

Алгоритм нахождения отрезков в задаче анализа аэрофотоснимков

Л. Новоторцев, А. Волобой
Институт прикладной математики им. М.В. Келдыша, Москва, Россия
torets13@gmail.com, voloboy@gin.keldysh.ru

Аннотация

На данный момент в области геоинформатики и фотометрии большое значение имеют задачи, связанные с обнаружением, обработкой и построением различных объектов на аэрофотоснимках. В частности, это имеет отношение к обнаружению построек, для которых в большинстве случаев характерны линейные черты. По этой причине целесообразно сначала выделить отрезки на изображении. Потом, на основе найденных отрезков, проводить дальнейший анализ. Данная работа посвящена вопросу выделения линейных черт, таких, как отрезки на изображениях для последующего объединения их в связанные контуры.

Ключевые слова: Компьютерное зрение, выделение линий, выделение углов, преобразование Хафа.

ВВЕДЕНИЕ

В задачах компьютерного зрения часто возникают задачи, связанные с реконструкцией всей сцены или отдельных объектов на ней по множеству фотографий этой сцены, сделанных с разных ракурсов. Алгоритмы решения этих задач активно используются в картографии и смежных областях.

Внимание данной статьи направлено на рассмотрение задачи нахождения и реконструкции отдельных объектов по заданному набору шаблонов, представляющих собой трёхмерную каркасную модель.

Существует большое количество алгоритмов, связанных с реконструкцией сцены по набору фотографий, но все они имеют определённые недостатки при применении к описанной выше задаче. Первый подход связан с построением плотной модели ЦМР (Цифровая модель рельефа). Потом ЦМР обрабатывается с целью получить границы строений [3, 5, 7, 8]. Но общий метод недостаточно точен для построения зданий. Поэтому применяются методы, использующие априорную информацию о форме зданий, к примеру, плоские крыши у зданий [2]. Второй подход, который наиболее часто используют для задач реконструкции строений по набору аэрофотоснимков, опирается на выделение линейных участков на изображении (отрезков) и применение подхода перебора плоскостей [1]. Однако этот алгоритм на изображениях большого размера может работать неоправданно долго из-за сопоставления прямых на большом количестве изображений (рекомендуемое количество пересекающихся изображений – 5 – 6). По этой причине возникает задача сужения рассматриваемых областей для уменьшения вычислительных затрат.

Для реализации подобного алгоритма необходимо сначала разработать метод обнаружения прямых линий и отрезков на изображении.

Первым шагом решения задачи нахождения прямых является задача обнаружения контуров. Наиболее распространённым методом решения этой задачи является алгоритм Кэнни [4] нахождения граней. Этот алгоритм решает целый ряд проблем, которые возникают в подобных задачах, таких, как

дублирование контура. Результатом работы этого алгоритма является двоичное изображение, которое имеет нули во всех точках, где контура не обнаружено и ненулевые значения в остальных точках.

Наиболее распространённым и эффективным алгоритмом выделения прямых на изображении является преобразование Хафа [6, 9] и его модификации [10, 11], призванные ускорить расчёты, улучшить результаты или адаптировать его к конкретным постановкам задач. В качестве основы алгоритма, приведённого в этой статье, было решено использовать иерархический алгоритм Хафа, описанный в [11]. Его суть заключается в том, что изображение разбивается на блоки, в каждом из которых применяется преобразование Хафа, после чего прямые, полученные в каждом блоке объединяются, используя иерархический подход. Однако этот алгоритм имеет определённые недостатки. В случае участков с низкой контрастностью часть линии может быть не найдена, в результате чего линия будет сегментирована, что отрицательно может сказаться на последующих вычислениях.

Метод, предоставленный в данной статье, направлен на устранение данного изъяна.

ОБЩАЯ КОНЦЕПЦИЯ

В рассматриваемой статье предлагается использовать регулярное разбиение изображения и последующее нахождение прямых и их концов. Алгоритм можно разделить на три стадии:

1. Определение размера одного блока изображения, исходя из значения допустимой погрешности угла наклона прямых (отрезков).
2. Нахождение прямых линий в каждом блоке.
3. Нахождение концов найденных прямых.

Рассмотрим каждую стадию подробнее.

Определение размера блока

Первая задача, которую необходимо решить – выбрать размер блока. Ввиду дискретности изображения, при слишком малых размерах сегментов, невозможно будет различить прямые с малой разницей углов наклона. Минимальным требованием для возможности различить прямые, имеющие разницу в угле наклона $\Delta\theta$, является несовпадение их конечных пикселей. Иными словами, необходимо, чтобы $\Delta x \geq 1$ или $\Delta y \geq 1$. Ввиду симметричности синуса и косинуса, достаточно рассмотреть область от 0 до 45° . В этом случае $\Delta x \geq \Delta y$; это означает, что достаточно удовлетворить только неравенство $\Delta y \geq 1$. Тогда, учитывая, что

$$(1) y = L \times \sin(\theta),$$

где L – сторона сегмента, получим:

$$(2) 1 \leq \Delta y = y_1 - y_2 = L \times \sin(\theta + \Delta\theta) - L \times \sin(\theta),$$

$$(3) 1 \leq 2 \times L \times (\sin(\Delta\theta/2) \times \cos(\theta + \Delta\theta/2))$$

Учитывая, что $\theta \in [0; \pi/4]$, $-\cos(\theta + \Delta\theta/2) \geq \sqrt{2}/2$. Подставляя это в неравенство (3), получим:

$$(4) L \geq 1 / (\sqrt{2} \times \sin(\Delta\theta/2)).$$

В таблице приведены значения L для различных $\Delta\theta$.

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

Допустимая погрешность и минимальные размеры сегмента для обеспечения этой погрешности

Нахождение прямых линий

На данном этапе применяется непосредственно алгоритм Хафа к каждому блоку. Суть этого алгоритма заключается в том, что для каждой точки сегмента перебираются все углы наклона прямой с заданным шагом. Для каждой точки сегмента, помеченной, как граница, и для каждого предполагаемого угла наклона θ высчитывается длина перпендикуляра ρ , опущенного на предполагаемую прямую из точки начала координат. Потом в ячейку, соответствующую паре (θ, ρ) зачисляется один голос. В результате получается карта голосования. Далее, на основе подобного голосования находятся параметры прямых: те пары (θ, ρ) , для которых количество голосов превышает заданную планку, считаются параметрами найденной прямой.

В рассматриваемом методе предлагается выделять только одну прямую, набравшую наибольшее количество голосов, превышающее заданную планку. Предполагается, что потери на этом этапе компенсируются на третьем этапе при построении отрезков.

Выбор только одной прямой обусловлен тем, что при достаточно малых размерах блоков (в разделе 2.1 полученные размеры можно вполне считать малыми) потеря информации минимальна.

Нахождение концов найденных прямых

Следующим шагом является окончательное нахождение отрезков по найденным прямым. Основная сложность этого этапа заключается в том, что на предыдущем этапе линии могли быть сегментированы в результате низкого контраста участка изображения или в результате того, что линия в сегменте была слишком короткая.

Для решения этой проблемы предлагается использовать следующий подход. Для найденной прямой рассматриваются сегменты, полученные на первом и втором этапах, через которые она проходит. В зависимости от конкретных целей применения рассматриваемого алгоритма и реализации можно или рассматривать все сегменты, или рассматривать последовательно, начиная от того, где прямая была найдена. В каждом сегменте, через который проходит прямая, вычисляются пересечения прямой с его границами. Далее, вычисляется длина отрезка прямой внутри сегмента и длина перпендикуляра к ней в данном сегменте (угол наклона не изменяется):

$$(5) \begin{cases} y_0 = \frac{x_0 \cdot \sin(\theta) + \rho}{\sin(\theta)} \\ x_0 = \frac{\rho}{\sin(\theta)} \end{cases}$$

где y_0 – точка пересечения с левой стороной сегмента или её продолжением; x_0 – точка пересечения с нижней стороной сегмента или её продолжением.

В результате мы получаем параметры θ и ρ , описывающие прямую в данном сегменте.

После этого рассматриваются результаты голосования в данном сегменте. В этом случае достаточно рассмотреть только точки (θ, ρ) на карте голосования или её окрестность. Во втором случае окрестность достаточно взять равной единице по каждому из параметров. Это определяется тем,

что мы и так строили сегменты, для гарантирования необходимой нам точности. На этот раз в качестве планки для определения наличия прямой используется её длина внутри сегмента (возможно, какой-то процент от длины). По аналогии со вторым этапом прямая считается найденной, если количество голосов превышает эту планку.

Если прямая найдена в данном сегменте, то он помечается как сегмент, содержащий эту прямую, и алгоритм повторяется для следующего сегмента, через который проходит прямая, а конец прямой обновляется. В случае если рассматриваемая прямая совпадает с уже найденной в рассматриваемом сегменте, то найденная удаляется или помечается, как уже обработанная. В некоторых случаях бывает нужно также требовать не полное совпадение, а малое отличие двух прямых.

Если прямая не найдена, то в зависимости от допустимой длины разрыва алгоритм продолжается для следующего сегмента или останавливается.

РЕЗУЛЬТАТЫ



Рис. 1. Исходное изображение

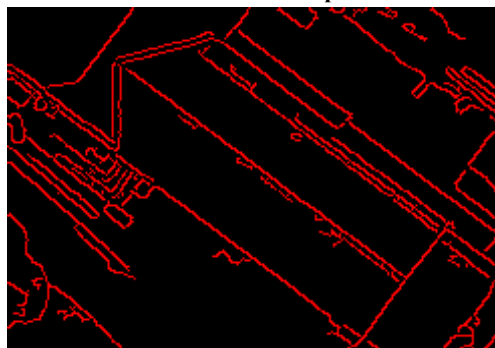


Рис. 2. Результат работы фильтра Кэнни

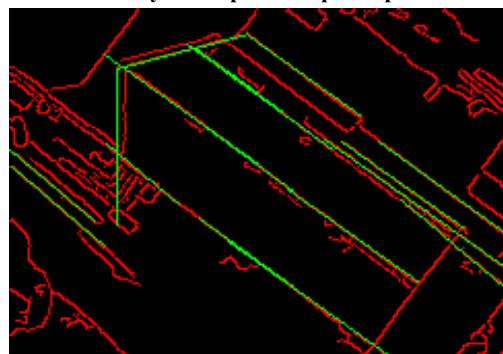


Рис. 3. Результаты, полученные алгоритмом Хафа, реализованным в OpenCV

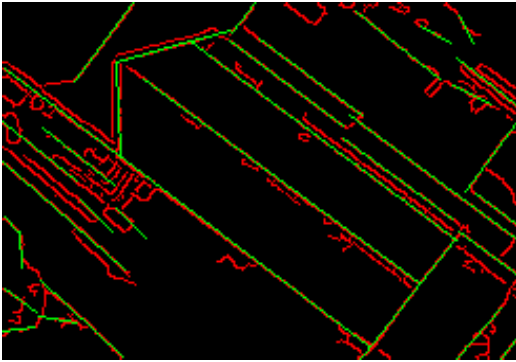


Рис. 4. Результаты, полученные рассматриваемым алгоритмом

На рис. 1 приведено исходное изображение. На рис. 2 представлен результат его обработки алгоритмом Кэнни. Красным обозначены найденные грани.

На рис. 3 приведён результат, полученный алгоритмом Хафа, который реализован в OpenCV. На рис. 4 – результат работы рассматриваемого алгоритма. Красным выделены грани, найденные фильтром Кэнни, зелёным – выделенные отрезки.

Рассматриваемый алгоритм имеет ряд преимуществ и недостатков по сравнению с другими вариантами реализации алгоритма Хафа.

К преимуществам можно отнести то, что показывают результаты (рис. 3, 4). Рассматриваемый алгоритм более точно выделяет отрезки по сравнению с алгоритмом Хафа, предложенным в OpenCV. На рис. 3 найдено гораздо меньше отрезков и присутствует сегментация отрезков, что негативно может сказаться на последующих вычислениях. Рассмотренный в данной статье метод даёт намного более приемлемый результат, что значительно облегчает дальнейшее связывание отрезков.

К недостаткам можно отнести то, что скорость работы, по сравнению с иерархическим алгоритмом Хафа, заметно меньше и то, что погрешность определения концов отрезков пропорциональна размерам сегментов.

ССЫЛКИ

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. – 2000. – Т. 33. – №. B2; PART 2. – С. 56 - 62.
2. Berthod M. et al. High-resolution stereo for the detection of buildings //Automatic Extraction of Man-Made Objects from Aerial and Space Images. – Birkhäuser Basel, 1995. – С. 135 - 144.
3. Brunn A., Weidner U. Extracting buildings from digital surface models //International Archives of Photogrammetry and Remote Sensing. – 1997. – Т. 32. – №. 3 SECT 4W2. – С. 27 - 34.
4. Canny J. A computational approach to edge detection //Pattern Analysis and Machine Intelligence, IEEE Transactions on. – 1986. – №. 6. – С. 679 - 698.
5. Cord M., Paparoditis N., Jordan M. Dense, reliable and depth discontinuity preserving DEM computation from HRV urban stereopairs //International Archives of Photogrammetry and Remote Sensing. – 1997. – Т. 32. – С. 49 - 56.
6. Duda R. O., Hart P. E. Use of the Hough transformation to detect lines and curves in pictures //Communications of the ACM. – 1972. – Т. 15. – №. 1. – С. 11 - 15.
7. Fernandes L. A. F., Oliveira M. M. Real-time line detection through an improved Hough transform voting scheme //Pattern Recognition. – 2008. – Т. 41. – №. 1. – С. 299 - 314.
8. Girard S. et al. Building detection from high-resolution color images //Remote Sensing. – International Society for Optics and Photonics, 1998. – С. 278 - 289.
9. Hough P. V. C. Method and means for recognizing complex patterns : пат. 3069654 США, 1962.
10. Wang Z., Wu F., Hu Z. MSLD: A robust descriptor for line matching //Pattern Recognition. – 2009. – Т. 42. – №. 5. – С. 941 - 953.
11. Yacoub S.B., Jolion J.M. Hierarchical line extraction // IEE Proceedings-Vision, Image and Signal Processing. – 1995. – Т. 142. – №. 1. – С. 7 - 14.