

Щекачева А.В.,

студент магистратуры

1 курс, факультет «Отдел аспирантуры и магистратуры»

Поволжский государственный университет телекоммуникаций и

информатики

Россия, г. Самара

Куляс О.Л.,

старший научный сотрудник, кандидат технических наук, доцент

Доцент кафедры «Информационные системы и технологии»

Поволжский государственный университет

телекоммуникаций и информатики

Россия, г. Самара

ВЫДЕЛЕНИЕ ЛИНИЙ НА ИЗОБРАЖЕНИЯХ С ПОМОЩЬЮ ПРЕОБРАЗОВАНИЙ ХАФА

***Аннотация:** В статье рассматриваются базовые методы обнаружения разрывов яркости: методы обнаружения точек, методы обнаружения прямой линии, методы обнаружения контура объекта. Подробно рассмотрены алгоритмы обнаружения прямых линий с помощью преобразований Хафа. Выполнено моделирование этих алгоритмов средствами Matlab.*

***Ключевые слова:** Сегментация изображений, детекторы контуров, преобразование Хафа, компьютерное моделирование, Matlab.*

***Abstract:** The article discusses the basic methods for detecting brightness gaps: methods for detecting points, methods for detecting a straight line, methods for detecting the contour of an object. Algorithms for detecting straight lines using Hough transformations are considered in detail. These algorithms were simulated using Matlab.*

Key words: Image segmentation, edge detectors, Hough transform, computer simulation, Matlab.

Задача автоматизированной обработки данных стояла перед человечеством ещё со времён появления самих данных. Но только с появлением цифровых форматов хранения этот процесс значительно продвинулся вперёд.

Со временем распространение цифровых форматов изображений дало ещё больший толчок к освоению области автоматической обработки изображений. И к преобразованию Хафа стали обращаться всё чаще.

Уже на текущий момент преобразование Хафа может применяться в таких областях как распознавание контуров зданий на изображениях, определение линии горизонта, нахождение линий дорожной разметки, распознавание регистрационных знаков автомобилей и прочих сферах.

Таким образом, исследования в данной области с целью изучения данного преобразования, его применимости и поиска методов его оптимизации являются очень актуальными.

Сегментация изображения – это разбиение изображения на множество покрывающих его областей. Сегментация применяется во многих областях, например, в производстве для индикации дефектов при сборке деталей, в медицине для первичной обработки снимков, также для составления карт местности по снимкам со спутников.

Обнаружение точек.

Наиболее общий способ поиска разрывов заключается в обработке изображения скользящей маской. Для маски размерами 3x3 эта процедура использует вычисление линейной комбинации коэффициентов маски со значениями яркости элементов изображения, накрываемых маской. Отклик R такой процедуры в каждой точке изображения задаётся выражением:

$$R = w_1z_1 + w_1z_2 + \dots + w_9z_9 = \sum_{i=1}^9 w_iz_i, \quad (1)$$

где z_i – это значение яркости пиксела, соответствующего коэффициенту маски w_i . Отклик маски присваивается элементу, над которым расположен центр маски [1].

Обнаружение изолированных точек на изображении не представляет большой сложности. Используя маску, представленную на рис. 1, будем считать, что в пикселе под центром маски обнаружена точка, если $|R| \geq T$, где T – это неотрицательное пороговое значение.

-1	-1	-1
-1	8	-1
-1	-1	-1

Рисунок 1. Маска для обнаружения точек

Обнаружение прямой линии.

Следующий уровень сложности заключается в обнаружении линий. Рассмотрим маски, представленные на рис. 2.

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Горизонтальная		+45°		Вертикальная		-45°					

Рисунок 2. Маски для обнаружения линий

Если первую маску перемещать по изображению, то наибольший отклик будет наблюдаться на горизонтальных линиях толщиной в один пиксел. Причем, если яркость фона одинакова, то отклик будет максимальным, когда линия проходит горизонтально через центр маски. Аналогично вторая маска на рис. 2 даст наибольший отклик на линиях, имеющих наклон +45°, третья маска – на вертикальных линиях, а четвёртая – на линиях, имеющих наклон -45°. Эти направления выделены на масках коэффициентами с наибольшим весом (в

данном случае равным 2). Сумма коэффициентов каждой маски равна нулю, что даёт нулевой отклик на областях постоянной яркости.

Обозначим через R_1, R_2, R_3 и R_4 отклики масок, показанных на рис. 2 (слева направо). Пусть изображение обрабатывается независимо каждой маской. Если для некоторой точки изображения $|R_i| > |R_j|$ при всех $j \neq i$, то эта точка, по видимому, лежит на линии, которое имеет направление маски i . Например, если в какой-то точке $|R_1| > |R_j|$ для $j = 2, 3, 4$, то эта точка, скорее всего, принадлежит горизонтальной линии [5]. Альтернативная задача заключается в поиске линий, идущих в заданном направлении. В этом случае можно обработать всё изображение соответствующей маской, применяя пороговое преобразование. Другими словами, если необходимо обнаружить все линии на изображении, идущие в заданном направлении, достаточно пройтись этой маской по всему изображению, сравнивая абсолютные значения откликов с заданным порогом. Выделенные таким способом точки, отвечающие наибольшему отклику, будут ближе всего примыкать к линиям (толщиной в один пиксел), направление которых задано выбранной маской.

Обнаружение контура объекта.

Наиболее актуальной и востребованной является задача выделения контуров. Протяженный перепад яркости называется контуром. На интуитивном уровне, перепад – это связное множество пикселей, лежащих на границе между двумя областями с разной яркостью.

При поиске контуров используются производные первого и второго порядка. Градиентом двумерной функции $f(x,y)$ называется вектор:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \quad (2)$$

Модуль вектора градиента равен:

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} = [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2]^{1/2}. \quad (3)$$

Для упрощения вычислений эту функцию иногда приближают следующей формулой:

$$\nabla f \approx |G_x| + |G_y|. \quad (4)$$

Эта величина ведёт себя примерно как производные, т.е. она равна нулю в областях с постоянной яркостью и её амплитуда пропорциональна скорости изменения яркости там, где яркость пикселей непостоянна. Часто на практике эту приближенную величину называют «градиентом», что не вполне строго.

Основное свойство вектора градиента заключается в том, что он указывает в сторону максимального роста изменения функции f в точке (x, y) . Угол наклона этого вектора равен:

$$a(x, y) = \arctg\left(\frac{G_y}{G_x}\right). \quad (5)$$

Важно уметь приближать производные G_x и G_y численно.

Вторые производные при обработке изображений работают при вычислении оператора Лапласа. Лапласиан двумерной функции $f(x, y)$ равен [2]:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}. \quad (6)$$

Основная идея обнаружения контуров базируется на поиске мест изображения, где яркость меняется быстро, с помощью следующих двух общих критериев:

- 1) Найти места, где первая производная яркости превосходит по модулю некоторый заранее заданный порог.
- 2) Найти места, где вторые производные яркости имеют пересечения нулевого уровня.

В Matlab, с помощью функции IPT edge, можно реализовать несколько оценок производных для использования в приведённых выше критериях. Для некоторых из этих оценок имеется возможность указать, к каким именно перепадам чувствительна данная оценка: к горизонтальным, вертикальным или к перепадам обоих типов. Общий синтаксис этой функции выглядит следующим образом:

```
>>[g,t]=edge(f,'method',parameters),
```

где f – это входное изображение, $method$ – одна из оценок, перечисленных в таблице 1, а $parameters$ – дополнительные параметры. Выходом функции служит логический массив g , в котором стоят 1 там, где обнаружены точки перепада на f , и 0 там, где перепады не обнаружены. Аргумент t является необязательным, в него записывается значение порога, которое функция $edge$ использовала для сравнения с градиентами точек изображения. На рис. 3 представлены маски детекторов и реализуемые ими формулы приближения.

Таблица 1

Детекторы контуров, доступные в функции $edge$

Детектор контура	Основные свойства
Собела	Обнаруживает края с помощью приближений Собела первых производных, заданных на рис. 3.
Превитт	Обнаруживает края с помощью приближений Превитт первых производных, заданных на рис. 3.
Робертса	Обнаруживает края с помощью приближений Робертса первых производных, заданных на рис. 3.
Лапласиан гауссиана	Обнаруживает края, выполняя поиск пересечений нулевого уровня после фильтрации $f(x, y)$ гауссианом.
Пересечения нулевого уровня	Обнаруживает края, выполняя поиск пересечений нулевого уровня после фильтрации $f(x, y)$ фильтром, заданным пользователем.
Канни	Обнаруживает края, выполняя поиск локальных максимумов градиента $f(x, y)$. Градиент вычисляется от гауссиана. Метод использует два порога для нахождения сильных и слабых краёв. Следовательно, этот метод с большей вероятностью обнаруживает настоящие слабые края.

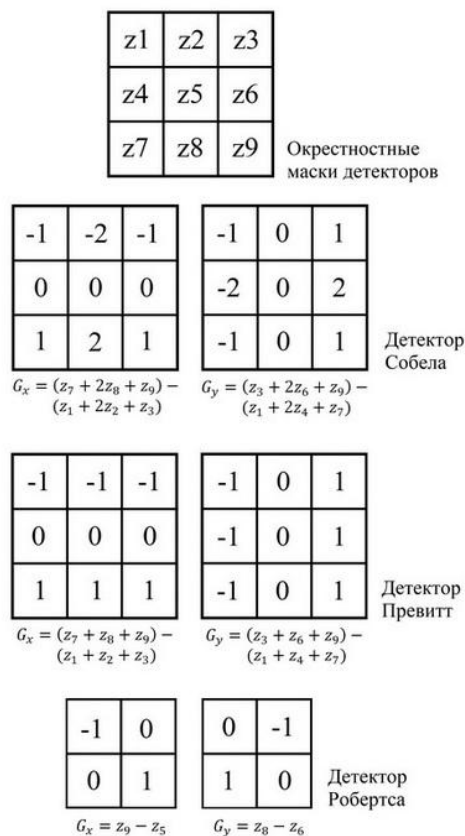


Рисунок 3. Маски детекторов и реализуемые ими формулы приближения для первых производных

Пример обнаружения контуров разными методами приведён на рис. 4.

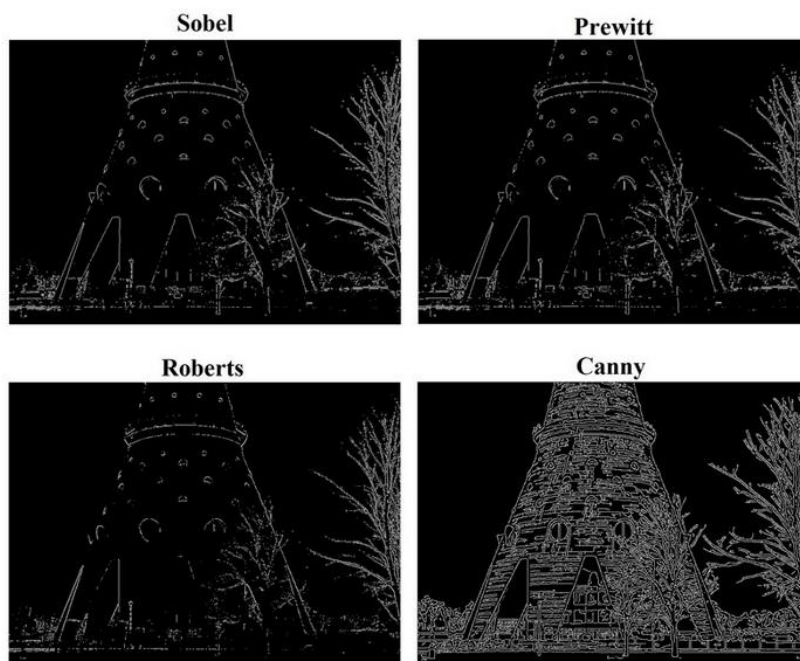


Рисунок 4. Результаты работы детекторов с параметрами по умолчанию

Обнаружение линий с помощью преобразования Хафа.

Рассмотренные ранее методы должны обнаруживать только пиксели, принадлежащие краям и перепадам яркости. Но на практике выделенные пиксели редко относятся только к этой категории в силу многих причин: воздействия шума, разрыва краёв из-за неравномерного освещения и других факторов, которые вносят ложные перепады яркости в изображения. Поэтому за алгоритмом обнаружения краёв обычно следует процедура компоновки выделенных пикселей краёв в настоящие, осмысленные линии и краевые сегменты. Один из подходов к выполнению подобных действий основан на преобразовании Хафа.

Рассмотрим любую точку (x_i, y_i) и все проходящие через неё прямые. Все такие прямые удовлетворяют уравнению $y_i = ax_i + b$ для произвольных значений a и b . Однако если переписать это уравнение в виде $b = -ax_i + y_i$ и рассмотреть соответствующую ab -плоскость (называемую пространством параметров), то оно задаст единственную прямую для каждой фиксированной координатной пары (x_i, y_i) . Более того, другой точке (x_j, y_j) соответствует своя прямая в пространстве параметров, и эти две прямые пересекаются в некоторой точке (a', b') , где a' – это угловой коэффициент, а b' – точка пересечения с осью y прямой, проходящей через точки (x_i, y_i) и (x_j, y_j) на xu -плоскости. На самом деле, у каждой точки этой прямой имеется прямая в пространстве параметров, причем все такие прямые пересекаются в точке (a', b') . Эти понятия иллюстрируются на рис. 5.

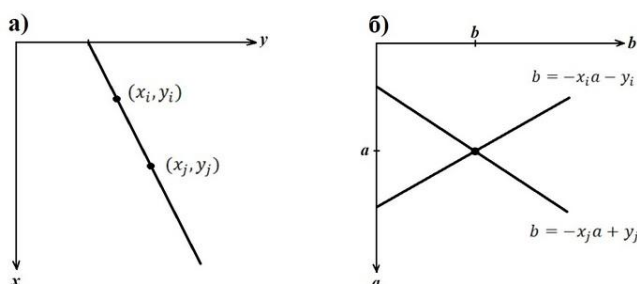


Рисунок 5. Основные понятия преобразований Хафа

а) Плоскость xu ; б) Пространство параметров ab

Первый шаг использования преобразования Хафа для обнаружения линий и связывания состоит в нахождении локальных максимумов преобразования. Поиск множества максимумов преобразованием Хафа может быть вполне перспективным. В силу квантования пространства цифрового изображения и пространства параметров, а также по причине того, что края и перепады на типичных изображениях не являются совершенно прямыми, максимумы преобразования Хафа могут достигать более чем в одной ячейке накопления [3]. Эту сложность можно преодолеть с помощью следующей стратегии:

- 1) Найти ячейку преобразования Хафа, в которой лежит наибольшая величина, и записать её местоположение.
- 2) Обнулить ячейки в ближайшей окрестности положения, найденного на шаге 1.
- 3) Повторять шаги 1 и 2 до тех пор, пока желаемое число максимумов не будет найдено, или после достижения заданного порога.

Функция `houghpeaks` реализует эту стратегию.

После обнаружения множества локальных максимумов преобразования Хафа остаётся определить, есть ли сегменты краёв, проходящие по соответствующим линиям, а также где они начинаются и заканчиваются. Для каждого максимума сначала следует найти положения всех ненулевых пикселей изображения, которые лежат на соответствующих прямых. Для этих целей существует функция `houghpixels`.

Пикселы, обнаруженные функцией `houghpixels`, необходимо сгруппировать в сегменты. Для этого можно воспользоваться следующей стратегией:

- 1) Повернуть пикселы на угол $90^\circ - \theta$ так, чтобы они легли примерно вдоль вертикальной прямой.
- 2) Упорядочить пикселы в порядке возрастания величин их повернутых x -координат.
- 3) Использовать функцию `diff` для определения зазоров (щелей). Заполнить малые зазоры. Это даёт эффект слияния примыкающих сегментов линий, которые разделены малыми промежутками.

- 4) Возвратить информацию о сегментах линий, которые длиннее некоторой минимальной пороговой длины.

Проведём моделирование с целью выделения объекта интереса, которым является номерная пластина регистрационного знака автомобиля.

Загрузим исходное изображение автомобиля в рабочее пространство Matlab, рис. 6, а). Полученное на рис. 6, б) бинарное изображение используем для нахождения контуров методом Собела, результат показан на рис 6, в). К контурному изображению применим преобразование Хафа, используя функцию Matlab. На рис. 6, г) представлен результат преобразований Хафа с максимумами. На рис. 6, д) и 6, е) показаны результаты выделения прямых линий на исходном изображении и на контурном.

Анализ изображений показывает, что номерная пластина идентифицирована как замкнутая прямоугольная область, ограниченная прямыми линиями, обнаруженными с помощью преобразования Хафа. Другие прямые линии, которые есть на изображении, соответствующие отрезкам прямых на контурном изображении не образуют замкнутых областей и могут быть легко отфильтрованы при дальнейшей обработке изображения.

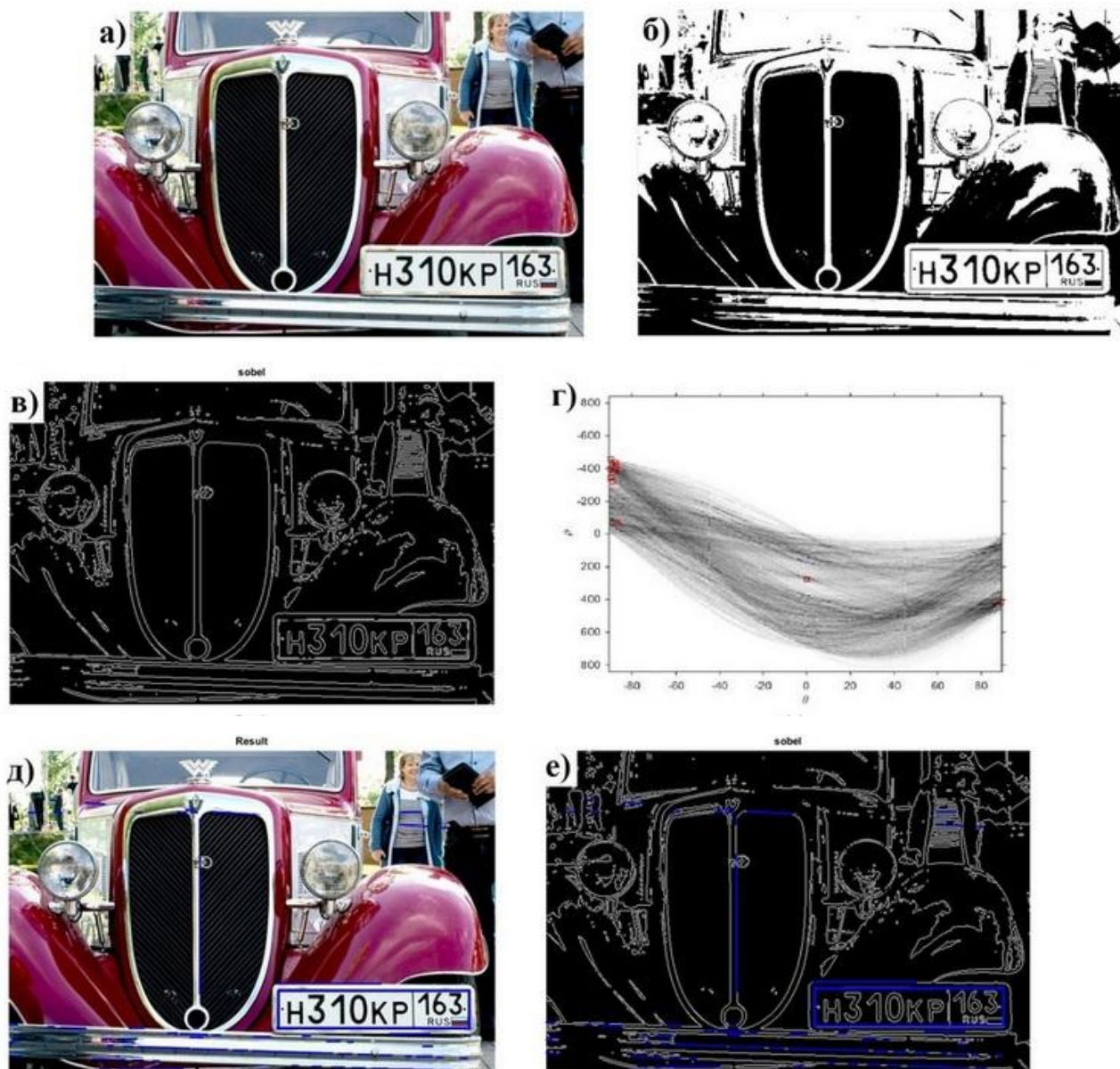


Рисунок 6. Пример работы программного кода

а) Исходное изображение; б) Бинарное изображение; в) Найденные контура на изображении; г) Преобразование Хафа с локальными максимумами; д) Сегменты линий, соответствующие максимумам преобразования Хафа; е) Сегменты линий, соответствующие максимумам преобразования Хафа на контурном изображении

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Запрягаев С.А. Программная оболочка для поиска примитивов на изображении – Воронеж, 2008. – № 2. – С. 37-47.

2. Сойфер В.А. Методы компьютерной обработки изображений – М., Физматлит, 2003. – 784 с.
3. Гонсалес Р. Цифровая обработка изображений в среде MATLAB – М., Техносфера, 2006. – 616 с.
4. Форсайт, Д. Компьютерное зрение. Современный подход – М., Издательский дом "Вильямс", 2004. – 928 с.