

## РЕФЕРАТ

**Отчет:** 19 страниц, 6 рисунков, 13 источников.

*ОБРАБОТКА ИЗОБРАЖЕНИЙ, ДЕСКРИПТОРЫ ТОЧЕК, DAISY, SIFT, SURF, 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 методы построения дескрипторов . . . . .	9
2 Разработка алгоритма построения дескрипторов изображений . . . . .	12
2.1 Алгоритм построения дескрипторов . . . . .	12
Заключение . . . . .	17
Список использованных источников . . . . .	18

## 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 [1].

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 [6]. 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.

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 [13]. 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.

Существует значительное число реализаций алгоритмов построения дескрипторов. Далее будут рассмотрены некоторые популярные методы и их особенности.

## 1.2 Существующие методы построения дескрипторов

### 1.2.1 SIFT

Большинство существующих дескрипторов основаны на методах сбора информации о градиенте освещенности изображения в некоторой локальной окрестности точки. Распространенным решением для построения дескрипторов является алгоритм SIFT [2].

В методе SIFT дескриптором является вектор. Как и направление ключевой точки, дескриптор вычисляется на гауссиане, ближайшем по масштабу к ключевой точке, и исходя из градиентов в некотором окне ключевой точки. Перед вычислением дескриптора это окно поворачивают на угол направления ключевой точки, чем и достигается инвариантность относительно поворота.

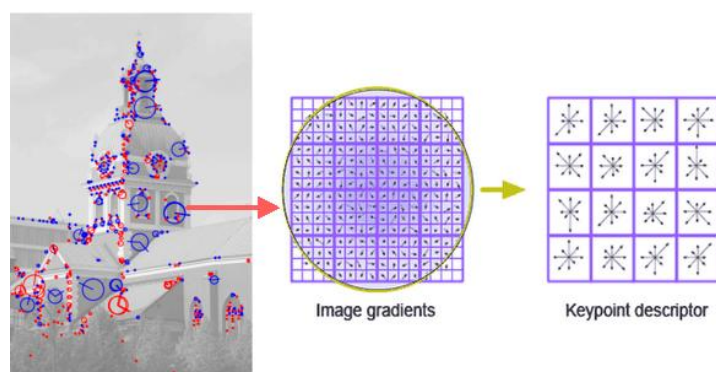


Рисунок 1 – Построение дескриптора SIFT

Дескриптор точки состоит из всех полученных гистограмм в ее локальной окрестности. На практике используются дескрипторы размерности 128 компонент ( $4 \times 4 \times 8$ ).

Полученный дескриптор нормализуется, после чего все его компоненты, значение которых больше 0.2, урезаются до значения 0.2 и затем дескриптор нормализуется ещё раз. В таком виде дескрипторы готовы к использованию.

Существует большое число модификаций SIFT, направленных на улучшение устойчивости к определенным преобразованиям.

### 1.2.2 SURF

Дескриптор SURF (Speeded up Robust Features) относится к числу тех дескрипторов, которые одновременно выполняют поиск особых точек и строят их описание, инвариантное к изменению масштаба и вращения. Кроме того, сам поиск ключевых точек обладает инвариантностью в том смысле, что повернутый объект сцены имеет тот же набор особых точек, что и образец.

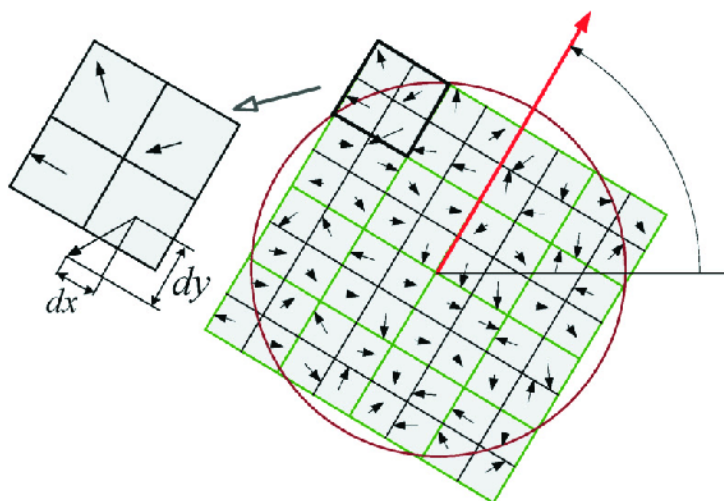


Рисунок 2 – Построение дескриптора SURF

Определение особых точек на изображении выполняется на основании матрицы Гессе. Использование Гессиана обеспечивает инвариантность относительно преобразования типа ”поворот но не инвариантность относительно изменения масштаба. Поэтому SURF применяет фильтры разного масштаба для вычисления Гессиана. Для каждой найденной особой точки вычисляется ориентация – преобладающее направление перепада яркости. Понятие ориентации близко к понятию направления градиента, но для определения ориентации особой точки применяется фильтр Хаара. Дескриптор формируется в результате склеивания взвешенных описаний



градиента для 16 квадрантов вокруг особой точки. Элементы дескриптора взвешиваются на коэффициенты Гауссова ядра. Веса необходимы для большей устойчивости к шумам в удаленных точках.

### 1.2.3 GLOH

Дескриптор GLOH (Gradient location-orientation histogram) является модификацией SIFT-дескриптора, который построен с целью повышения надежности. Вычисляется SIFT дескриптор, но используется полярная сетка разбиения окрестности на бины: 3 радиальных блока с радиусами 6, 11 и 15 пикселей и 8 секторов. В результате получается вектор, содержащий 272 компоненты, который проецируется в пространство размерности 128 посредством использования анализа главных компонент (PCA).

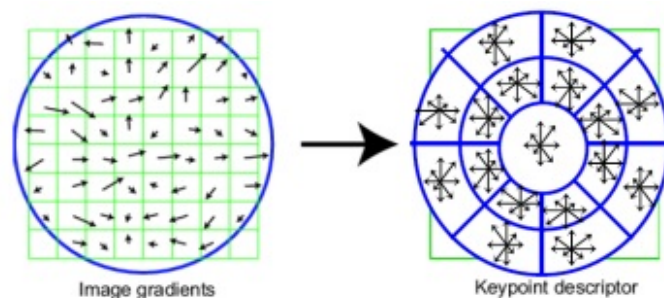


Рисунок 3 – Построение дескриптора GLOH

Общим для всех описанных алгоритмов является основанность на механизмах расчета локального градиента. Данный процесс, несмотря на высокую точность, может быть достаточно вычислительно сложным, что в особенности проявляется в задачах плотного сопоставления, где вычисления производятся для каждой точки изображения [6]. Альтернативное решение используется в методе DAISY [3], выбранном в качестве основы алгоритма в данной работе.

## 2 Разработка алгоритма построения дескрипторов изображений

### 2.1 Алгоритм построения дескрипторов

В качестве основы для разрабатываемого метода используется алгоритм DAISY [3]. В основе алгоритма лежит метод построения гистограмм направленных градиентов.

Гистограммой направленных градиентов называют набор направленных 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.

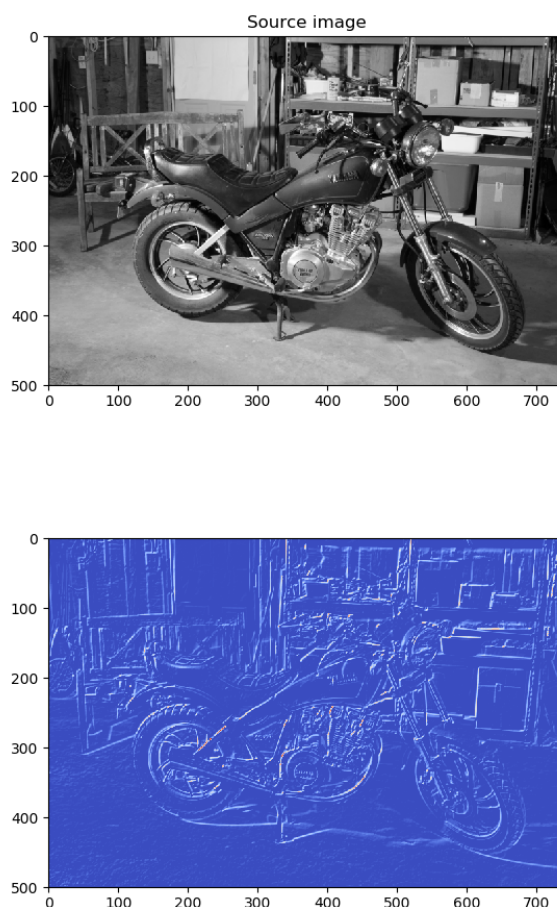


Рисунок 4 – Пример построения градиента изображения

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

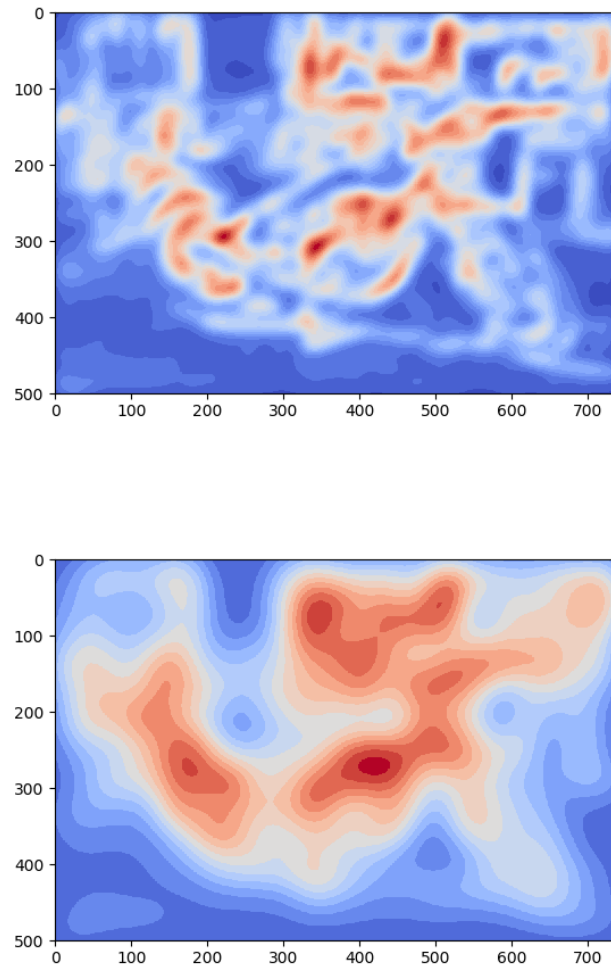


Рисунок 5 – Пример COM с различными степенями размытия

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 используются для выборки результатов.

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

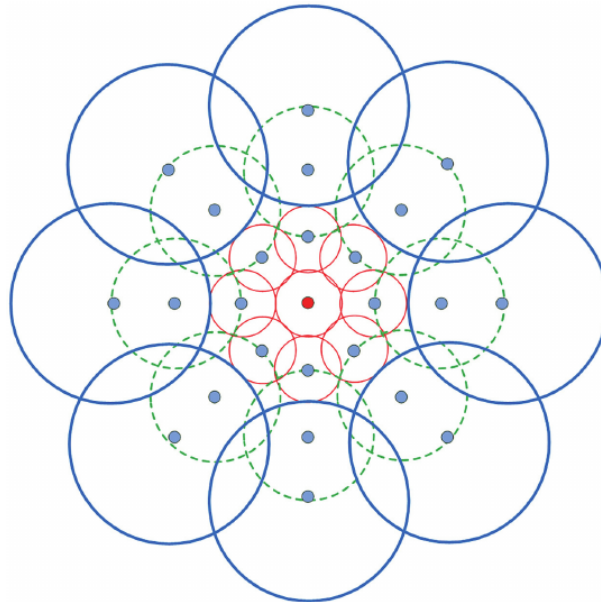


Рисунок 6 – Схема выборки значений из набора SOM

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

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

Точки выборки 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 СОМ  $h_\sigma(u, v)$  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:

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

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:

1. Нормализация отдельных векторов  $h_\sigma(u, v)$ ;
2. Полная нормализация вектора  $D(u_0, v_0)$ ;
3. Метод нормализации, применяемый в SIFT. После полной нормализации  $D(u_0, v_0)$  вектор приводится к единичной сумме.

Как можно видеть, описанный алгоритм отличается достаточно низкой вычислительной сложностью и простотой программной реализации.

Наиболее ресурсоемким этапом является построение диаграммы направленных градиентов и последовательное размытие SOM [13].

Данные операции хорошо поддаются распараллеливанию или оптимизации путем замены свертки с крупным ядром последовательными свертками с ядрами меньшего размера, и выполняются один раз для всего изображения. Конфигурация паттерна выборки и методы нормализации влияют на производительность в незначительной степени, и могут варьироваться в широких пределах без серьезного повышения вычислительной сложности.

## **ЗАКЛЮЧЕНИЕ**

В ходе данной работы была собрана и проанализирована информация о существующих методах построения дескрипторов изображений. На основе существующего алгоритма DAISY был разработан алгоритм, потенциально обладающий высокой степенью устойчивости к пространственным преобразованиям исходного изображения. Разработанный алгоритм обладает набором параметров, позволяющих достаточно свободно варьировать точность и вычислительную сложность построения без изменения исходного кода, что позволяет эффективно адаптировать алгоритм к конкретным задачам. Построенные дескрипторы можно использовать как в задачах распознавания, так и в задачах плотного сопоставления изображений.

## 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 Tola, E. DAISY: A Fast Local Descriptor for Dense Matching [Текст] / E. Tola, V. Lepetit, P. Fua // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010. Vol. 32, num. 5 – P. 815-830.
- 4 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.
- 5 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.
- 6 Rekhil, M. A Survey on Image Feature Descriptors [Текст] / M. Rekhil, K. Sreekumar // (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, – P. 7668-7673.
- 7 Funayama, R. Robust interest points detector and descriptor [Текст] / R. Funayama, H. Yanagihara // Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3 – P. 346–359.



8 Panchal, P. A Comparison of SIFT and SURF [Текст] / R. Funayama, H. Yanagihara // International Journal of Innovative Research in Computer and Communication Engineering Vol. 1, Issue 2, April 2013 – P. 287–302.

9 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: 20.11.2019).

10 Scikit-image Documentation [Электронный ресурс] : официальная документация библиотеки scikit-image. / The scikit-image team. – Электрон. дан. – 2019. – URL: <https://scikit-image.org/docs/stable/> (дата обращения: 25.11.2019).

11 Karami, E. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images [Электронный ресурс] / E. Karami, S. Prasad, M. Shehata // Journal of Visual Communication and Image Representation – 2015. – Электрон. дан. – URL: <https://www.sciencedirect.com/science/article/pii/S1047320315002230> (дата обращения: 03.12.2019).

12 Xue, B. A DAISY descriptor based multi-view stereo method for large-scale scenes [Электронный ресурс] / B. Xue, L. Cao // Advances in Intelligent Systems and Computing – 2015. – Электрон. дан. – URL: <https://doi.org/10.1016/j.jvcir.2015.11.007> (дата обращения: 06.12.2019).

13 Tola, E. DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo [Электронный ресурс] / E. Tola // IEEE Xplore Digital Library – 2014. – Электрон. дан. – URL: <https://ieeexplore.ieee.org/document/4815264> (дата обращения: 05.12.2019).