

АЛГОРИТМ ВЫДЕЛЕНИЯ ЛИНЕЙНЫХ ОБЪЕКТОВ НА АЭРОФОТОСНИМКАХ И КОСМИЧЕСКИХ ИЗОБРАЖЕНИЯХ

Б. А. Залесский, Э. Н. Середин

Объединённый институт проблем информатики Национальной академии наук Беларуси

Минск, Республика Беларусь

E-mail: eduard.seredin@tut.by, zalesky@newman.bas-net.by

Предлагается новый подход к решению задачи выделения линейных объектов на аэрофотоснимках и космических изображениях. Описывается алгоритм, разработанный на основе предложенного подхода. Преимуществами алгоритма является возможность его быстрого выполнения за счет распараллеливания, легкая настройка для практической работы, возможность удобной ручной корректировки решения. Описываются особенности последовательных и параллельных версий алгоритма.

ВВЕДЕНИЕ

В настоящее время одной из наиболее трудных является задача сегментации и дешифрирования снимков среднего или низкого пространственного разрешения с большим количеством спектральных каналов, полученных на основе данных дистанционного зондирования земли. Известные алгоритмы автоматического выделения объектов на этих снимках не отличаются высокой точностью, поэтому интерактивные методы с различной долей автоматизации [1-9], по-прежнему остаются более предпочтительными.

I. КРАТКОЕ ОПИСАНИЕ АЛГОРИТМА

Предлагается алгоритм интерактивного выделения линейных объектов на аэрофотоснимках и космических изображениях среднего и низкого пространственного разрешения [10]. Входными параметрами алгоритма являются: начальный и конечный пиксели участка линейного объекта; число узлов ломаной; допустимый угол между соседними звеньями ломаной, ограничивающий кривизну решения; ширина в пикселях растрового представления ломаной; тип градиентного оператора. Алгоритм состоит из следующих шагов:

Шаг 1. Пользователь задает исходные данные: начальный a и конечный b пиксели участка линейного объекта на аэрофотоснимке или космическом изображении, который будет приближен ломаной линией; число звеньев ломаной; число точек на каждой направляющей; максимально разрешенный угол излома ломаной, ограничивающий допустимую кривизну решения (рис. 1).

Шаг 2. Строится отрезок прямой $[a,b]$, который разбивается на заранее заданное пользователем количество одинаковых отрезков.

Шаг 3. Выбирается один из известных операторов для построения градиента исходного изображения, например, оператор Превитта, Собеля и т.д. или желаемый оператор задается пользователем в зависимости от характеристик обрабатываемого изображения.

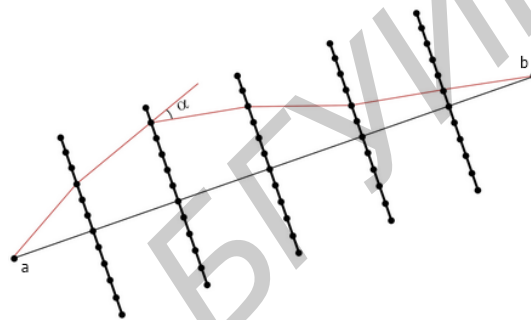


Рис. 1 – Пример расчета частичных сумм с учетом угла между отрезками

Шаг 4. Вычисляется градиент исходного изображения.

Шаг 5. В пикселях разбиения отрезка $[a,b]$, строятся направляющие ломаных – отрезки прямых линий перпендикулярные отрезку $[a,b]$. Каждая направляющая разбивается на интервалы одинаковой длины, число которых задано на Шаге 1.

Шаг 6. Рассматриваются всевозможные растровые ломаные с началом в пикселе a , концом в пикселе b и узлами в концах интервалов разбиения направляющих. Для каждой растровой ломаной вычисляется сумма модулей синусов углов между градиентом попавшем на ее звенья и отрезком ломаной.

Шаг 7. Находится максимальная сумма из всех сумм, вычисленных на Шаге 6 при полном переборе всех ломаных с учетом задаваемого пользователем максимального допустимого угла между отрезками двух соседних направляющих.

Шаг 8. Найденная ломаная линия, которая принимается за приближение участка линейного объекта, отображается на фоне исходного изображения.

Шаг 9. В случае необходимости пользователь может скорректировать форму ломаной путем изменения одного или нескольких ее узлов.

Шаг 10. Полученное решение принимается или отменяется пользователем. Если результат принят, его можно сохранить и продолжить работу алгоритма. В случае продолжения, пиксел

b становится начальным для нового участка линейного объекта. Далее выбирается новый пиксел – конец нового участка и повторяются шаги 1-9.

II. ОЦЕНКА БЫСТРОДЕЙСТВИЯ ПОСЛЕДОВАТЕЛЬНОЙ РЕАЛИЗАЦИИ АЛГОРИТМА

При дешифрировании изображений программная реализация алгоритма запускается многократно, поэтому одним из основных требований к ней является возможность выполнения алгоритма в режиме реального времени либо не более чем с односекундной задержкой. Тестирование неоптимизированной версии разработанного алгоритма, основанной на полном переборе всех ломаных, показало, что большая часть времени вычисления приходится на выбор наилучшей ломаной из всех построенных. Очевидно, что количество операций N при полном переборе всех вариантов зависит от числа всех возможных ломаных. Если m – число точек на каждой направляющей, k – их количество, то $N = O(m^k)$. Для ускорения расчетов было необходимо, во-первых, переработать алгоритм, уменьшив количество операций, необходимых для поиска оптимального решения, и во-вторых, максимально распараллелить его программную реализацию, задействовав видеокарту, которая обеспечивает одновременное выполнение огромного числа операций. Если не учитывать ограничение на кривизну ломаной, методы динамического программирования позволяют найти наилучшее приближение за $N \leq m^2(k-1) + m$ операций. Однако для выделения линейных объектов с помощью ломаных предпочтительно использовать ограничение на их кривизну, так как это позволяет значимо повысить точность выделения на изображениях невысокого разрешения даже при наличии шумов и искажений. Специально для такого случая была разработана версия метода динамического программирования, позволяющая находить приближения ломаными заданной кривизны за $N \leq m^3(k-2) + m^2$ операций.

III. ПАРАЛЛЕЛЬНАЯ ВЕРСИЯ АЛГОРИТМА ПОИСКА НАИЛУЧШЕГО РЕШЕНИЯ

Параллельная версия алгоритма специально разработана для большого количества одновременно работающих потоков с учетом трех основных требований: сократить время выполнения и общее количество операций, а также уменьшить объем памяти, используемой на промежуточных шагах работы алгоритма. Эффективное распараллеливание алгоритма возможно в случае, когда функция-критерий алгоритма, на основе которой строится ломаная, приближающая линейный объект, является аддитивной

функцией относительно звеньев ломаной. Параллельная версия алгоритма реализована с использованием технологии программирования CUDA. Полученные данные тестирования показывают, что CUDA-версия параллельного алгоритма выполняет расчеты за приемлемое время (доли секунды) в широком диапазоне значений входных параметров.

ЗАКЛЮЧЕНИЕ

К преимуществам разработанного алгоритма можно отнести его быстрое выполнение (в течение долей секунды), легкая настройка для практической работы, возможность удобной ручной корректировки решения. Параллельная версия алгоритма основана на разработанной версии метода динамического программирования. Реализация параллельной версии алгоритма с использованием технологии программирования CUDA позволяют уменьшить на несколько порядков объем требуемых вычислений и сократить время выполнения работы алгоритма.

1. Supreet, S. Automatic Road Detection of Satellite Images – A Survey / S. Supreet, B. Seema // Intern. J. of Computer Applications and Information Technology. – 2013. – Vol. 3(2). – P. 32-34.
2. Kalaivanan, R. Survey on Road Extraction From High Resolution Satellite Images / R. Kalaivanan, S. Mishmala // Intern. J. of Advanced Research in Computer and Communication Engineering. – 2013. – Vol. 2(10). – P. 4156-4159.
3. A Family of Quadratic Snakes for Road Extraction / M.N. Dailey [et al.] // Lecture Notes in Computer Science. – 2007. – Vol. 4843. – P. 85-94.
4. Dal Poz, A.P. Dynamic Programming Approach For Semi-Automated Road Extraction From Medium- And High-Resolution Images / A.P. Dal Poz, G.M. do Vale // ISPRS Archives. – Vol. 34 (3/W8). – P. 87-91.
5. Urban digital map updating from satellite high resolution images using GIS data as a priori knowledge / T. Bailloeuil [et al.] // Remote Sensing and Data Fusion over Urban Areas, 2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas. – Urban, 2003. – P. 283-287.
6. Niu, X. A semi-automatic framework for highway extraction and vehicle detection based on a geometric deformable model / X. Niu // Photogrammetric Engineering and Remote Sensing. – 2006. – Vol. 61. – P. 170-186.
7. Automatic Road Extraction from Satellite Imagery Using LEGION Networks / J. Yuan [et al.] // Proc. of Intern. Joint Conference on Neural Networks. – Atlanta, Georgia, USA, 2009. – P. 3471-3476.
8. Lacoste, C. Unsupervised line network extraction in remote sensing using a polyline process / C. Lacoste, X. Descombes, J. Zerubia // Pattern Recognition. – 2010. – Vol. 43 (4). – P. 1631-1641.
9. Color image segmentation: advances and prospects / H. Cheng [et al.] // Pattern Recognition. – 2001. – № 34. – P. 2259-2281.
10. Середин, Э.Н. Алгоритм интерактивного выделения линейных объектов на аэрофотоснимках и космических изображениях / Э.Н. Середин, Б.А. Залесский // Информатика. – 2014. – №4. – С. 66-73.