Line fitting, или методы аппроксимации набора точек прямой

Анна Дегтярева anna_d_666@mail.ru
Вежневец Владимир vvp@graphics.cs.msu.su

Содержание

Во многих алгоритмах распознавания встречается задача аппроксимации набора точек на изображении аналитической кривой. В статье рассмотрены классические алгоритмы решения этой задачи для случая аппроксимации прямых.

В общем виде задача распознавания формулируется следующим образом: разделить имеющиеся объекты на группы по определенным характеристикам – например цвету, форме. Частным случаем является задача разбиения множества точек на группы в зависимости от типа геометрической формы, образованной группами точек множества (например, точки, образующие прямую или точки, образующие сектор окружности). Основные вопросы, возникающие при решении этой задачи, таковы:
- какую кривую составляют точки на изображении (какому уравнению удовлетворяют координаты точек)?
- какие из точек принадлежат конкретной кривой на изображении?
- сколько всего кривых на изображении?

Преобразование Хафа (описанное в [1]) дает ответ на все три вопроса, однако обладает рядом серьезных недостатков, из-за чего не всегда возможно его применение в практических приложениях. Во-первых, трудно подобрать оптимальное разбиение фазового пространства - в зависимости от насыщенности элементами, масштаба и уровня шума исходного изображения размер ячеек фазового пространства необходимо определять эмпирически, что неприемлемо для автоматических систем. Кроме того, алгоритм Хафа чувствителен к некоторым видам шума.

Каждый из описанных ниже алгоритмов дает ответ лишь на один из вопросов, и применим для частных случаев, но при удачной комбинации результат более устойчив к малым изменениям входных данных и обладает большей точностью.

Ниже описаны методы:

- Метод наименьших квадратов.
- Метод последовательных приближений.
- Метод k-средних.
- Модификация метода наименьших квадратов (M-estimators).
- Метод RANSAC.

Метод наименьших квадратов.

Постановка задачи. Входными данными является набор точек на изображении. Требуется найти параметры прямой, наилучшим образом приближающей этот набор.

Алгоритм. Для решения задачи достаточно подобрать такой набор параметров прямой, при которых все точки изображения были бы расположены к ней максимально близко. Выберем, например, прямую, которая как можно более точно приближает значения y-координат каждой точки (x_i, y_i) изображения при совпадающих x-координатах.
Прямая однозначно определяется парой параметров (a, b), где a – тангенс угла наклона прямой, b – расстояние

до прямой по оси ординат (уравнение прямой в виде $y = ax + b$ или, что то же самое, $y - ax - b = 0$). При этом разница y-координат произвольной точки изображения $(x_i\ y_i)$ и точки прямой с той же x-координатой вычисляется по формуле $y_i - ax_i - b$. Таким образом, оптимальные параметры прямой определяются минимизацией суммы

$$\sum_i (y_i - ax_i - b)^2$$

Однако, при таком подходе метод теряет точность с увеличением угла наклона результирующей прямой, а если прямая вертикальна, то вовсе не сходится. Восстановить точность можно, если вместо разницы координат считать расстояние, то есть длину перпендикуляра, от точки до прямой (см Рис 1.).

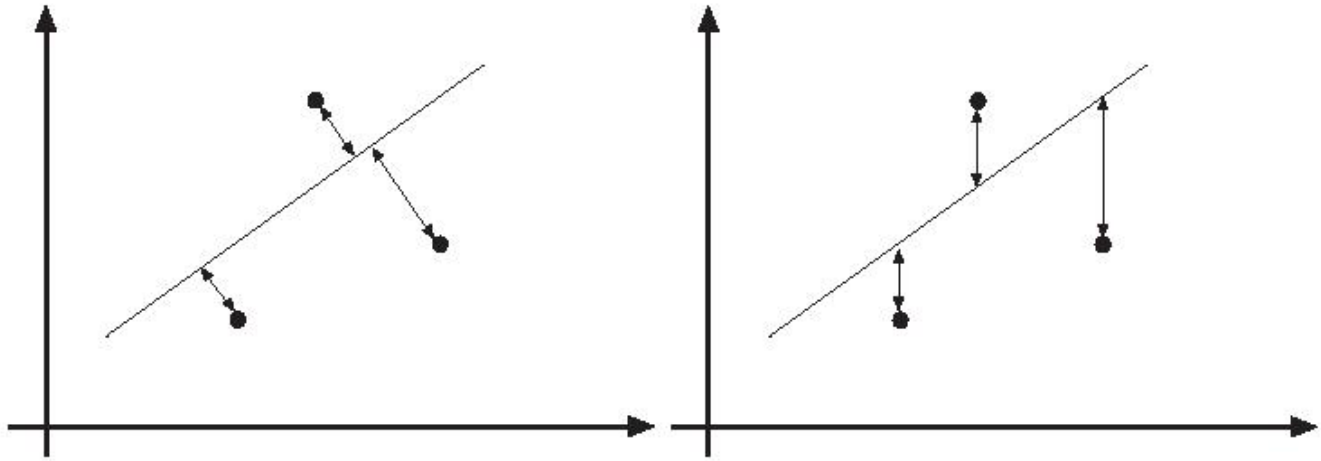

Рис 1. Варианты расчета расстояния до прямой в методе наименьших квадратов

Представим уравнение прямой в виде $ax + by + c = 0$. Заметим, что такое представление не единственно: тройка (a, b, c) определяет ту же прямую, что и тройка $(\lambda a, \lambda b, \lambda c)$ для любого числа $\lambda \neq 0$, поэтому для определенности вводится нормализующее правило, например, $a^2 + b^2 = 1$. При таком представлении прямой длина перпендикуляра из точки (u, v) к прямой выражается формулой $au + bv + c$ (при $a^2 + b^2 = 1$). Тогда задача аппроксимации сводится к минимизации функции

$$\sum_i (ax_i + by_i + c)^2$$

при условии $a^2 + b^2 = 1$.

Метод последовательных приближений.

Постановка задачи. На изображении - набор точек, представляющий собой некоторую связную кривую. Нужно найти набор прямых, аппроксимирующих эту кривую.

Алгоритм. Через любые две соседние точки можно провести только одну прямую. Далее, если каждая следующая соседняя точка лежит достаточно близко к уже проведенной прямой, прямая корректируется с учетом этой точки. В противном же случае новая точка и следующая за ней считаются началом следующей прямой (см. Рис 2). Алгоритм заканчивает свою работу, когда все точки будут приписаны той или иной прямой.

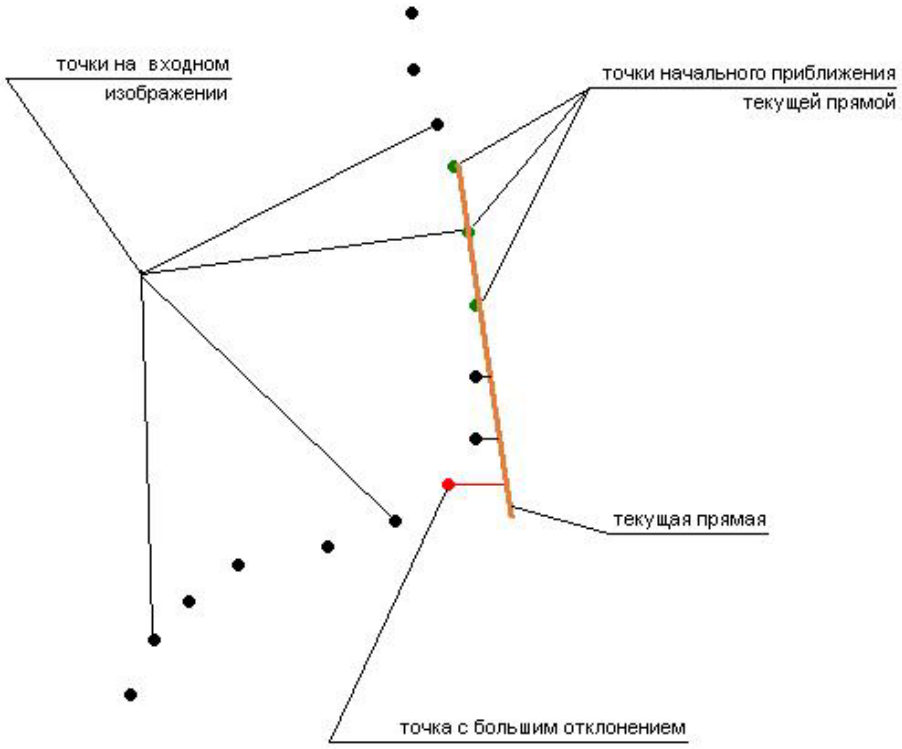

Рис 2. Работа итерационного алгоритма. Зеленым обозначены начальные точки текущей прямой, красным – точка, принадлежащая следующей прямой.

Метод k-средних.

Постановка задачи. Входными данными является набор точек на изображении. Известно, что точки образуют k прямых. Необходимо определить какие точки какой прямой принадлежат.

Алгоритм. Наилучшее решение будет достигнуто при минимизации функции

$$\sum_{l_i \in \text{lines}} \sum_{x_j \in \text{data due to } i\text{'th line}} \text{dist}(l_i, x_j)^2$$

Здесь $\texttt{dist}(l_i, x_j)$ – расстояние от точки x_i до прямой l_i. Минимизация проводится как по параметрам прямых, так и по набору точек, приписанному к каждой прямой. При большом количестве точек параметрическое пространство для перебора очень велико. Для решения этой проблемы (при известном числе прямых k) легко приспособить алгоритм k-средних. Алгоритм состоит из двух многократно повторяющихся шагов:

- каждая точка приписывается прямой, к которой она ближе всего расположена (расстояние до которой минимально),
- параметры прямых модифицируются так, чтобы наилучшим образом аппроксимировать приписанные к ним точки.

Полная схема метода k-средних:

1. Выберем (например случайным образом) k прямых
2. До тех пор пока что-то меняется
3. Каждую точку приписать к ближайшей прямой
4. Пересчитать параметры прямых под новый набор точек

Модификация метода наименьших квадратов (M-estimators).

Большой недостаток метода наименьших квадратов – высокая зависимость от начальных данных, и, как следствие неустойчивость к наличию ошибочных точек. Так, например, если среди входных данных координаты одной точки имеют сильное отклонение от остальных, то результат аппроксимации может получиться неточным (см.Рис 3).

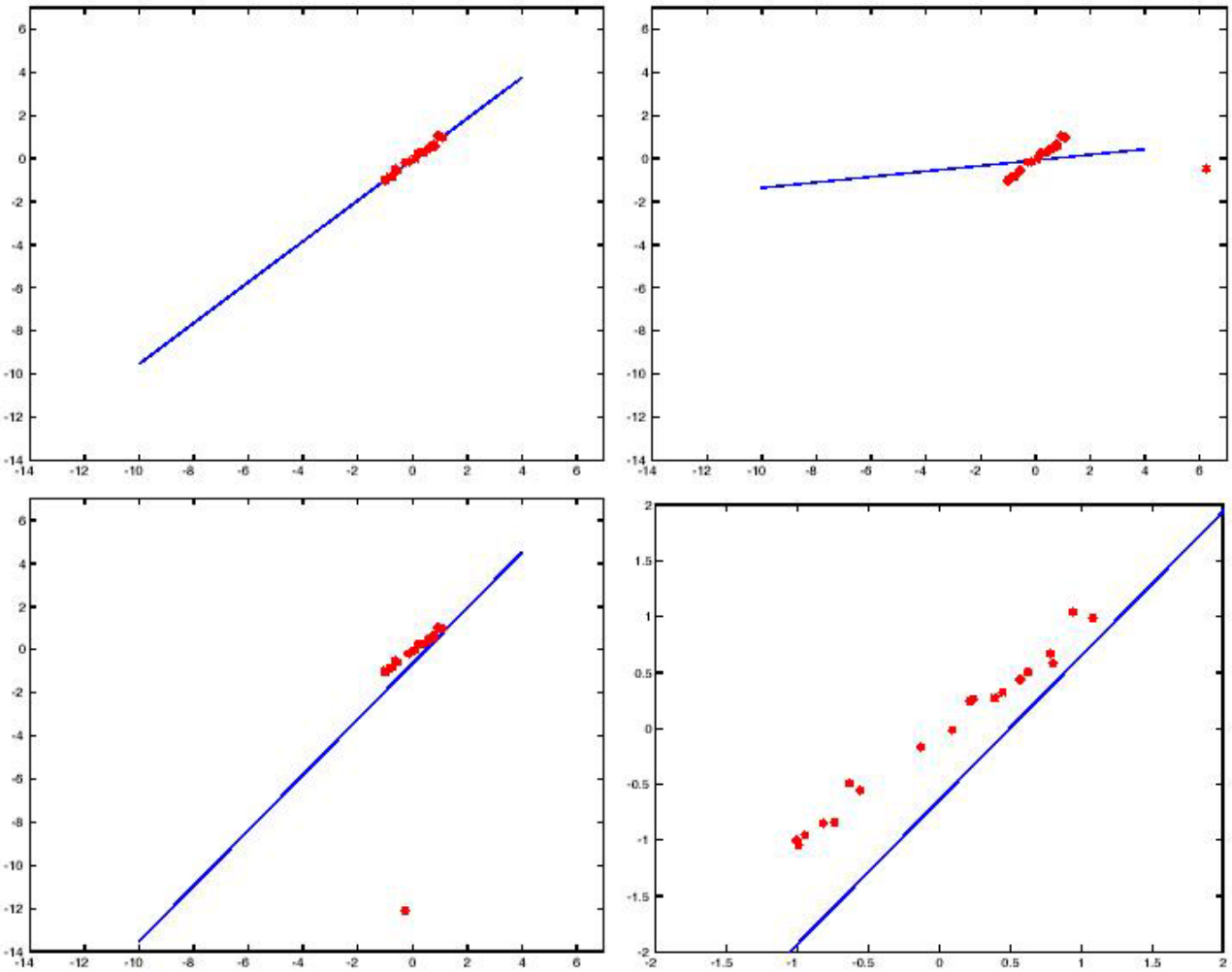

Рис 3. Влияние входных точек с сильным отклонением на результат работы алгоритма аппроксимации методом наименьших квадратов. *Сверху слева* – пример корректной работы алгоритма, *сверху справа* – добавлена точка с большим отклонением по оси абсцисс, *снизу слева* – добавлена точка с большим отклонением по оси ординат, то же в увеличенном масштабе *снизу справа*.

Чтобы избежать подобного эффекта, вводится функция специального вида, модифицирующая расчет суммы квадратов расстояний от точек до прямой таким образом, что влияние каждой конкретной точки уменьшается по мере ее удаления от прямой. Назовем эту функцию $\rho(r, \sigma)$, где r – расстояние от точки до прямой в какой-либо метрике, σ - параметр, определяющий скорость уменьшения влияния точки. Часто функция ρ имеет вид:

$$\rho(u;\sigma) = \frac{u^2}{\sigma^2 + u^2}$$

Учитывая, что расстояние от точки до прямой – это функция $r = r(x, \theta)$, где θ - вектор параметров прямой, x – координатный вектор точки, задача аппроксимации сводится к задаче минимизации суммы

$$\sum_i \rho(r_i(x_i, \theta); \sigma)$$

Введение подобной модификации значительно уменьшает влияние ошибочных точек набора на результат аппроксимации.

Алгоритм метода:

1. Получить начальное приближение набор точек прямой (например, методом наименьших квадратов), получить вектор параметров прямой θ^0
2. Для θ^0 посчитать начальное приближение σ^0
3. Пока вектор параметров существенно изменяется: $\theta^n - \theta^{n-1} > \varepsilon$
4. Аппроксимировать набор точек, используя параметр σ^{n-1}, получить θ^n
5. Пересчитать σ^n

Существует общепринятый способ расчета параметра σ^n, исходя из текущей точности приближения точек прямой:

$$\sigma^n = 1.4826 * \text{median}_i |r(x_i; \theta^{n-1})|$$

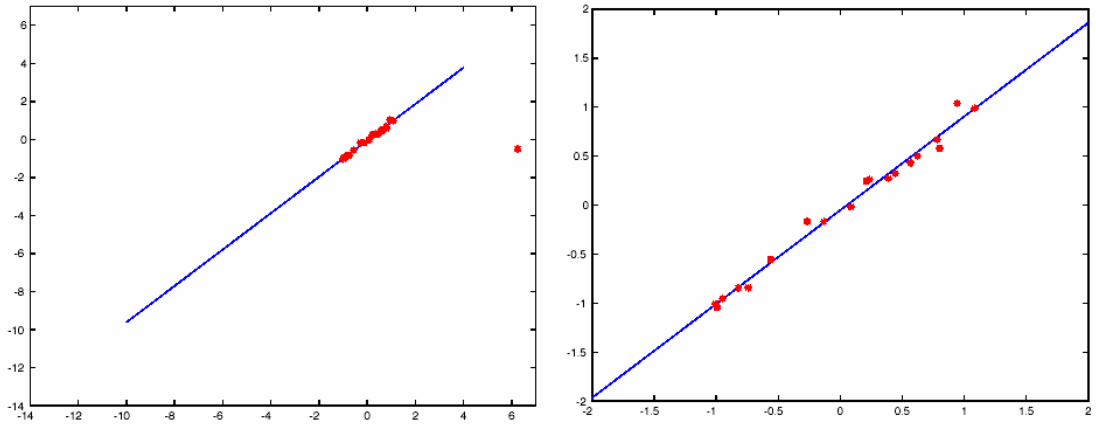

Рис 4 Приближение множества точек с рисунка 3 с помощью модифицированного метода наименьших квадратов.

Более подробно с использованием m-estimators можно познакомиться в статье [2].

Метод RANSAC.

Один из подходов к решению проблемы аппроксимации основан на сборе статистики о входных данных. Из общего набора входных точек случайным образом выбирается некоторое подмножество фиксированного размера, которое аппроксимируется прямой. Общее количество точек входного набора, оказавшихся вблизи полученной прямой, запоминается. Этот процесс повторяется несколько раз. Прямая, вблизи которой оказалось наибольшее число точек с высокой вероятностью является наилучшей аппроксимацией всего множества входных точек. Алгоритм получил название RANSAC – RANdom SAmple Consensus [3].

Чтобы эффективно использовать алгоритм на практике, необходимо ответить на следующие вопросы:

- сколько итераций с выбором подмножеств надо произвести?
- сколько точек должно быть в подмножестве?

В [1] показано, что минимальное количество итераций `k`, обеспечивающее высокую вероятность корректности результата в случае, когда на вход подается `n` точек, среди которых `w` точек заданы корректно, а остальные

являются шумом, есть

$$k = w^{-n} + \frac{\sqrt{1 - w^n}}{w^n}$$

при минимально возможном количестве точек в подмножестве – 2.

Таким образом, при определенных:

- Минимально необходимом количестве точек в подмножестве - n
- Необходимом количестве итераций - `k`
- Пороге расстояния от прямой до точки, при котором точка считается лежащей вблизи прямой - `t`
- Минимальном количество точек, при которых прямая не считается шумом - `d`

алгоритм выглядит так:

1. пока количество итераций не достигло k
2. выбрать подмножество из `n` точек случайным образом
3. аппроксимировать текущее подмножество прямой (например, методом наименьших квадратов)
4. обнулить счетчик близких к прямой точек
5. для каждой точки вне подмножества
6. посчитать расстояние от нее до текущей прямой
7. если расстояние меньше либо равно порогу t
8. инкрементировать счетчик
9. если значение счетчика достигло `d`
10. аппроксимировать текущую прямую по всем близким к ней точкам
11. добавить прямую в набор "хороших" прямых
12. из набора "хороших" прямых выбрать прямую с максимальным счетчиком (количеством точек, близких к данной прямой)

Литература:

[1] А. Дегтярева, В. Вежневец "Преобразование Хафа", *Онлайн журнал "Графика и Мультимедиа",* выпуск 1, http://cgm.graphicon.ru/issue1/hough/index.html.
[2] C.V. Stewart, "Robust Parameter Estimation in Computer Vision", *SIAM REVIEW,* Vol.41,No.3,pp.513 -537, 1999.
[3] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography", *CACM*, 24:381--395, 1981.