

**ФАНО РОССИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ**

**«ФЕДЕРАЛЬНЫЙ НАУЧНЫЙ ЦЕНТР НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ СИСТЕМНЫХ ИССЛЕДОВАНИЙ РОССИЙСКОЙ АКАДЕМИИ НАУК»**

**(ФГУ ФНЦ НИИСИ РАН)**

**ОТДЕЛ АСПИРАНТУРЫ**

**РЕФЕРАТ на тему:**

**«Основные алгоритмы библиотеки компьютерного зрения OpenCV и способы их применения»**

|  |  |
| --- | --- |
|  | Выполнил(а):  Аспирант 3 года обучения  Бесшапошников Никита Олегович  Ф.И.О. (полностью)  Направление подготовки: 09.06.01  Направленность (профиль): 05.13.11 |
|  | Проверил:  к.ф.-м.н., доцент, заведующий сектором информатики отдела учебной информатики Леонов А.Г. |

**Москва – 2018**

1. **Введение**

OpenCV [1] (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. Она имеет реализацию на различных языках программирования, таких как C/C++, Python, Java, Ruby, Matlab, Lua и других. Распространяется в условиях лицензии BSD.

Библиотека OpenCV была создана корпорацией Intel (США) в 1999 году как часть другого большого проекта по стерео дисплеям. С 2006 года по настоящее время разработано 3 поколения библиотеки. В современном варианте OpenCV включает в себя более 500 функций компьютерного зрения и обработки изображений, а также некоторые полезные алгоритмы общего назначения. Разработчики библиотеки утверждают, что созданный ими система обладает высокой мобильностью и эффективностью.

На данный момент библиотека состоит из набора компактных модулей с узкой специализацией:

* opencv\_core — основная функциональность. Включает в себя базовые структуры, вычисления (математические функции, генераторы случайных чисел) и линейную алгебру, DFT, DCT, ввод/вывод для XML и YAML и т. д.
* opencv\_imgproc — обработка изображений (фильтрация, геометрические преобразования, преобразование цветовых пространств и т. д.).
* opencv\_highgui — простой UI, ввод/вывод изображений и видео.
* opencv\_ml — модели машинного обучения (SVM, деревья решений, обучение со стимулированием и т. д.).
* opencv\_features2d — распознавание и описание плоских примитивов (SURF, FAST и другие, включая специализированный фреймворк).
* opencv\_video — анализ движения и отслеживание объектов (оптический поток, шаблоны движения, устранение фона).
* opencv\_objdetect — обнаружение объектов на изображении (нахождение лиц с помощью алгоритма Виолы-Джонса, распознавание людей HOG и т. д.).
* opencv\_calib3d — калибровка камеры, поиск стерео-соответствия и элементы обработки трёхмерных данных.
* opencv\_flann — библиотека быстрого поиска ближайших соседей (FLANN 1.5) и обертки OpenCV.
* opencv\_gpu — ускорение некоторых функций OpenCV за счет CUDA, создан при поддержке NVidia.

1. **Основные алгоритмы**
   1. **Детектор границ Кэнни**

Оператор Кэнни (детектор границ Кэнни, алгоритм Кэнни) — оператор обнаружения границ изображения. Был разработан в 1986 году Джоном Кэнни [2] и использует многоступенчатый алгоритм для обнаружения широкого спектра границ в изображениях. Целью Кэнни было разработать оптимальный алгоритм обнаружения границ, удовлетворяющий трём критериям:

* хорошее обнаружение (Кэнни трактовал это свойство как повышение отношения сигнал/шум);
* хорошая локализация (правильное определение положения границы);
* единственный отклик на одну границу.

Из этих критериев затем строилась целевая функция стоимости ошибок, минимизацией которой находится «оптимальный» линейный оператор для свёртки с изображением.

Основные этапы алгоритма:

1. Сглаживание. Размытие изображения для удаления шума
2. Поиск градиентов. Границы отмечаются там, где градиент изображения приобретает максимальное значение. Они могут иметь различное направление, поэтому алгоритм Кэнни использует четыре фильтра для обнаружения горизонтальных, вертикальных и диагональных ребер в размытом изображении.
3. Подавление «немаксимумов». Только локальные максимумы отмечаются как границы.
4. Двойная пороговая фильтрация. Потенциальные границы определяются порогами.
5. Трассировка области неоднозначности. Итоговые границы определяются путём подавления всех краёв, не связанных с определенными (сильными) границами.
   1. **Оператор Собеля**

Оператор Собеля — дискретный дифференциальный оператор, вычисляющий приближённое значение градиента яркости изображения [3]. Результатом применения оператора Собеля в каждой точке изображения является либо вектор градиента яркости в этой точке, либо его норма.

Оператор Собеля основан на свёртке изображения небольшими сепарабельными целочисленными фильтрами в вертикальном и горизонтальном направлениях. Он вычисляет градиент яркости изображения в каждой точке. Так находится направление наибольшего увеличения яркости и величина её изменения в этом направлении. Результат показывает, насколько «резко» или «плавно» меняется яркость изображения в каждой точке, а значит, вероятность нахождения точки на грани, а также ориентацию границы. Результатом оператора Собеля в точке, лежащей в области постоянной яркости, будет нулевой вектор, а в точке, лежащей на границе областей различной яркости, — вектор, пересекающий границу в направлении увеличения яркости.

* 1. **Преобразование Хафа**

Преобразование Хафа — численный метод, применяемый для поиска объектов, принадлежащих определённому классу фигур, с использованием процедуры голосования [3]. Процедура голосования применяется к пространству параметров, из которого и получаются объекты определённого класса фигур по локальному максимуму в так называемом накопительном пространстве, которое строится при вычислении трансформации Хафа. Классический алгоритм преобразования Хафа связан с идентификацией прямых в изображении, но позже алгоритм был расширен возможностью идентификации позиции произвольной фигуры.

В простейшем случае преобразование Хафа является линейным преобразованием для обнаружения прямых. Прямая может быть задана уравнением y = mx + b и может быть вычислена по любой паре точек (x, y) на изображении. Главная идея преобразования Хафа — учесть характеристики прямой не как уравнение, построенное по паре точек изображения, а в терминах её параметров, то есть m — коэффициента наклона и b — точки пересечения с осью ординат. Исходя из этого прямая, заданная уравнением y = mx + b, может быть представлена в виде точки с координатами (b, m) в пространстве параметров.

Алгоритм преобразования Хафа использует массив, называемый аккумулятором, для определения присутствия прямой y = mx + b. Размерность аккумулятора равна количеству неизвестных параметров пространства Хафа. Два измерения аккумулятора соответствуют квантованным значениям параметров m и b. Для каждой точки и её соседей алгоритм определяет, достаточен ли вес границы в этой точке. Если да, то алгоритм вычисляет параметры прямой и увеличивает значение в ячейке аккумулятора, соответствующей данным параметрам.

Потом, найдя ячейки аккумулятора с максимальными значениями, могут быть определены наиболее подходящие прямые. Самый простой способ — это пороговая фильтрация. Так как полученные прямые не содержат информацию о длине, следующим шагом является нахождение частей изображения, соответствующих найденным прямым.

* 1. **Метод Виолы-Джонса**

Метод Виолы—Джонса — алгоритм, позволяющий обнаруживать объекты на изображениях в реальном времени. Его предложили Паул Виола и Майкл Джонс в 2001 году [4]. Хотя алгоритм может распознавать различные классы изображений, основной задачей при его создании было обнаружение лиц. Данный алгоритм является адаптацией идеи вейвлетов Хаара и ее результатом являются так называемые признаки Хаара.

Признак Хаара состоит из смежных прямоугольных областей. Они позиционируются на изображении, далее суммируются интенсивности пикселей в областях, после чего вычисляется разность между суммами. Эта разность и будет значением определенного признака, определенного размера, определенным образом спозиционированного на изображении.

В основном используются прямоугольные, которые определяется как разность сумм пикселей двух смежный областей внутри прямоугольника, который может занимать различные положения и масштабы на изображении.

Так как для описания объекта с достаточной точностью необходимо несколько признаков Хаара, в методе Виолы — Джонса признаки Хаара организованы в каскадный классификатор.

Алгоритм состоит из четырех основных этапов:

1. Выбор признаков Хаара
2. Создание интегрального изображения. Интегральное изображени – это структура данных и алгоритм для быстрого и эффективного генерирования суммы значений в прямоугольном подмножестве сетки. В данном случае суммируются признаки Хаара.
3. Обучение с помощью AdaBoost. Оно используется как для выбора лучших функций, так и для обучения классификаторов, которые их используют. Этот алгоритм строит «сильный» классификатор как линейную комбинацию взвешенных простых «слабых» классификаторов.
4. Каскадная классификация. Каждый этап каскадирования состоит из сильного классификатора. Таким образом, все функции сгруппированы в несколько этапов, где каждый этап имеет определенное количество функций. Задача каждого этапа состоит в том, чтобы определить, является ли данное под-окно определенно не лицом или может быть лицом. Данное под-окно сразу же отбрасывается как не лицо, если оно не работает на любом из этапов.
   1. **Алгоритмы обнаружения углов**

Обнаружение углов - это подход, используемый для извлечения определенных функций и вывода содержимого изображения. Угловое обнаружение часто используется при обнаружении движения, регистрации изображений, видеонаблюдении, мозаике изображений, панорамной строчке, 3D-моделировании и распознавании объектов. Угловое обнаружение перекрывается с темой обнаружения точки интереса.

Угол может быть определен как пересечение двух ребер. Угол также может быть определен как точка, для которой есть два доминирующих и разных направления граней в локальной окрестности точки.

Точка интереса - это точка изображения, которая имеет четко определенную позицию и может быть надежно обнаружена. Это означает, что точкой интереса может быть угол, но также может быть, например, изолированная точка максимальной или минимальной локальной интенсивности, окончания линии или точка на кривой, где кривизна локально максимальна.

На практике большинство так называемых методов определения угла обнаруживают точки интереса в целом. Eсли нужно обнаружить только углы, необходимо провести локальный анализ обнаруженных точек интереса, чтобы определить, какие из них являются реальными углами.

Угловые детекторы обычно не очень надежны и часто требуют допущений, чтобы предотвратить влияние отдельных ошибок на доминирование над задачей распознавания.

Одним из определения качества углового детектора является его способность обнаруживать один и тот же угол в нескольких похожих изображениях в условиях различного освещения, трансляции, вращения и других преобразований.

* + 1. **Алгоритм SUSAN**

SUSAN - это аббревиатура, обозначающая наименьшее одновалентное сегментное ассимилирующее ядро (smallest univalue segment assimilating nucleus).

Для обнаружения функции SUSAN помещает круговую маску поверх тестируемого пикселя (ядро). Для обнаружения угла используются два дополнительных шага. Во-первых, найден центроид SUSAN. Правильный угол будет иметь центроид вдали от ядра. Второй шаг настаивает на том, что все точки на линии от ядра через центроид к краю маски находятся в SUSAN.

* + 1. **Алгоритм AST**

AST - это аббревиатура для теста ускоренного сегмента (accelerated segment test). Этот тест является расслабленной версией критерия угла SUSAN. Вместо оценки кругового диска рассматриваются только пиксели в круге Брешнама радиуса r вокруг точки кандидата. Если n смежных пикселей все ярче, чем ядро, по крайней мере, t или все темнее ядра t, то пиксель под ядром считается признаком. Этот тест обеспечивает очень стабильные свойства. Выбор порядка, в котором тестируются пиксели, является так называемой проблемой «Двадцать вопросов». Создание деревьев с короткими решениями для этой проблемы приводит к появлению наиболее эффективных вычислительных функций.

Первый алгоритм обнаружения угла, основанный на AST, является FAST (функции из теста ускоренного сегмента). Хотя r в принципе может принимать любое значение, FAST использует только значение 3 (соответствующее окружности с окружностью 16 пикселей), а тесты показывают, что наилучшие результаты достигаются с n, равным 9. Это значение n является самым низким, при котором ребра не обнаруживаются. Порядок, в котором пиксели тестируются, определяется алгоритмом ID3 из обучающего набора изображений.

* 1. **Алгоритмы обнаружения двоичный больших объектов**

Методы обнаружения двоичный больших объектов (blob detection) направлены на обнаружение областей в цифровом изображении, которые отличаются по свойствам, таким как яркость или цвет, по сравнению с окружающими регионами. Неформально blob - это область изображения, в которой некоторые свойства являются постоянными или приблизительно постоянными; все точки в блобе можно в каком-то смысле считать похожими друг на друга. Наиболее распространенным методом обнаружения blob является свертка.

Учитывая некоторое свойство, выраженное как функция положения на изображении, существуют два основных класса детекторов blob:

* 1. дифференциальные методы, основанные на производных функции по отношению к положению
  2. методы, основанные на нахождении локальных максимумов и минимумов функции.

Существует несколько мотивов для изучения и разработки детекторов blob. Одна из основных причин заключается в предоставлении дополнительной информации о регионах, которые не получены из краевых детекторов или угловых детекторов. На ранних работах в этом районе обнаружение blob использовалось для получения областей, представляющих интерес для дальнейшей обработки. Эти регионы могут сигнализировать о наличии объектов или частей объектов в домене изображения с приложением для распознавания объектов и / или отслеживания объектов. В других областях, таких как анализ гистограммы, дескрипторы blob также могут использоваться для обнаружения пиков с применением к сегментации. Еще одно распространенное использование дескрипторов blob - это основные примитивы для анализа текстуры и распознавания текстуры. Дескрипторы blob стали все более популярными в качестве точек интереса для широкого базового стерео сопоставления и для сигнализации присутствия информационных функций изображения для распознавания объектов на основе основанных на статистике локального изображения. Существует также связанное понятие обнаружения гребня, чтобы сигнализировать о наличии удлиненных объектов.

* + 1. **MSER**

Максимально устойчивые экстремальные области (MSER) используются как метод детектирования blob в изображениях. Этот метод был предложен Матасом и др [5]. найти соответствия между элементами изображения с двух изображений с разными точками зрения. Этот метод извлечения исчерпывающего числа соответствующих элементов изображения способствует согласованию с базой, и это привело к лучшему алгоритму сравнения объемных объектов и распознавания объектов. Поскольку регионы определяются исключительно функцией интенсивности в регионе и внешней границе, это приводит ко многим ключевым характеристикам областей, которые делают их полезными. В большом диапазоне порогов локальная бинаризация стабильна в определенных регионах и имеет перечисленные ниже свойства:

1. Инвариантность аффинного преобразования интенсивностей изображений
2. Ковариация для сохранения смежности (непрерывного) преобразования
3. Стабильность: выбираются только регионы, поддержка которых почти одинакова в диапазоне пороговых значений.
4. Обнаружено многомасштабное обнаружение без какого-либо сглаживания, обнаружена как тонкая, так и большая структура.
5. Множество всех экстремальных областей можно перечислить в худшем случае за O (n), где n - количество пикселей в изображении.
   * 1. **PCBR**

Детектор на основе центральной кривизны, также называемый PCBR (principal curvature-based region detector), является детектором признаков, использующий криволинейные структуры для обнаружения особенностей.

Определение криволинейных структур генерирует одинаковый результат для обеих линий и краев, создавая более четкий структурный эскиз изображения, чем обычно обеспечивается градиентным изображением величины. Алгоритм Штегера [6] модифицирован для получения криволинейных изображений. Поскольку используется только первый шаг этого алгоритма, который заключается в вычислении изображений основной кривизны, главная кривизна принимается как имя этого детектора.

Чтобы сделать инвариантность масштаба детектора и улучшить надежность обнаружения, процесс детектирования SIFT Дэвида Лоу [7] симулируется для определения основной криволинейной структуры в пространстве шкалы. Для определения областей используются локальные максимальные изображения значений главной кривизны.

Изображения главной кривизны очищаются путем морфологического закрытия и порогового гистерезисного порогового значения. Затем традиционный алгоритм водораздела применяется на изображениях для приобретения регионов.

Подобно процессу выбора стабильных областей через пороговое значение в MSER, устойчивые области выбираются по локальным изменениям масштаба. Чтобы достичь этого, ошибка перекрытия вычисляется по каждому триплету последовательных масштабов. Если ошибка перекрытия области превышает 90%, сохраняется только одна область. Если ошибка превышает 70% и менее 90%, все регионы сохранены. Если перекрытие составляет менее 70%, отбросьте эти регионы. Эти числа определяются анализом чувствительности дескриптора SIFT.

# Список литературы

|  |  |
| --- | --- |
| [1] | «OpenCV,» [В Интернете]. Доступно: http://opencv.org. |
| [2] | J. CANNY, «A Computational Approach to Edge Detection,» *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* 1986. |
| [3] | Дуда и Харт, Распознавание образов и анализ сцен, 1976. |
| [4] | P. Viola и M. Jones, «Detection, Robust Real-time Objec,» *International Journal of Computer Vision,* 2001. |
| [5] | Pajdla, Matas, Chum и Urban, «Proc. of British Machine Vision Conference,» в *Robust wide baseline stereo from maximally stable extremal regions*, 202. |
| [6] | Steger, «An unbiased detector of curvilinear structures,» *PAMI,* 1998. |
| [7] | D. G. Lowe, « Distinctive image features from scale-invariant keypoints,» *IJCV,* 2004. |