

Распознавание дорожных знаков на изображениях методом Speeded Up Robust Features (SURF)

Е.А. Краснобаев

Учреждение образования «Витебский государственный университет имени П.М. Машерова»

В статье рассматривается задача автоматического обнаружения и распознавания дорожных знаков на изображениях. Для ее решения применялась технология Speeded Up Robust Features (SURF), позволяющая находить на изображениях сцены и знака особенные точки, рассчитывать их дескрипторы, инвариантные к масштабу и вращению и выполнять их сопоставление. В результате работы найден критерий отбора корректных соответствий пар точечных особенностей, повышающий качество распознавания дорожного знака. Разработано программное обеспечение, реализующее алгоритм распознавания с использованием библиотеки OpenCV 2.4. Точность распознавания составила 85% при ограничениях на соотношение разрешения сцены, эталонного знака и знака в сцене 5:2:1, максимальный угол наблюдения – 15°, расстояние до знака – 50 м.

Ключевые слова: машинное зрение, цифровая обработка изображений и распознавание образов.

Recognition of Traffic Signs in Images Using Speeded Up Robust Features (SURF)

Krasnobaev E.A.

Educational establishment «Vitebsk State University named after P.M. Masherov»

We consider the problem of automatic detection and recognition of traffic signs in the image. The chosen technology is «Speeded Up Robust Features (SURF)». This technology is used to find singular points in the scenes and traffic sign, to calculate their descriptors that are invariant to scale and rotation, and perform their comparison. We found the selection criteria of correctly corresponding of features point, which enhance the quality of recognition of the traffic sign. Software has been developed, which implements recognition algorithm using library OpenCV 2.4. Recognition accuracy was 85% at a ratio of resolution of scene, calibrating traffic sign and traffic sign in scene as 5:2:1, maximum viewing angle – 15°, the distance from the traffic sign – 50 m.

Key words: car sight, digital processing of images and detection of images.

Задача автоматической локализации и распознавания дорожных знаков на изображениях актуальна, так как до сих пор не имеет универсального решения, позволяющего получать 100%-ную точность распознавания при любых условиях наблюдения.

Важность решения этой проблемы обусловлена широким спектром ее практического применения: навигационные карты, системы помощи водителю, системы управления инфраструктурой и др. Система распознавания дорожных знаков смогла бы информировать водителей об опасных участках на дороге, о соблюдении скоростного режима, а в перспективе отдавала бы команды автомобилю по выполнению предписаний дорожных знаков. Это позволило бы снизить аварийность на дорогах, помочь водителям быть более внимательными.

Кажущаяся простота задачи является обманчивой, несмотря на то, что дорожные знаки имеют простую форму, цвет и предназначены

быть максимально заметными. Разнообразие типов дорожных знаков, форм, наличие на них текста, от которого зависит интерпретация знака, оставляет данную задачу нерешенной.

Известны системы распознавания, такие, как Siemens VDO TSR, FOSTS, Opel Eye TSR, а также системы распознавания автомобилей BMW, Ford. Однако, даже они имеют точность распознавания до 95%, с учетом ограничения на дальность, угол обзора, степень зашумленности знака, время распознавания.

Решением данной проблемы занимается множество авторов, результаты работ которых изложены, например, в [1–2]. Применяемые авторами методы распознавания можно условно разделить на две группы:

- методы распознавания без обучения;
- методы распознавания с обучением.

К первой группе следует отнести методы, опирающиеся на известные признаки дорожных

знаков, такие, как форма и цвет. Это методы контурные, корреляционные, а также основанные на линейной и нелинейной фильтрации изображений.

Ко второй группе относятся методы, использующие машинное обучение: нейронные сети, обучаемые классификаторы.

Обобщая подходы к решению задачи, можно сделать вывод, что и те и другие группы методов используют следующие этапы распознавания дорожных знаков:

1. Сегментация знака на изображении по заданным признакам.

2. Локализация знака (определение координат описанного прямоугольника).

3. Качественное улучшение изображения знака, коррекция яркости, контраста.

4. Вращение полученного знака в пространстве, с целью проецирования получаемых объектов.

5. Масштабирование знака, в соответствии со знаками в базе данных.

6. Определение соответствий между полученным знаком и знаками в базе данных по заданным признакам.

7. Распознавание текста на знаке (при необходимости).

Как видим, помимо самого распознавания, необходимо решать множество задач по предварительной подготовке изображения к распознаванию, которые сами по себе являются трудоемкими.

На наш взгляд, для комплексного решения данных задач удобен метод SURF (Speeded Up Robust Features), изложенный в [3], а также реализованный в библиотеке компьютерного зрения OpenCV. Таким образом, целью исследования является разработка алгоритма и программного обеспечения для распознавания изображений дорожных знаков методом SURF (Speeded Up Robust Features).

Материал и методы. Объектом исследования, изложенного в данной статье, являются цифровые изображения сцен, содержащие дорожные знаки. Предметом исследования выступают модели, алгоритмы и методы анализа, обработки и распознавания дорожных знаков на изображениях.

Результаты и их обсуждение. Метод SURF (Speeded Up Robust Features) основывается на понятии «особенности изображения». Под особенностью изображения понимают некоторые изолированные точки (точечные особенности), линии (линейные особенности), характеризуемые своими локальными окрестностями. Суще-

ствует несколько подходов к определению понятия особенности, в частном случае, в соответствии с [4] их можно охарактеризовать как разрывы яркости или однородности в изображениях.

В целом, особенность определяется как область изображения, обладающая более существенными признаками, чем яркость, и, исходя из этих признаков, ей может быть найдено однозначное соответствие в ряде изображений. Если некоторая точка или область изображения определена как особенная, она может являться отправной точкой для последующих алгоритмов и методов обработки изображений.

Соответствием между точечными особенностями называется согласованная пара пикселей x_1 и x_2 из изображений I_1 и I_2 , которые отражают одну и ту же точечную особенность x' . При определении соответствий между особенностями могут возникать ложные соответствия, в этом случае вводится понятие корректного соответствия. Оно называется корректным, если существует точка трехмерной сцены X , проекциями которой являются точки x_1 и x_2 из изображений I_1 и I_2 . Вопросы сопоставления точечных особенностей описаны в [4].

В соответствии с [4], точечной особенностью x' называется такая точка изображения, чья окрестность отличается от окрестностей близлежащих точек по выбранной мере, то есть

$$\forall x: |x' - x| < r \rightarrow \rho(\Omega_x, \Omega_{x'}) > \varepsilon$$

где Ω_x – окрестность точки x , называемая окном поиска или маской, а $\rho(\Omega_x, \Omega_{x'})$ – функция близости окрестностей.

Метод SURF (Speeded Up Robust Features). Метод SURF (Speeded Up Robust Features) позволяет выполнять поиск особых точек на изображении и рассчитывать их дескрипторы, которые будут инвариантны к масштабу и вращению. Данный метод выполняет поиск особенных точек с помощью матрицы Гессе. Гессеан изображения, который вычисляется как детерминант матрицы Гессе, в точках максимального перепада яркости достигает экстремума. Это позволяет обнаружить на изображении такие особенные точки, как пятна, углы, края линий и др.

Для вычисления матрицы Гессе и фильтров Хаара удобнее перейти к интегральному представлению изображений. Интегральное представление изображения представляет собой матрицу, размерность которой совпадает с размерностью исходного изображения. Элементы матрицы рассчитываются по следующей формуле [3]:

$$D(x, y) = \sum_{i,j=0}^{x,y} S(i, j),$$

где $S(i, j)$ – яркость пикселя исходного изображения. Каждый элемент матрицы представляет собой сумму яркостей в прямоугольнике от $(0, 0)$ до (x, y) .

Перейдя к такому представлению, удобно и легко можно вычислить общую яркость произвольной прямоугольной области АВСЕ:

$$I_{ABCE} = D(A) + D(B) - D(C) - D(E),$$

где АВСЕ – вершины прямоугольника.

Матрица Гессе для двумерной функции и ее детерминант вычисляются следующим образом:

$$H(S(x, y)) = \begin{bmatrix} \frac{\partial^2 S}{\partial x^2} & \frac{\partial^2 S}{\partial x \partial y} \\ \frac{\partial^2 S}{\partial x \partial y} & \frac{\partial^2 S}{\partial y^2} \end{bmatrix},$$

$$\det(H) = \frac{\partial^2 S}{\partial x^2} \frac{\partial^2 S}{\partial y^2} - \left(\frac{\partial^2 S}{\partial x \partial y} \right)^2.$$

Значение определителя этой матрицы достигает экстремума в точках локального максимума и минимума яркости изображения. Элементы матрицы Гессе вычисляются, по сути, как свертка пикселей изображения и маски фильтра. Однако, по мнению авторов, использование вычисления лапласиан гауссиана для этого в исходном виде неэффективно, поэтому в SURF используется бинарный аналог фильтров для вычисления матрицы Гессе. Эти маски более устойчивы к вращению и их можно просто вычислить с помощью интегральной матрицы:

$$\det(H_{approx}) = S_{xx} S_{yy} - (0.9 D_{xy})^2,$$

где S_{xx} , S_{yy} , S_{xy} – свертки по фильтрам.

Для отнесения точки к разряду особенных к гауссиану применяется порог, выше которого, точка считается особенной. Также следует отметить, что гауссиан является величиной производной и не зависит от абсолютного значения яркости, что позволяет обнаруживать те же особенности на изображениях с другим уровнем освещенности.

Как уже говорилось, гессиан зависим от масштаба изображения, поэтому для поиска особенных точек поочередно перебираются различные масштабы фильтров. Размеры таких фильтров не могут принимать произвольные значения, иначе их было бы очень много. Допустимые размеры: 9, 15, 21, 27 и т.д. Для оп-

тимизации перебора фильтров в методе применяется принцип разделения всего диапазона масштабов на октавы. Причем, на октаву приходится несколько фильтров, так как среди разных масштабов и октав одна точка может иметь несколько локальных максимумов гессиана. Исходя из этого, шаг размера фильтра в первой октаве составляет 6, во второй – 12, в третьей – 24 пикселя и т.д., причем октавы перекрывают друг друга для увеличения надежности нахождения точек. По мнению авторов, для изображения с разрешением 1024x768 пикселей требуется около 5–6 октав.

Для вычисления локального экстремума гессиана используется метод соседних точек 3x3x3. Суть метода заключается в том, что точка является локальным максимумом, если ее гессиан больше гессиана соседних точек по принципу 8-связанности в том же масштабе, а также для соседних точек масштабом меньше и больше в данной октаве. При этом понятно, что октава должна содержать не менее трех фильтров.

Для повышения скорости работы метода фильтры октавы вычисляются не для всех пикселей изображения. Первая октава считается для каждого второго пикселя, вторая – для каждого четвертого и т.д., так как размер максимумов пропорционален масштабу фильтра.

Так как алгоритм перебирает не все пиксели изображения, реальный максимум и вычисленный могут расходиться. Для нахождения точного максимума используется интерполирование найденных гессианов куба 3x3x3 квадратичной функцией.

Для того чтобы дескрипторы особенных точек были инвариантны к повороту, необходимо определить преобладающую ориентацию их градиента или вектор ориентации. Для начала вычисляются градиенты в пикселях в окрестности особенной точки. Для этого применяется фильтр Хаара. Он дает точечное значение перепада яркости по осям dX и dY . Данные значения достаточно легко вычисляются с помощью интегральной матрицы. Далее все найденные значения в виде точек наносятся на плоскость $dXdY$ и вычисляется вектор, изображающий приоритетное направление в области особенной точки. Следует отметить, что инвариантность относительно вращения не всегда нужна, и метод допускает не рассчитывать ее.

Для вычисления дескриптора выбирается прямоугольная область вокруг особенной точки. Причем, квадрат ориентируется вдоль направления особенности.

Дескриптор формируется из описания 16 квадрантов вокруг особенной точки. Нужно

отметить, что само изображение при расчете не переворачивается, фильтр считается в обычных координатах, а уже полученные координаты градиента, поворачиваются в соответствии с полученной ранее ориентацией.

Дескриптор особенности изображения состоит из четырех величин:

$$\sum_i^w dX_i, \sum_i^w |dX_i|, \sum_i^w dY_i, \sum_i^w |dY_i|,$$

являющихся суммарными градиентами по каждому из 16 квадрантов. Для всех 16 квадрантов изображения получаем 64 компонента дескриптора. В дополнение к дескриптору для описания точки вычисляется величина:

$$\text{sign}(S_{xx} + S_{yy}),$$

которая позволяет различать темные и белые пятна.

Разработка алгоритма распознавания с использованием библиотеки OpenCV. Рассмотрим этапы разработанного алгоритма распознавания изображений дорожных знаков, а также особенности его программной реализации с использованием функций библиотеки OpenCV.

Как уже было сказано во введении, первым этапом подготовки изображения к распознаванию является улучшение его яркостных характеристик, а также локализация на изображении самого дорожного знака. Одним из преимуществ метода SURF является возможность опустить оба этих этапа. Это обосновывается тем, что:

1. Метод SURF может выполнять поиск и сопоставление дескрипторов точечных особенностей на всем изображении сцены. В вычислительном плане, судя по проведенным испытаниям, это делать быстрее, чем сначала локализовать дорожный знак, а уже в нем – искать особенные точки.

2. Вычисляемый гауссиан является величиной производной и не зависит от абсолютного значения яркости, что позволяет обнаруживать те же особенности на изображениях с другим уровнем освещенности.

Однако, в определенных задачах, локализация дорожных знаков может выполняться. В этом случае можно воспользоваться преобразованием Хафа. Недостатком такого способа является необходимость выполнения преобразования Хафа для каждого типа фигур: окружности, прямоугольника, треугольника. Если задача состоит в распознавании одного типа знаков, например, запрещающих (круглых), этот подход может себя оправдать. В библиотеке OpenCV поиск круглых объектов может доста-

точно легко выполняться с помощью функции `cv::HoughCircles()`. Однако для треугольных и прямоугольных форм преобразование Хафа будет значительно замедлять скорость работы общего алгоритма.

Для работы с особенностями изображений использовались классы `cv::SurfFeatureDetector` и `cv::SurfDescriptorExtractor`.

Класс `cv::SurfFeatureDetector` является детектором особенных точек на изображении, который вычисляет определители матрицы Гессе и отбирает особенности по заданным критериям. Метод класса `detect()` возвращает вектор координат, найденных особенностей. Класс `cv::SurfDescriptorExtractor` выполняет расчет дескрипторов особенных точек, а его метод `compute()` возвращает найденные дескрипторы. В программе поиск и расчет дескрипторов выполнялись для каждой пары сцена–знак.

Для сопоставления особенностей найденных на изображении сцены и каждого знака применялся класс `cv::BruteForceMatcher`. Отличием данного класса по сравнению с `cv::FlannBasedMatcher` является возможность нахождения для каждой особенности одного изображения, заданного количества кандидатов на сопоставление из второго (в данном случае выбирались первые два наиболее вероятных кандидата). Данные сопоставления выполнялись с помощью метода `knnMatch()` класса `cv::BruteForceMatcher`, который возвращает вектор пар особенностей из двух изображений.

Не все найденные соответствия будут являться корректными, поэтому необходим критерий отбора пар особенностей от которого в значительной степени зависит качество распознавания. Такая проверка выполнялась путем вычисления близости дескрипторов двух кандидатов для каждой особенности. Если дескрипторы кандидатов близки, значит данной точечной особенностью можно поставить в соответствие не одну, а множество особенностей второго изображения и, следовательно, данное соответствие считается некорректным. Также близость дескрипторов кандидатов может говорить о нахождении плохо различимой особенной точки, и ее также нужно исключить из рассмотрения. Если дескрипторы двух кандидатов существенно различны, значит, имеется преимущественный кандидат, отражающий ту же точку реальной сцены, и соответствие считается найденным.

Критерием соответствия знака из базы данных знаку из сцены является наличие максимального количества соответствий между точечными особенностями.

Описание структуры программного обеспечения. Разрабатываемая информационная система предназначена для распознавания изображений предупреждающих дорожных знаков. Приложение имеет оконный интерфейс, написано на языке C++ в среде Microsoft Visual Studio 2008 на управляемом коде (CLR). Работа с информационной системой подразумевает загрузку изображения распознаваемой сцены, загрузку перечня изображений дорожных знаков, которые будут являться эталонными при

распознавании. Данный принцип отражен на рис. 1. Результатом работы программы является вывод в системное окно изображения эталонного знака, соответствующего знаку в сцене.

Для графического описания объектно-ориентированного программного кода широко применяется язык UML. В UML используются следующие виды диаграмм: классов, компонентов, структуры, развертывания, объектов, пакетов, профилей, деятельности, состояний, прецедентов и др.

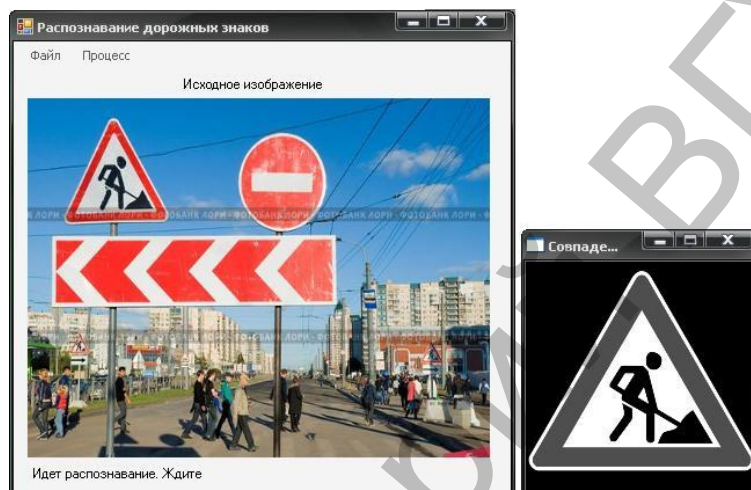


Рис. 1. Основные формы программы.

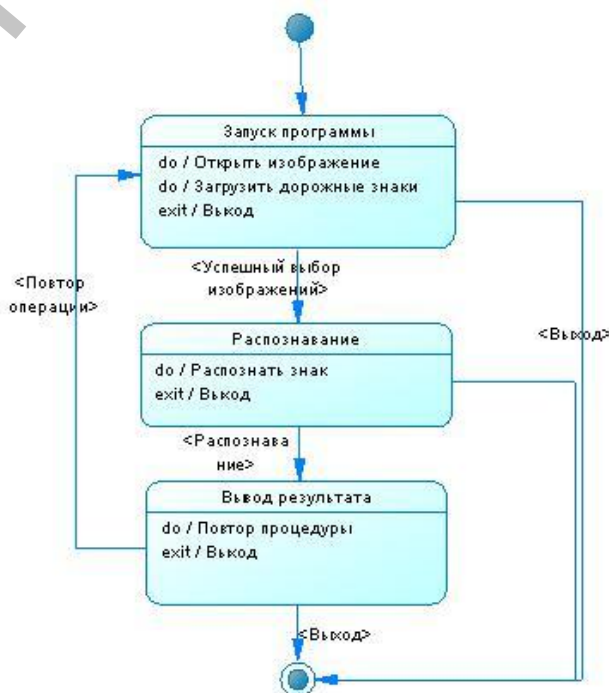


Рис. 2. Диаграммы классов и состояний.

В частности, диаграмма класса Recognition, реализующего основной функционал приложения, отражает все атрибуты класса, операторы и методы, составляющие его (рис. 2). Также приведена диаграмма пользовательского фрагмента класса Form1. Диаграмма состояний отражает состояния системы в понятии теории автоматов. На диаграмме изображаются переходы между состояниями, а также названия событий, вызывающих переход. Все диаграммы построены в системе Sybase Power Designer.

Заключение. Как и для любой системы распознавания, важным этапом ее разработки является определение ограничений и условий, при которых система может эффективно работать. Экспериментально определено, что для наилучшего распознавания знаков рекомендуется соблюдать номинальное соотношение разрешения сцены, эталонного знака и знака в сцене как 5:2:1, угол зрения к плоскости знака в сцене должен составлять не более 15°, расстояние до знака – 50 м.

Для тестирования программы выбирались сцены, удовлетворяющие описанным выше требованиям. В качестве эталонных знаков выбрано 35 знаков из категории «предупреждающие». В результате тестирования программы на 30 сценах точность распознавания составила 85%.

Проведена оценка времени работы алгоритма на тестовых изображениях. Скорость распознавания одного знака с разрешением 550x350 пикселей составила 1 с. Проверка выполнялась на компьютере с процессором AMD Athlon X2 4400+, объемом ОЗУ 2 Гб, среда разработки – Microsoft Visual C++ 2008. Уровень загрузки центрального процессора составлял 35–45%.

Таким образом, в результате исследования разработано программное обеспечение для распознавания изображений дорожных знаков методом SURF (Speeded Up Robust Features), из-

ложенном в [3]. С помощью данного метода можно выполнять поиск особых точек на изображении, рассчитывать их дескрипторы, которые будут инвариантны к масштабу и вращению, а также выполнять их сопоставление. Вычисляемый гауссиан особенности является величиной производной и не зависит от абсолютного значения яркости, что способствует обнаружению тех же особенностей на изображениях с другим уровнем освещенности. Найден критерий отбора корректных соответствий точечных особенностей, повышающий качество распознавания дорожного знака.

К недостаткам метода можно отнести тот факт, что для изображений простой формы, без ярко выраженной текстуры, таких, как дорожные знаки, количество особенностей будет невелико. Однако графические элементы дорожных знаков имеют простую геометрическую форму и цвет, что позволяет получить более различимые дескрипторы. Все описанное помогает сделать вывод о возможности использования подобного подхода для распознавания дорожных знаков, а также необходимости продолжения работы по повышению точности и уменьшению времени работы полученных алгоритмов.

ЛИТЕРАТУРА

1. Попов, Е.Ю. Алгоритм распознавания дорожных знаков ограничения скорости / Е.Ю. Попов, Д.И. Крыжановский // Электронный журнал «Современные научные исследования и инновации» [Электронный ресурс]. – 2013. – Режим доступа: <http://web.snauka.ru/issues/2012/06/14717>. – Дата доступа: 1.03.2013.
2. Чигорин, А. Распознавание знаков дорожного движения на изображениях с обучением на синтетических данных / А. Чигорин, А. Конев, Г. Кривовязь, А. Велижев, А. Конушин // Техническое зрение в системах управления, 2012. – С. 65–68.
3. Bay, H. SURF: Speeded Up Robust Features / H. Bay, A. Ess, T. Tuytelaars, L. Van Gool // Computer Vision and Image Understanding (CVIU). – 2008. – Vol. 110, № 3. – P. 346–359.
4. Краснобаев, Е.А. Алгоритмы сопоставления опорных точек изображения для систем видеодетекции / Е.А. Краснобаев // Вестн. Віцебск. дзярж. ун-та. – 2010. – № 1(55). – С. 126–132.

Поступила в редакцию 02.04.2013. Принята в печать 17.06.2013
Адрес для корреспонденции: e-mail: krasnobaev@tut.by – Краснобаев Е.А.