

## РЕФЕРАТ

**Отчет:** 17 страницы, 1 рисунка, 0 таблицы, 10 источников, 1 приложение.

*ОБРАБОТКА ИЗОБРАЖЕНИЙ, ДЕСКРИПТОРЫ ТОЧЕК, DAISY, PYTHON.*

В данной работе рассмотрены алгоритмы построения дескрипторов в задачах обработки изображений.

Цель работы – разработка устойчивого к пространственным преобразованиям алгоритма построения дескрипторов.

Рассмотрены принципы построения дескрипторов изображений, создана программная реализация с применением алгоритма DAISY, проведено экспериментальное исследование работы алгоритма.

## СОДЕРЖАНИЕ

Введение . . . . .	5
1 Исследование алгоритмов построения дескрипторов изображений . . . . .	6
1.1 Постановка задачи . . . . .	6
1.2 Существующие алгоритмы построения дескрипторов . . . . .	8
2 Разработка алгоритма построения дескрипторов изображений . . . . .	9
2.1 Алгоритм построения дескрипторов . . . . .	9
2.2 Программная реализация . . . . .	12
Заключение . . . . .	14
Список использованных источников . . . . .	15
Приложение А Код программы . . . . .	17

## ВВЕДЕНИЕ

В ряде задач цифровой обработки изображений возникает необходимость однозначного численного описания представленных на изображении точек. В частности, на численном описании отдельных точек и областей изображения основаны решения задач идентификации объектов, совмещения изображений, построения карт глубины и восстановления трехмерной сцены методами фотограмметрии, и многих других.

Математические объекты, описывающие точку изображения, называют дескрипторами. Существует значительное число алгоритмов построения дескрипторов, отличающихся деталями своей реализации, скоростью и точностью работы, а также корректностью результатов для различных исходных данных.

Основными требованиями к дескриптору являются однозначность результата для одинаковых точек изображений, устойчивость к пространственным и яркостным преобразованиям исходных данных, а также вычислительная сложность алгоритмов построения.

Целью работы является разработка дескриптора изображений, устойчивого к пространственным преобразованиям.

В данной работе были рассмотрены некоторые дескрипторы и алгоритмы их построения, методы программной реализации. Разработана реализация построения дескрипторов на основе алгоритма DAISY.

Используемые алгоритмы реализованы на языке программирования общего назначения Python 3.7 с использованием open-source библиотек для обработки изображений OpenCV версии 3.4.2 и scikit-image версии 0.6.12. В качестве тестовых данных использовались наборы изображений из ряда открытых источников.

# 1 Исследование алгоритмов построения дескрипторов изображений

## 1.1 Постановка задачи

Дескриптором называют математический объект, сопоставленный с определенной точкой изображения, и представляющий достаточно однозначное ее описание, позволяющее с высокой степенью уверенности идентифицировать аналогичную точку или область на другом изображении.

Как правило, дескриптор представляет собой вектор значений, вычисляемый определенной функцией для точки изображения.

Стоит заметить, что на практике для однозначного описания точки изображения недостаточно исключительно информации о яркости отдельного пикселя. В связи с этим, в качестве исходных данных для построения дескриптора используется набор из нескольких точек изображения, находящихся в окрестности заданной.

Рассмотрим следующую постановку задачи. Обозначим дескриптор точки  $p$  как  $d_p$ . Пусть дано изображение:

$$I : S \rightarrow \mathbb{R}, \quad S \subset \mathbb{R}^k, \quad (1)$$

где  $k$  – размерность, для плоских изображений равная 2.

Построение дескриптора будет выглядеть следующим образом:

$$d_p = f(I(\hat{p})), \quad \hat{p} \in S. \quad (2)$$

где  $f$  – функция построения дескриптора,

$\hat{p}$  – набор точек изображения  $I$ .

Данная схема сохраняется независимо от конкретного алгоритма построения дескриптора. Конкретный вид и свойства полученного результа-

та будут зависеть как от вида функции  $f$ , так и от конфигурации набора исходных точек  $\hat{p}$ .

Основным требованием к дескриптору является однозначность результата, т.е. удовлетворение значения (2) следующему выражению для произвольного числа изображений  $N$ :

$$f(I_i(\hat{p}_i)) = f(I_j(\hat{p}_j)), \quad i = 1 \dots N, \quad j = 1 \dots N, \quad (3)$$

где  $p_i, p_j$  – пара соответствующих точек изображений  $I_i, I_j$ .

Следует заметить, что построенные дескрипторы, как правило, подвергаются нормализации для уменьшения влияния на точность идентификации точек яркостных характеристик изображения и пространственных преобразований. Конкретные методы нормализации будут рассмотрены в следующих разделах.

Таким образом, используя различные функции построения и варьируя набор исходных точек, можно получать дескрипторы, обладающие различными свойствами, достоинствами и недостатками для конкретных задач.

Основными факторами, принимаемыми в расчет при разработке алгоритма построения дескрипторов, являются:

1. Устойчивость к пространственным преобразованиям - сохранение значения дескриптора при повороте, сдвиге, масштабировании изображения;
2. Устойчивость к яркостным преобразованиями - сохранение значения дескриптора при изменении яркости и контрастности;
3. Вычислительная сложность построения;
4. Информативность - достаточность данных дескриптора для дальнейшей идентификации и сравнения точек.

Существует значительное число реализаций алгоритмов построения дескрипторов. Рассмотрим наиболее популярные подходы.

## **1.2 Существующие алгоритмы построения дескрипторов**

Следует отметить разницу в требованиях к алгоритму построения дескрипторов в зависимости от предметной области применения. В случае использования дескрипторов для описания особых точек (features) необходима повышенная точность идентификации и инвариантность к преобразованиям, вычислительная сложность же не является самым важным критерием, т.к. число особых точек на изображении, как правило, на порядки меньше его размерности.

Напротив, в задачах плотного сопоставления изображений, возникающих в фотограмметрии и построении карт глубины, требуется вычисление дескриптора для каждого пикселя, что накладывает дополнительные ограничения на вычислительную сложность. При этом точностью построения и идентификации возможно в некоторой степени пренебречь, т.к. в данных задачах дескрипторы применяются для оценки общих тенденций преобразований между изображениями, что позволяет за счет их большого количества добиться хороших результатов усреднением и дополнительной валидацией.

SIFT, SURF, ORB. Все сосут, ибо тяжелые, хороши для фичей, плохи для dense. Отъебитесь от меня пожалуйста, спасибо.

## 2 Разработка алгоритма построения дескрипторов изображений

### 2.1 Алгоритм построения дескрипторов

В качестве основы для разрабатываемого метода используется алгоритм DAISY. В основе алгоритма лежит метод построения гистограмм направленных градиентов.

Гистограммой направленных градиентов называют набор направлений градиентов изображения, построенный в определенной области. Данная техника позволяет описать точку распределением интенсивности в ее окрестности.

Для построения градиента используются стандартные фильтры, например, оператор Собеля.

Построенные карты градиентов подвергаются последовательному размытию с постепенным увеличением ядра гауссовского фильтра. Полученные карты называются свернутыми картами направлений (Convolved Orientation Maps, COM).

Вычисление COM для изображения  $I$  производится следующим образом:

$$G_o^\sigma = G^\sigma * \left( \frac{dI}{do} \right)^+, \left( \frac{dI}{do} \right)^+ = \max \left( \frac{dI}{do}, 0 \right), \quad (4)$$

где  $G^\sigma$  – ядро Гауссовского фильтра с стандартным отклонением  $\sigma$ ,  
 $o$  – направление градиента.

Таким образом, получается набор карт градиента, каждая из которых с увеличением уровня размытия представляет собой все более обобщенную информацию о направлениях интенсивности исходного изображения.

Описанные операции выполняются один раз при начале работы алгоритма, полученные  $T$  карт с различными уровнями размытия далее исполь-

зуются для выборки результатов.

Выборка результатов для каждой точки  $(u, v)$  изображения  $I$  производится по следующей схеме:

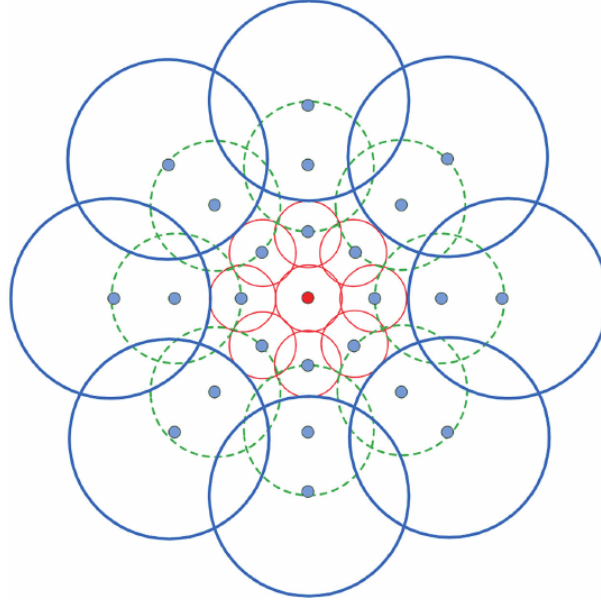


Рисунок 1 – Схема выборки значений из набора COM

Для каждой указанной на схеме точки выбираются значения COM, составляя очередной элемент дескриптора  $h_\sigma(u, v)$ .

Каждый круг на схеме соответствует одной области выборки значений COM в его центральной точке. COM строятся для  $H$  отдельных направлений, получая таким образом  $H$  значений в точке.

Точки сэмпирования расположены  $Q$  концентрическими кольцами вокруг центральной, для которой производится вычисление дескриптора, каждое кольцо задает  $T \cdot H$  значений результата.

Следовательно, общее число элементов дескриптора будет равно:

$$D_s = (Q \cdot T + 1) \cdot H. \quad (5)$$

Вектор значений COM  $h_\sigma(u, v)$  для точки  $(u, v)$  задается следующим образом:

$$h_\sigma(u, v) = [G_1^\sigma(u, v), \dots, G_H^\sigma(u, v)]^T. \quad (6)$$



Полученный вектор значений нормализуется.

Полный дескриптор для точки  $(u_0, v_0)$  получается путем конкатенации всех нормализованных векторов значений СОМ  $h_{\sigma_i}(u, v)$ , начиная с центральной точки:

$$D(u_0, v_0) = [h_{\sigma_1}(u_0, v_0), h_{\sigma_1}(I_1(u_0, v_0, R_1)), \dots, h_{\sigma_1}(I_T(u_0, v_0, R_1), \\ h_{\sigma_2}(I_1(u_0, v_0, R_2)), \dots, h_{\sigma_2}(I_T(u_0, v_0, R_2), \\ \dots \dots \dots h_{\sigma_Q}(I_1(u_0, v_0, R_Q)), \dots, h_{\sigma_Q}(I_T(u_0, v_0, R_Q))],$$

где  $I_j(u, v, R)$  – точка на расстоянии  $R$  от  $(u, v)$  в направлении  $j$  при числе направлений  $T$ .

При необходимости может производиться нормализация вектора значений дескриптора одним из следующих способов. Петух, вернись и дополни, каких именно.

## 2.2 Программная реализация

Алгоритм сшивки изображений для исследования был реализован на языке Python 3.7 с использованием библиотеки OpenCV 3.4.2.

Функция, осуществляющая процедуру совмещения, принимает на вход два изображения и тип используемого дескриптора. Используются дескрипторы SIFT, SURF, ORB, BRIEF. Заметим, что алгоритмы SIFT и SURF являются патентованными, и не присутствуют в стандартной библиотеке OpenCV из-за лицензионных ограничений. Данные алгоритмы доступны в пакете `xfeatures2d` при использовании версии библиотеки, включающей неофициальные алгоритмы, `opencv-contrib`. Использование данных алгоритмов разрешается только в некоммерческих целях [11].

Для вычисления дескрипторов производится преобразование исходных изображений в градации серого.

```
# convert the image to grayscale  
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

Производится поиск наборов особых точек при помощи выбранного дескриптора.

```
# detect keypoints in the image  
detector = cv2.FeatureDetector_create("SIFT")  
kps = detector.detect(gray)  
  
# extract features from the image  
extractor = cv2.DescriptorExtractor_create("SIFT")  
(kps, features) = extractor.compute(gray, kps)
```

Найденные наборы проверяются на совпадение дескрипторов методом knn. Полученные совпадения дополнительно тестируются на предмет нахождения расстояния между точками в заданных пределах. Пары, прошедшие тест, используются для вычисления преобразования.

```
# compute the raw matches and initialize the list of actual  
# matches
```

```

matcher = cv2.DescriptorMatcher_create("BruteForce")
rawMatches = matcher.knnMatch(featuresA, featuresB, 2)
matches = []

# loop over the raw matches
for m in rawMatches:
    # ensure the distance is within a certain ratio of each
    # other (i.e. Lowe's ratio test)
    if len(m) == 2 and m[0].distance < m[1].distance * ratio:
        matches.append((m[0].trainIdx, m[0].queryIdx))

```

После нахождения совпавших точек, по их наборам для каждого изображения вычисляется матрица проективного преобразования. Если совпавших опорных точек найдено меньше, чем четыре, проективное преобразование не может быть вычислено, и функция возвращает статус ошибки.

```

# compute the homography between the two sets of points
(H, status) = cv2.findHomography(ptsA, ptsB, cv2.RANSAC,
reprojThresh)

```

Построенная матрица проективного преобразования применяется ко второму входному изображению, трансформируя его до совпадения положений опорных точек.

Исходное первое изображение и преобразованное второе записываются в одно результирующее. В зависимости от требуемых параметров, сложение может производиться с различными значениями альфа-канала, в том числе с выделением яркостью пересекающейся области либо одного из изображений в целях повышения наглядности.

Функция возвращает результирующее изображение и отчет о работе алгоритма, включающий время выполнения и построенную матрицу проективного преобразования.

Полный код функции представлен в приложении А.

## **ЗАКЛЮЧЕНИЕ**

В ходе данной работы был сделан мем.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Bres, S. Detection of interest points for image indexation [Текст] / S. Bres, J. M. Jolion // International Conference on Advances in Visual Information Systems. – Springer, Berlin, Heidelberg, 1999. – P. 427-435.
- 2 Lowe, D. G. Distinctive image features from scale-invariant keypoints [Текст] / D. G. Lowe // International journal of computer vision. – 2004. – Vol. 60. – I. 2. – P. 91-110.
- 3 Rublee, E. ORB: an efficient alternative to SIFT or SURF [Текст] / E. Rublee, D. Brando, J. Joestar // ICCV '11 Proceedings of the 2011 International Conference on Computer Vision. – IEEE Computer Society Washington, DC, USA, 2011. – P. 2564-2571.
- 4 Проективное преобразование [Электронный ресурс] // Википедия : свободная энцикл. – Электрон. дан. – 2019. – URL: <https://ru.wikipedia.org/?oldid=99107559> (дата обращения: 15.04.2019).
- 5 Аффинное преобразование [Электронный ресурс] // Википедия : свободная энцикл. – Электрон. дан. – 2019. – URL: <https://ru.wikipedia.org/?oldid=93707864> (дата обращения: 15.04.2019).
- 6 Karami, E. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images [Электронный ресурс] / E. Karami, S. Prasad, M. Shehata // arXiv preprint. – 2015. – Электрон. дан. – URL: <https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf> (дата обращения: 25.05.2019).
- 7 Kong, H. A generalized Laplacian of Gaussian filter for blob detection and its applications [Текст] / H. Kong, H. C. Akakin, S. E. Sarma // IEEE

transactions on cybernetics. – IEEE Computer Society Washington, DC, USA, 2013. – V. 43. – I. 6. – P. 1719-1733.

8 Moeslund, T. B. BLOB Analysis: An introduction to video and image processing [Текст] / T. B. Moeslund. – Springer, London, 2012. – 227 p. – p. 5-20.

9 Цифровой шум изображения [Электронный ресурс] // Википедия : свободная энцикл. – Электрон. дан. – 2019. – URL: <https://ru.wikipedia.org/?oldid=95235149> (дата обращения: 15.04.2019).

10 Deng, G. An adaptive Gaussian filter for noise reduction and edge detection [Текст] / G. Deng, L. W. Cahill // IEEE Conference Record. – IEEE Computer Society Washington, DC, USA, 1993. – P. 1615-1619. b

11 OpenCV Documentation [Электронный ресурс] : официальная документация библиотеки OpenCV. / Intel Corporation, Willow Garage Inc., Itseez Ltd. – Электрон. дан. – 2019. – URL: <https://docs.opencv.org> (дата обращения: 25.04.2019).

## **ПРИЛОЖЕНИЕ А**

### **Код программы**