

# Постобработка прямых и ломаных, полученных методом Хафа\*

Л. Новоторцев, А. Волобой

torets13@gmail.com | voloboy@gin.keldysh.ru

Институт ИПМ им. М.В. Келдыша РАН, Москва, Россия

*В задачах распознавания изображений часто возникает необходимость выделения прямых, ломаных или многоугольных областей. Одним из самых распространённых методов решения этой задачи является метод Хафа. Однако, ввиду особенностей реализации этого метода и характеристик изображения, прямые могут получаться разорванными. Данная статья посвящена постобработке результатов, полученных методом Хафа, с целью устранить разрывы прямых и приблизить полученный результат к выделению, выполненному человеком.*

**Ключевые слова:** компьютерное зрение, метод Хафа, распознавание образов

## 1. Введение

Среди задач компьютерного зрения нередко встречаются подзадачи, связанные с выделением различных примитивов: отрезков, ломаных, окружностей, овалов и т. д. К одним из самых распространённых примитивов относятся прямые, отрезки и ломанные (в том числе и замкнутые). Особенно необходимость выделения прямых и отрезков часто встречается в задачах, связанных с обнаружением и реконструкцией по снимкам, в связи с тем, что в подавляющем большинстве случаев строения имеют линейные черты.

Одним из распространённых способов обнаружения прямых является метод Хафа [6]. Суть этого метода заключается в том, что сначала изображение обрабатывается фильтром Кэнни [3], после чего собирается статистика по выделенным этим фильтром граням и на её основе делается вывод о наличии на изображении прямой. На данный момент существует много модификаций метода Хафа [5,8,9,10], но практически у всех них есть общий недостаток: разрывы между найденными отрезками. Проблема заключается в том, что интересные объекты на изображениях не всегда имеют чёткую границу, которая обнаруживается фильтром Кэнни, и как следствие непосредственно методом Хафа. В некоторых случаях такие артефакты не имеют значения, поскольку алгоритмы, обрабатывающие полученные методом Хафа результаты сами могут восстановить пропуски [1]. Однако нередко встречаются и такие случаи, когда более полное выделение отрезков может оказать значительное влияние на полученные результаты.

Данная статья посвящена автоматической обработке и анализу прямых и отрезков, полученных методом Хафа. Описанный метод будет принимать решения об объединении отрезков в один автоматически на основе проведённого анализа. Помимо этого данный метод можно распространить на построение ломаных.

Для понимания рассуждений, приведённых в данной статье, рассмотрим основные этапы фильтра Кэнни и алгоритма Хафа

## 2. Фильтр Кэнни

Фильтр Кэнни можно разделить на четыре последовательно выполняемых шага.

На первом шаге к изображению применяется фильтр Гаусса. Это делается для устранения шума, который может давать ложные грани.

На втором шаге происходит вычисление градиентов по осям X и Y на основе фильтра Собеля [7]. Для градиентов в каждой точке вычисляются их величина и угол (направление) с округлением до 0, 45, 90, 135 градусов.

На третьем шаге выполняется "подавление немаксимумов". Суть этого шага заключается в том, что все пиксели, которые не имеют максимальную величину градиента среди соседних пикселей по направлению (как положительному, так и отрицательному) градиента, помечаются как не относящиеся к грани.

На четвёртом шаге выполняется двойная пороговая фильтрация. Все пиксели, у которых величина градиента превышает верхний порог, помечаются как грани ("сильные грани"). Те, у которых величина градиента меньше нижней планки, помечаются как не относящиеся к грани. Остальные помечаются как "слабые грани".

На последнем этапе происходит "трассировка области неоднозначности". Все пиксели, помеченные, как "слабая грань" и не имеющие в своей окрестности сильных граней, подавляются.

Существует много модификаций фильтра Кэнни [2, 4], которые улучшают те или иные этапы этого алгоритма, но общая концепция алгоритма изменяется незначительно.

## 3. Метод Хафа

Метод Хафа основан на методе голосования. Общий подход заключается в том, что для каждого пикселя, относящегося к грани, строятся всевозможные прямые, проходящие через этот пиксель.

Работа опубликована по гранту РФФИ №16-07-20482.

Если рассматривать параметрическое задание прямых, то получается, что для данного пикселя мы перебираем углы с некоторым шагом. Для точки и угла вычисляем величину нормали, опущенной из начала координат. Полученные параметры (угол и нормаль) округляем в соответствии с наперёд заданной точностью и прибавляем единицу в таблицу (строки - углы, столбцы - нормаль). Далее, после заполнения таблицы проводится её анализ и прямые, соответствующие локальным максимумам в таблице, помечаются как найденные.

Существует большое количество модификаций, направленных на уменьшение ошибок дискретизации [5,10], подбору критерия определения выделения прямой по таблице и скорости работы метода в целом.

В данной статье предлагается использовать следующий подход. В первую очередь всё исходное изображение разбивается на одинаковые квадратные сегменты (возможно за исключением крайних областей). Размер области определяется тем, насколько большой точности необходимо добиться и насколько точно надо определить длину отрезка. Чем больше сторона области, тем точнее можно определить угол поворота прямой в каждом сегменте. С другой стороны при увеличении области увеличивается погрешность определения концов отрезка. Это вызвано тем, что метод Хафа находит прямые, но не концы отрезков. В связи с этим концы отрезков определяются границами областей. В таблице 1 приведены соответствие размера стороны области и погрешности в определении угла.

Таблица 1: Погрешность определения угла и минимальные размеры сегмента для обеспечения этой погрешности.

$\Delta\theta$	1	2	3	4	5	6	7	8	9	10
$L$	82	41	28	21	17	14	12	11	10	9

Далее в каждой области применяется метод Кэнни. В данной статье предлагается использовать метод, основанный на анализе окрестностей параметров, вместо их конкретных значений [5]. Это позволит минимизировать ошибку при вычислении нормали. Прямые, полученные на этом шаге предлагается обрезать по границам областей.

## 4. Объединение отрезков

### 4.1 Объединение по областям

После получения отрезков в каждой области необходимо выполнить объединение отрезков. На этом этапе необходимо учитывать некоторые аспекты. Во-первых, в некоторых областях отрезки могут

быть пропущены из-за малой контрастности снимка или из-за особенностей работы метода Хафа. Вторых, отрезки, принадлежащие к одной линии, могут иметь разные углы из-за ошибок дискретизации.

В первую очередь все найденные отрезки помечаем как необработанные. Далее последовательно проходим по всем сегментам, на которые было разбито изображение. В каждом сегменте обрабатываем полученные в нём отрезки, которые всё ещё помечены как необработанные.

Для дальнейших рассуждений будем полагать, что погрешность вычисления угла составляет  $\Delta\alpha$ .

Для каждого отрезка сначала вычисляем участок границы сегмента (далее будем называть это образом отрезка), на котором может находиться конец отрезка с учётом погрешности (на рис. 1 выделен красным). Продолжение отрезка ищется на тех соседних сегментах, на границу которых попал вычисленный нами участок. В выбранных сегментах ищутся отрезки, которые имеют угол, отличающийся от рассматриваемого не более чем на  $2 \times \Delta\alpha$ . Это обусловлено тем, что угол каждого из отрезков может отклоняться от фактического угла прямой, которую хотим найти на  $\Delta\alpha$ , но в разные стороны. Для искомого отрезка также вычисляется его образ. Объединение отрезков происходит в том случае, если их образы имеют непустое пересечение. При этом углом новообразованного длинного отрезка считается среднее арифметическое их углов. Помимо этого для объединённого отрезка записывается количество отрезков, в него вошедших.

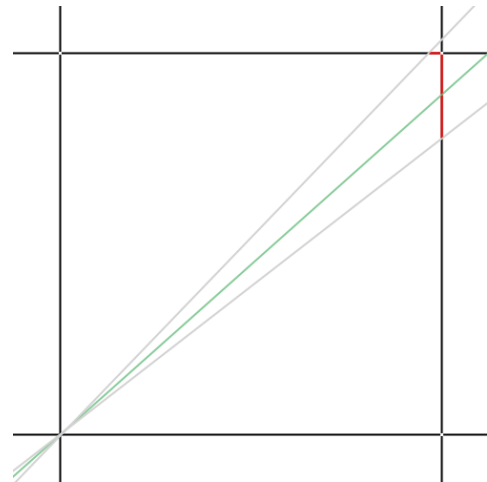


Рис. 1: Сегмент (черный) и найденная в нём прямая (зелёный). Серым цветом обозначены максимальные возможные отклонения (погрешности). Красным - участки сегмента, где может заканчиваться отрезок с учётом погрешности.

Дальнейшее объединение отрезков происходит аналогичным образом, за тем исключением, что для "большого" объединённого отрезка погрешность рассчитывается по формуле:

$$\Delta\alpha' = \frac{2 \times \Delta\alpha}{N} \quad (1)$$

где  $N$  - количество отрезков, вошедших в данную прямую.

В случае если на каком-либо этапе в соседних сегментах не найдено продолжения для отрезка, то отрезок мнимо продолжается на этот сегмент. Если длина мнимого продолжения не превышает  $2/3$  стороны сегмента, то далее проверка ведётся, как если бы этот сегмент реально существовал, за тем исключением, что величина  $N$  не увеличивается. Если и на следующей итерации не обнаружилось сегмента или его мнимая длина превышает  $2/3$  стороны сегмента, то отрезок помечается законченным (с рассматриваемой стороны).

Для каждого отрезка запоминается его  $\Delta\alpha'$ , полученная в результате всех итераций по формуле (1), а также погрешность  $\Delta\rho/N$ , где  $\Delta\rho$  - погрешность величины нормали, определяемая на этапе, описанном в разделе 3.

## 4.2 Общее объединение

После полного выполнения операций, описанных в разделе 4.1, мы получаем набор отрезков, однако для того, чтобы максимально устранить разрывы, необходимо продолжить слияние отрезков, только уже используя больше информации об изображении.

### 4.2.1 Сбор статистики и расчёт параметров

На данном этапе, при сборе статистики будем рассматривать только те отрезки, которые были получены на предыдущем этапе путём хотя бы одного объединения отрезков. При группировке и объединении мы будем рассматривать все отрезки.

Для дальнейшего исследования полученных отрезков производятся следующие действия. Во-первых, вычисляется средняя длина полученных отрезков (далее СДО). Также для них вычисляется среднеквадратичное отклонение (СКОДО). Во-вторых, все отрезки группируются по углам наклона для ускорения дальнейших вычислений. Шаг группировки определяется как величина погрешности угла наклона  $\Delta\alpha$ , полученная на этапе применения метода Хафа. В зависимости от требований к скорости работы алгоритма можно также выполнить группировку по зонам на изображении, но ввиду нетривиальности данной задачи и незначительном её влиянии на конечный результат эта оптимизация в данной статье не рассматривается.

Далее для каждого отрезка (из объединённых) ищется не более двух отрезков, выполняющих следующие условия:

- они находятся с ним в одной или в соседних группах по углам
- расстояние не превышает заданной пользователем величины (в случае аэрофотоснимков - порядка размера нескольких зданий). В общем случае данный пункт необязателен, но он позволяет ускорить и улучшить качество алгоритма
- между отрезками можно провести общий перпендикуляр (с некоторой точностью), не пересекающий продолжения отрезков (только сами отрезки)
- искомые отрезки находятся в разных полуплоскостях, разделённых прямой, на которой лежит рассматриваемый отрезок
- отрезки имеют минимальные расстояния среди всех прямых, удовлетворяющим предыдущим свойствам.

После этого рассчитывается среднее расстояние между отрезками (СРО) и среднее квадратичное отклонение это расстояния (СКОРО).

### 4.2.2 Объединение отрезков

Для каждого отрезка рассматриваются отрезки из его группы по углам или соседних групп.

Пара отрезков объединяется, если:

1. Разница между их углами не превышает суммы погрешностей по углам этих отрезков.
2. Разница между величинами нормалей не превышает сумму погрешностей величин нормалей двух отрезков
3. Отрезки не имеют пересечения
4. Расстояние между ближайшими концами разных отрезков не больше чем СРО-СКОРО.

Отрезки отмечаются как разные, если не выполняется один из первых трёх пунктов или расстояние между ближайшими концами отрезков не меньше чем СДО+СКОДО.

В остальных случаях (пункты 1-3 выполнены, а расстояние между ближайшими концами больше, чем СРО-СКОРО и меньше, чем СДО+СКОДО) проводится дополнительный анализ области между отрезками. В этой области применяется снова фильтр Кэнни, но уже с входными параметрами, делающими его более чувствительным. После применения фильтра Кэнни рассматривается область, где должна проходить линия соединения отрезков и её окрестность (2х пикселей достаточно). Если в этой области достаточно количество пикселей, определённых фильтром как грани, то отрезки соединяются. В противном случае они помечаются как разные. Критерий, определяющий достаточность количество пикселей, задаётся на основе расстояния между концами отрезков и процентного

допуска, задаваемого как вход алгоритма и определяющего его чувствительность.

## 5. Объединение отрезков в ломанные

Используя аналогичный подход можно проводить объединение построенных отрезков в ломанные. Существенные различия с методом, описанным в предыдущем разделе, заключаются в следующем: рассматривается возможность продления каждого из отрезков до пересечения друг с другом и вместо пунктов 1-3 из предыдущего раздела используется отрицание пункта 1. В остальном рассуждения остаются те же.

Также стоит отметить особый случай: один из отрезков или его предполагаемое продолжение разбивает другой отрезок на две части. В этом случае имеет смысл разбить данный отрезок на два для упрощения дальнейших действий.

## 6. Результаты

На рисунке 2 показано исходное изображение.



Рис. 2: Исходное изображение.

На рисунке 3 показано изображение, полученное в результате использования описанного метода без слияния линий в ломанные.



Рис. 3: Результаты, полученные рассматриваемым алгоритмом..

Из иллюстраций видно, что предложенный алгоритм точно находит границы крыши даже в областях с нечёткими гранями. Помимо этого, рассматриваемый алгоритм дал небольшое количество ложных срабатываний на участках с лесом.

К недостаткам можно отнести пропуск мелких деталей на крышах строений, таких как трубы.

## 7. Заключение

В данной статье приведён алгоритм постобработки метода Хафа обнаружения прямых и отрезков с некоторой модификацией самого алгоритма Хафа. Результаты использования данного алгоритма показывают большое приближение полученных данных к данным, взятым при ручном выделении прямых человеком.

Результаты, полученные представленным методом, позволяют значительно снижают вычислительную нагрузку и увеличивают точность дальнейшей обработки за счёт точного выделения отрезков с малым количеством ложных срабатываний.

## Литература

- [1] Baillard C., Zisserman A. A plane-sweep strategy for the 3D reconstruction of buildings from multiple images. *International Archives of Photogrammetry and Remote Sensing*, 33(B2) part 2, pp 56-62, 2000.
- [2] Bao P., Zhang D., Wu X. Canny edge detection enhancement by scale multiplication. *IEEE transactions on pattern analysis and machine intelligence*, 27(9), pp. 1485-1490, 2005.
- [3] Canny J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, pp. 679-698, 1986.
- [4] Dollár P., Zitnick C. L. Fast edge detection using structured forests. *Pattern Analysis and Machine*

- Intelligence, IEEE Transactions, 37(8), pp. 1558-1570, 2015.
- [5] Du, S., van Wyk, B. J., Tu, C., & Zhang, X. An improved Hough transform neighborhood map for straight line segments. IEEE Transactions on Image Processing, 19(3), pp. 573-585, 2010.
- [6] Hough P. V. C. Method and Means for Recognizing Complex Patterns, U.S. Patent 3069654, Dec. 18, 1962.
- [7] I. Sobel and G. Feldman. A  $3 \times 3$  Isotropic Gradient Operator for Image Processing. R. Duda and P. Hart (Eds.), Pattern Classification and Scene Analysis, pp. 271-272, 1973.
- [8] Li H., Lavin M. A., Le Master R. J. Fast Hough transform: A hierarchical approach. Computer Vision, Graphics, and Image Processing, 36(2), pp. 139-161, 1986.
- [9] Nguyen T. T., Pham X. D, and Jeon J. An improvement of the Standard Hough Transform to detect line segments. IEEE International Conference on Industrial Technology, pp. 1-6, 2008.
- [10] O'Rourke J. Dynamically Quantized Spaces for Focusing the Hough Transform. IJCAI, 81, pp. 24-28, 1981.