

## ОБЗОР И ПРИМЕНЕНИЕ БИБЛИОТЕКИ OPENCV 2.4.9 ДЛЯ ПОИСКА И СОПОСТАВЛЕНИЯ КЛЮЧЕВЫХ ТОЧЕК ИЗОБРАЖЕНИЙ

Баглаева Е.А.

Научный руководитель: Цапко С.Г.  
Томский политехнический университет  
634050, Россия, г. Томск, пр–т Ленина, 30  
E-mail: eab14@tpu.ru

### Введение

**OpenCV** (англ. *Open Source Computer Vision Library*, библиотека компьютерного зрения с открытым исходным кодом) — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Разработана исследовательским центром *Intel Russia* в Нижнем Новгороде, на сегодняшний день находится при поддержке исследовательской лаборатории *Willow Garage* и компании *Itseez*. [1]

*OpenCV* может свободно использоваться в академических и коммерческих целях, т.к. распространяется в условиях лицензии *BSD (Berkeley Software Distribution license)* — Программная лицензия университета Беркли. [2] Библиотека содержит интерфейсы языков программирования *C++*, *C*, *Python* и *Java*, поддерживается операционными системами *Windows*, *Linux*, *Mac OS*, *iOS* и *Android*. Разработана для обеспечения эффективности вычислений, с сильным акцентом на обработке изображений в режиме реального времени. Написанная на оптимизированном *C/C++*, библиотека *OpenCV* может использовать преимущества многоядерных процессоров. Задействованная вместе с *OpenCL*, библиотека может использовать аппаратное ускорение базовой гетерогенной вычислительной платформы.

Сообщество пользователей *OpenCV* составляет более 47 тысяч участников, и предполагаемое число загрузок превышает 9 миллионов. Библиотека находит свое применение в различных сферах от интерактивного искусства до инспекции шахт, геоинформатики и перспективных направлений робототехники.

На сегодняшний день последнее стабильное обновление библиотеки носит название *OpenCV 2.4.9* и было выпущено в апреле 2014.

### Структура OpenCV

*OpenCV* имеет модульную структуру, что означает, что пакет включает в себя несколько общих (*shared*) и статических (*static*) библиотек. Доступны следующие основные модули:

*Core* – модуль, определяющий основные структуры данных, в том числе многомерный массив *Mat* и основные функции, используемые всеми остальными модулями.

*Imgproc* – модуль обработки изображений, который включает методы линейной и нелинейной

фильтрации, геометрические преобразования изображений (масштабирование, аффинные и перспективные преобразования, текстурирование), преобразования цветового пространства, гистограмм и т.д.

*Video* – модуль анализа видео, включает методы оценки параметров движения, вычитания заднего плана и алгоритмы отслеживания объекта.

*Calib3d* – модуль, содержащий базовые геометрические алгоритмы стереоскопической визуализации, методы калибровки обычной и стереоскопической камеры, оценки расположения объектов, алгоритмы поиска соответствий для стерео изображений, элементы *3D* реконструкции.

*Features2d* – модуль, предназначенный для поиска ключевых точек изображений, определения и сопоставления дескрипторов.

*Objdetect* – модуль, отвечающий за распознавание определенных объектов и экземпляров предопределенных классов (например, лиц, глаз, людей, автомобилей и т.д.).

*Highgui* – модуль, содержащий видео кодеки, простой в использовании интерфейс для захвата видео, методы обработки изображений, а также простые возможности пользовательского интерфейса.

*Gpu* – модуль, хранящий алгоритмы ускорения на *GPU* из других модулей *OpenCV*.

Также присутствуют вспомогательные модули, такие как *FLANN*, тестовые оболочки *Google*, *Python* привязки и др. [3]

### Обработка изображений в OpenCV

Допустим, имеются два изображения: объекта и сцены со многими объектами. Поставим задачу найти и выделить объект, который есть на обоих изображениях, соединив ключевые точки линиями.

Процесс решения данной задачи можно разбить на два шага: на первом шаге происходит выявление ключевых точек, на втором – их сопоставление.

В качестве понятия **ключевых точек изображения** в компьютерном зрении подразумеваются «особые» части изображений, которые можно принять в качестве исходных данных для решения определенных задач сравнения и распознавания изображений. Ключевыми точками могут выступать края, углы, пятна, выступы объектов изображения.

**Дескриптором** изображения в компьютерном зрении называют описание элементарных характеристик изображения в окрестности ключевой точки, таких как цвет, текстура, форма, движение, расположение и др.

Дескрипторы можно считать первым шагом, позволяющим выяснить связь между пикселями в цифровом изображении и тем, что человек может вспомнить через несколько минут после просмотра одного или нескольких изображений.

В *OpenCV* используется общий интерфейс *DescriptorExtractor*, позволяющий переключаться между разными алгоритмами извлечения дескрипторов. В данной работе применяется *SurfDescriptorExtractor*, использующий *SURF* (англ. *Speeded Up Robust Features*, ускорение надежных средств) алгоритм, запатентованный в 2006 году Г.Бэем [4].

На втором шаге происходит сравнение дескрипторов, определяющих ключевые точки изображений, с целью поиска совпадений. В данной работе для выполнения этого шага применяются три способа:

1) с использованием *BFMatcher*, который перебирает все дескрипторы первого изображения и сравнивает их с дескрипторами второго изображения, т.е. применяет алгоритм полного перебора (Рис. 1). [5]



Рис. 1. Результат применения *BFMatcher*

2) с использованием *FlannBasedMatcher*, который при сравнении дескрипторов изображений применяет алгоритм поиска ближайшего соседа из библиотеки FLANN (Fast Library for Approximate Nearest Neighbors, быстрая библиотека приближенных ближайших соседей), распространяемой под лицензией *BSD* (Рис. 2). [6]



Рис. 2. Результат применения *FlannBasedMatcher*

3) с использованием *FlannBasedMatcher* и функций *findHomography* для нахождения и *perspectiveTransform* для выполнения перспективного преобразования ключевых точек изображений – способ позволяет реагировать и наглядно отображать поворот (или его отсутствие) одного изображения относительно другого (Рис. 3). [7]



Рис. 3. Результат применения *FlannBasedMatcher* и перспективных преобразований

### Выводы

Из полученных результатов следует, что алгоритм полного перебора является самым ненадежным способом решения задачи поиска и сопоставления ключевых точек изображений. Данный метод находит большое число совпадений, более половины из которых – ложные.

Метод поиска ближайшего соседа обладает высокой точностью, повышая или понижая которую можно регулировать число найденных совпадений. В *OpenCV* для него реализованы несколько алгоритмов в модуле FLANN, из которых автоматически выбирается нужный.

### Список источников

1. OpenCV. – Wikipedia [Website]. – URL: <https://en.wikipedia.org/wiki/OpenCV>
2. BSD licenses. – Wikipedia [Website]. – URL: [https://en.wikipedia.org/wiki/BSD\\_licenses](https://en.wikipedia.org/wiki/BSD_licenses)
3. OpenCV. – Opencv.org [Website]. – URL: <http://opencv.org/>
4. Bay, H. and Tuytelaars, T. and Van Gool, L. “SURF: Speeded Up Robust Features”, 9th European Conference on Computer Vision, 2006
5. Feature Description. – Docs.opencv.org [Website] – URL: [http://docs.opencv.org/doc/tutorials/features2d/feature\\_description/feature\\_description.html#feature-description](http://docs.opencv.org/doc/tutorials/features2d/feature_description/feature_description.html#feature-description)
6. Feature Matching with FLANN. – Docs.opencv.org [Website] – URL: [http://docs.opencv.org/doc/tutorials/features2d/feature\\_flann\\_matcher/feature\\_flann\\_matcher.html#feature-flann-matcher](http://docs.opencv.org/doc/tutorials/features2d/feature_flann_matcher/feature_flann_matcher.html#feature-flann-matcher)
7. Features2D + Homography to find a known object. – Docs.opencv.org [Website] – URL: [http://docs.opencv.org/doc/tutorials/features2d/feature\\_homography/feature\\_homography.html#feature-homography](http://docs.opencv.org/doc/tutorials/features2d/feature_homography/feature_homography.html#feature-homography)