абсолютная разница: для метода абсолютной разности основной метрикой качества является сходство ( $S$ ), вычисленное как:

$$S = 1 - \frac{\sum D(x, y)}{N \times 255}$$

где:

$D(x, y)$  - абсолютная разница для пикселя  $(x, y)$ ,

$N$  - общее количество пикселей в изображении,

255 - максимальное значение яркости пикселя в монохромном изображении.

Чем ближе значение  $S$  к 1, тем выше сходство между изображениями.

2. Метод SIFT: для метода SIFT используются следующие метрики:

Среднее смещение ключевых точек ( $avgDisplacement$ ): Эта метрика измеряет среднее расстояние между соответствующими ключевыми точками на текущем и эталонном изображениях. Формула для вычисления среднего смещения:

$$avgDisplacement = \frac{\sum displacement_i}{numMatches}$$

где

$displacement_i$  - смещение для каждой пары соответствующих ключевых точек,

$numMatches$  - общее количество сопоставленных пар ключевых точек.

### 3.2. Обоснование выбора метрик


Выбор метрик основан на нескольких критериях: Простота реализации, соответствие методу, легкость получения результата. Метрика для каждого метода является дополняющим фактором. Например, абсолютная разница при нормировке явно показывает в долевым отношении насколько 2 картинки схожи между собой.

## 4. Описание результатов

### 4.1. Сравнение эффективности методов

Для сравнения методов и оценки результата было собрано несколько фотографий разного состояния окна. Всего состояний было 4: полностью закрытое, закрытое с повернутой ручкой, открытое на проветривание, открытое полностью. Всего было собрано 42 фотографии (7 из них представлено в репозитории на гитхаб), остальные же отличались лишь углом поворота или степенью открытости. Оценка каждого метода проводилась из их метрик и кол-ву верно определенных состояний.

Метод абсолютной разницы для 7 фотографий не ошибся ни разу при пороге 95%, а вот для 42 фотографий верно названы были 40 из них. Данный метод очень сильно зависит от обработки яркости картинки и угла поворота камеры, т. к. при изменении этих параметров разность изображений будет давать большие числа. Далее приведу результат определения открытого окна (см. Рис.9):



Window is open. Similarity: 85.60%

Рис.9 «Результаты определения открытого состояния ABSDiff»

Метод SIFT при пороге в 10 пикселей для среднего значения ошибся 1 раз для 7 фотографий, определив едва открытое окно как закрытое. А вот при 42 фотографиях показал большую ошибку, чем 1-й метод: 37 верно классифицированных состояний. Такая погрешность была вызвана выбором

ключевых точек, т. к. при изменении состояния окна, могут не измениться точки, находящиеся вне рамы или «внутри» стекла. Повышение точности данного метода может заключаться в обработке заднего фона и корректировке алгоритма выбора значимых точек (расположить их на оконной раме). Далее приведу результат определения открытого окна (см. Рис.10):



Window is open. Average displacement: 13.30 pixels.

Рис.10 «Результат определения открытого окна SIFT»

## 5. Инструкция по использованию кода

### 5.1. Требования для установки и тестирования

1. Стаке минимальной версии 3.20.

2. Язык C++17.

3. Менеджер зависимостей vcpkg.

- Для macOS или Linux:

```
git clone https://github.com/Microsoft/vcpkg.git  
./vcpkg/bootstrap-vcpkg.sh  
vcpkg integrate install
```

- Для Windows:

```
git clone https://github.com/Microsoft/vcpkg.git  
.\vcpkg\bootstrap-vcpkg.bat  
vcpkg integrate install
```

4. Библиотека `opencv`, которую можно скачать с помощью следующей команды: `vcpkg install opencv`

## ЗАКЛЮЧЕНИЕ

Данная курсовая работа была посвящена разработке алгоритма классификации состояния окна(открыто/закрыто) и сравнению разных подходов для реализации. Основная цель исследования заключалась в создании алгоритма, который позволяет сравнить фотографии и расстояние на них.

В ходе работы были изучены и реализованы следующие методы:

- Метод вычитания изображений при помощи OpenCV
- Метод сравнения расстояния между точками при помощи функции SIFT
- Методы обработки изображений, бинаризация, размытие, обнаружение краев методом Canny
- И другие базовые методы языка C++ и библиотеки OpenCV, такие как реализация классов, разметка и создание прямоугольников, исследование параметров изображений и т.п.

Проведенное исследование показало, что предложенные методы успешно решают поставленную задачу и могут быть использованы для классификации состояния окон. Таким образом, данная работа не только теоретически обосновала возможность эффективного решения задачи, но и продемонстрировала практическую реализацию метода при определенных условиях.

#### **ССЫЛКИ НА РЕСУРСЫ**

1. Проект: сайт. – URL: <https://github.com/NikitaUskov/misis2024s-21-01-uskov-n-a>
2. CMAKE: сайт. – URL: <https://cmake.org/>
3. Vcpkg: сайт. – URL: <https://vcpkg.io/en/index.html>