

УДК 004.932.2

**АЛГОРИТМ ВЫЧИСЛЕНИЯ ПОРОГОВЫХ ЗНАЧЕНИЙ ДЛЯ ПОВЫШЕНИЯ
АВТОМАТИЗМА СИСТЕМ РАСПОЗНАВАНИЯ ГРАФИЧЕСКИХ ОБРАЗОВ**

Ю.А. Гатчин, С.В. Москаленко

Представлена система идентификации линейных объектов на инженерных чертежах. По завершении работы данная система получает набор данных, необходимый для определения различных пороговых значений, повышающих автоматизм алгоритмов анализа изображения.

Ключевые слова: сегментация, линейный объект, пороговое значение, распознавание образов.

Введение

Сегментация – одна из ключевых открытых проблем в компьютерном зрении. До недавнего времени распознавание и сегментацию не связывали вместе. Однако в области сегментированного представления мало тем, изучение которых не помогло бы решить задачи, связанные с распознаванием [1].

На вход большинства систем распознавания графических изображений подается некоторый избыточный набор неструктурированных данных. После этого система должна отсеять лишнюю информацию и выделить множества с интересующими характеристиками. Данный процесс получения компактного представления полезного содержания объекта называется сегментацией или идентификацией. Верное выявление важной информации на изображениях обеспечивает надежное функционирование систем распознавания образов, а также является базисом для систем восстановления изображений. На практике проблема сегментации изображений является довольно трудноразрешимой. Для обеспечения корректного функционирования процесса сегментации необходимо вовлечение некоторых пороговых значений. В данной работе предложен алгоритм, вычисляющий основные пороги в автоматическом режиме.

Теоретическое обоснование

Как правило, многие системы распознавания графических изображений включают процедуру сегментации в этап первичной векторизации, во время которого происходит конвертирование пиксельного набора изображения в векторное представление [2]. В контексте этапа векторизации сегментация занимается не только поиском информативных графических примитивов, но и организацией совокупностей данных примитивов в простейшие группы. В частности, множество примитивов включает: точки, векторы, полигоны, объединение которых может быть определено как текст или графика, площадной объект или линейный. Процедура сегментации в таких системах часто нуждается в определении специальных пороговых значений, которые задаются в диалоговом режиме оператором [3–5]. Так, применимо к задаче анализа инженерных чертежей почти всегда требуется, чтобы пользователь задал следующие пороги: максимальная ширина линии (критерий принадлежности объекта классу площадных или линейных), максимальная длина луча, определяющего ширину объекта, расстояние между штрихами для пунктирных линий. Основываясь на данных порогах, система строит вывод о типе конкретного линейного объекта: основная или тонкая линия, штрихпунктирная линия или оборванная (вследствие помех на растре).

В большинстве случаев данные пороговые значения определяются эмпирическим путем, что лишает систему автоматизма. Поэтому разработка автоматизированных методов вычисления порогов остается актуальной.

Реализация алгоритма

Разработанный алгоритм сегментации должен отделять линейные объекты от площадных и дискретных объектов. Примерно 90% объектов на инженерных чертежах составляют линии. Исходя из [6], все эти линии могут быть разделены на 2 группы по признаку ширины: основные (сплошная толстая основная линия), тонкие (сплошная тонкая, сплошная волнистая, штрихпунктирная тонкая линия). Суть работы алгоритма заключена в предварительной идентификации всех линейных объектов на изображении. Изначально алгоритм не будет знать, к какой группе отнести обнаруженный линейный объект: данный объект может принадлежать либо к группе основных линий, либо к группе тонких, либо к группе площадных объектов. Здесь даже площадной объект может обладать признаками линейного, когда отношение длины объекта к ширине больше 4, что существенно затрудняет задачу классификации. Замеряя ширину найденных линий, алгоритм формирует выборку, исходя из которой вычисляет среднее арифметическое значение ширины линии для каждой группы. Полученная информация о ширине линейных объектов поможет на следующих

этапах анализа изображения отделять линейные объекты от площадных объектов, определять пороговые значения в автоматическом режиме и вычислять точки центральной линии.

Работа алгоритма начинается с введения в систему бинарного растрового чертежа. Далее происходит наложение сетки на введенное изображение и обнаружение точек (пикселей), лежащих на пересечении графических примитивов со штрихами сетки. Данный процесс представляет собой сначала горизонтальное, а потом вертикальное сканирование изображения на наличие черных точек, говорящих о встрече некоторого примитива. Что касается выбора интервала между параллельными штрихами сканирующих линий, то здесь можно отталкиваться от разрешения изображения. Например, опытная эксплуатация показала, что при размере изображения 1600×1200 пикселей и интервале между штрихами в 10 пикселей алгоритм будет обладать оптимальным соотношением скорости работы к точности вычисления ширины линейных объектов.

После обнаружения некоторого объекта от полученной исходной точки (i, j) в основных направлениях строится циклическая гистограмма, отражающая ширину объекта в каждом из направлений. Здесь ширина объекта характеризуется длиной пробега в одном направлении по черным пикселям до первого белого. Направления измерения ширины можно последовательно строить по часовой стрелке с приращением угла α , начиная от луча, совпадающего со сканирующей линией, обнаружившей данный объект. В связи с недостатком высоким разрешением изображения и с ограниченным числом лучей циклической гистограммы длина участка пересечения лучом графического объекта может отличаться от реальной толщины объекта в данной точке. Ошибка вычисления толщины для такого случая определяется по формуле (1):

$$\Delta = \frac{1}{\cos(\alpha/2)} - 1, \tag{1}$$

где α – угол между смежными лучами. Число лучей гистограммы может быть взято равным 18, так как при $\alpha = 10^\circ$ ошибка вычисления ширины объекта не превысит 1,5%.

Для построения циклической гистограммы ширина объекта в каждом из направлений рассчитывается типовым алгоритмом Брезенхема [7]. В результате могут получиться гистограммы с пиками и впадинами. Гистограммы в большинстве случаев будут иметь гладкий непрерывный вид (рис. 1, а), однако в некоторых случаях, когда исходный растр достаточно зашумлен и имеет «дыры» в объектах, график будет терять свою гладкость (рис. 1, б). Для повышения точности определения типа объекта и его ширины необходимо для зашумленных изображений сглаживать гистограммы. Это можно сделать путем вычисления в каждой точке графика усредненного значения, принимая во внимание значения соседних точек.

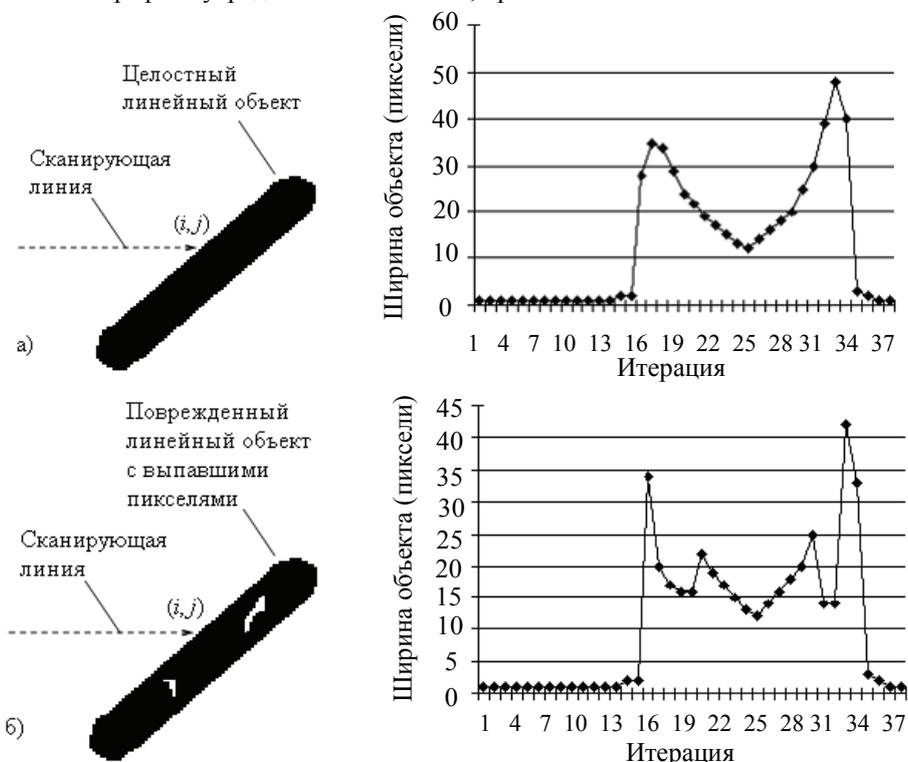


Рис. 1. Примеры гистограмм линейных объектов (слева линейный объект, справа график гистограммы): а) гладкая; б) возмущенная

Под пиком понимается точка экстремума на графике, при которой функция на прилегающем участке слева возрастает, а справа убывает. Под впадиной понимается точка экстремума на графике, при которой функция на прилегающем участке слева убывает, а справа возрастает. Будем считать любой

объект, одно из измерений которого больше измерения другого в 4 раза, линейным (длина в 4 раза больше ширины). Следовательно, для линейного объекта на графике должно существовать два пика с высотами минимум в два раза большими, чем величина впадины между ними (рис. 2). Далее из данного положения будет сформулирован критерий идентификации линейных объектов.

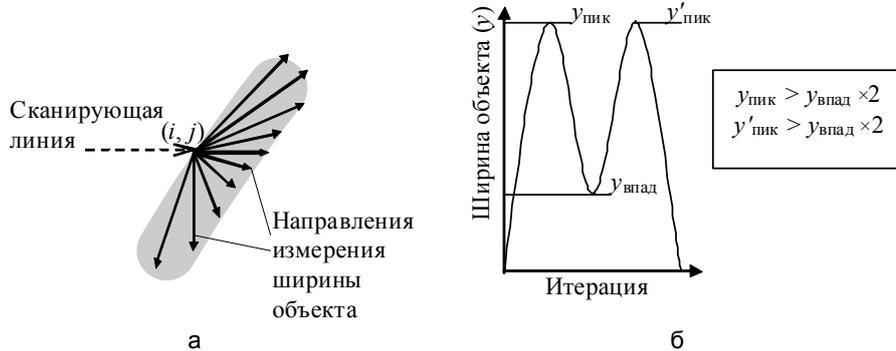


Рис. 2. Представления линейного объекта: вид на растре (а); усредненная гистограммная функция ширин объекта (б)

Получив гистограммную функцию, следует минимизировать ее до схемы экстремумов с формированием массива экстремальных точек. Данная мера является чисто оптимизационным моментом, так как для анализа графического объекта достаточно лишь информации об экстремумах. Массив экстремальных точек будет состоять из повторяющихся пар экстремумов типа «пик–впадина». Отрезок, соединяющий пару последовательных экстремумов типа «впадина–пик», будем называть возрастающим, а отрезок, соединяющий пару последовательных экстремумов типа «пик–впадина» – убывающим.

Далее находим в массиве наибольший экстремум с $y_{\text{пик}(\text{max})}$, являющийся самым высоким пиком (рис. 3, а). Отталкиваясь от $y_{\text{пик}(\text{max})}$, можно однозначно заключить, что если за данным пиком существует впадина с $y_{\text{впад}} < y_{\text{пик}(\text{max})} / 2$, а за впадиной существует еще один пик с $y_{\text{пик}} > y_{\text{впад}} \times 2$, то рассматриваемый объект – линия. Для проверки данных условий надо исследовать, с какими возрастающими и убывающими отрезками на схеме экстремумов пересекается линия $f = y_{\text{пик}(\text{max})} / 2$ (рис. 3, а). Если прямая f пересечет 4 отрезка, значит, идентифицируемый объект – линейный, и теперь остается только проверить график между этими двумя пиками на непрерывность и найти на данном интервале минимальную впадину с $y_{\text{впад}(\text{min})}$. Данная величина $y_{\text{впад}(\text{min})}$ будет являться искомой шириной линии. Будем называть явным пиком ситуацию, когда линия f пересекает возрастающий, а затем убывающий отрезок на схеме экстремумов. Следовательно, критерий идентификации линейного объекта на растре будет выглядеть следующим образом: если на схеме экстремумов, принадлежащей точке (i, j) рассматриваемого объекта, обнаружено два явных пика, значит, объект – линейный.

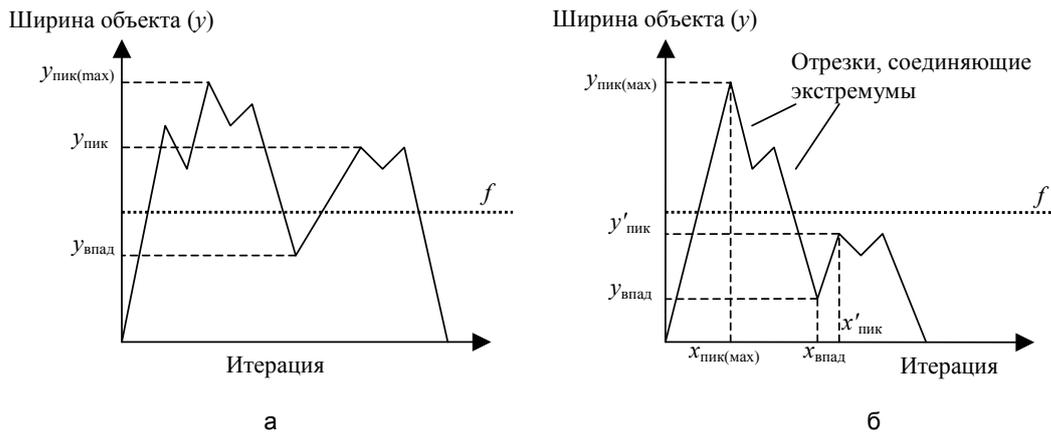


Рис. 3. Схемы экстремумов для линейных объектов: обнаружено два явных пика (а); обнаружен один явный пик (б)

Действительно, опыт показал, что для большинства обнаруженных на растре объектов данный критерий будет справедлив. Однако есть вероятность, что при построении прямой f будет найден лишь один явный пик, хотя идентифицируемый объект все же будет линейным (рис. 3, б). Для таких случаев существует ответвление алгоритма. Происходит запуск перебора массива экстремумов вначале вперед от текущего элемента $x_{\text{пик}(\text{max})}$, равного $\phi(y_{\text{пик}(\text{max})})$. Во время этого перебора происходит проверка: если для очередного пика $x'_{\text{пик}}$, равного $\phi(y'_{\text{пик}})$, существует уже рассмотренный на данном направлении экстремум

с $y_{\text{впад}}$ таким, что $y'_{\text{пик}} > 2 \times y_{\text{впад}}$, значит, идентифицирован линейный объект, и далее идет поиск минимальной впадины $x_{\text{впад}(\min)}$ между двумя пиками $x_{\text{пик}(\max)}$ и $x'_{\text{пик}}$, выход из текущей итерации попиксельного сканирования и запуск следующей итерации. Если перебор массива в данном направлении не дает повода думать, что рассматриваемый объект –линейный, тогда происходит перебор массива в другом направлении, обратном от $x_{\text{пик}(\max)}$.

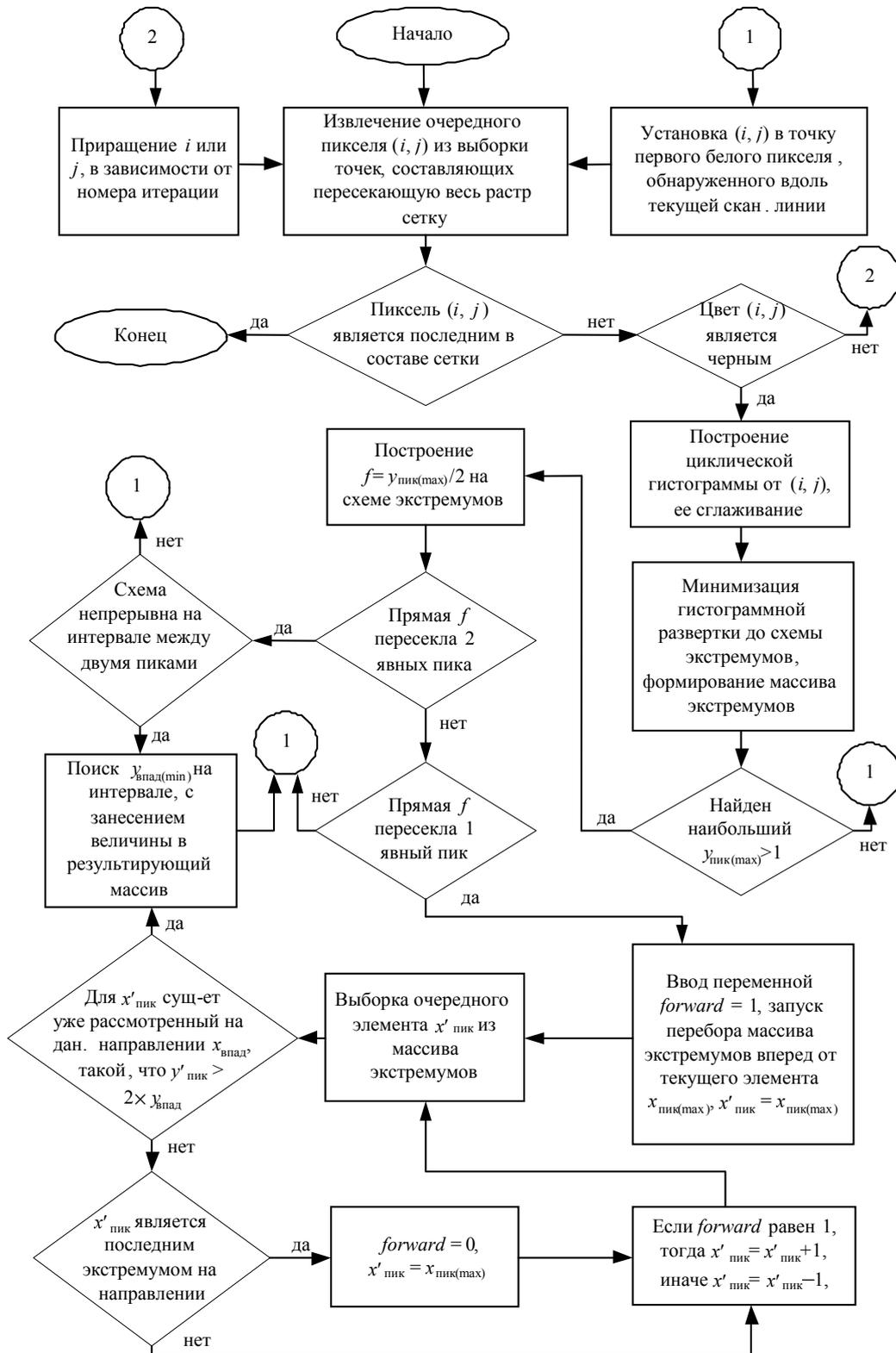


Рис. 4. Схема алгоритма идентификации линейных объектов на растровом изображении

На рис. 4 представлена блок-схема, изображающая обобщенный алгоритм работы системы идентификации линейных объектов на растре с вычислением их ширин.

Таким образом, по окончании операции поиска линейных объектов алгоритм будет обладать статистической выборкой, содержащей значения ширин всех найденных линий. Как было сказано, все линейные объекты можно грубо отнести к группе основных или тонких линий, значит, каждая выборка должна иметь две преобладающие величины, где меньшая величина соответствует усредненной ширине тонкой линии, а большая – ширине основной линии.

Апробация

Реализация алгоритма была выполнена в среде разработки Borland Delphi 7. Так как данный алгоритм должен применяться в системах в качестве дополнительного уровня предобработки, принципиальным остается вопрос о скорости его работы. Большое внимание было уделено его оптимизации. В частности, реализован разреженный просмотр растра: наложение сетки на изображение вместо попиксельного сканирования, ограниченное число направлений измерения ширины объекта для построения гистограммы.

Лучший результат алгоритм показал при анализе отсканированных чертежей формата А4 с 600 dpi (разрешение 3350 на 5694 пикселей). На компьютере средней конфигурации Celeron 1,7 Ghz с 512 Mb оперативной памяти усредненный расчет каждого из данных чертежей проходил около 1 с вместе с точным выявлением ширин линий. Также в результате апробации выявлено, что для обеспечения необходимой точности работы алгоритма требуется на вход подавать чертежи со сканированным разрешением от 600 dpi и средней шириной тонких линий от 5 пикселей.

На рис. 1 представлен пример работы реализации алгоритма, где изображены гистограммы, построенные для линейных объектов растровых изображений разных степеней зашумленности. Если реализовать данный алгоритм на более низкоуровневом языке программирования, то скорость работы увеличится в несколько раз.

Заключение

Разработанный алгоритм, предназначенный для анализа графических данных, дает на выходе две важные величины: ширина основной и ширина тонкой линий. Далее, отталкиваясь от этих двух значений, можно вывести основные пороговые значения, которые задаются в современных системах распознавания образов и обработки изображений человеком. Таким образом, данный алгоритм автоматизирует работу систем обработки изображений.

Точное и быстрое определение пороговых значений является важным этапом в системах обработки изображений. От этих значений будут зависеть процедуры анализа изображения более высокого уровня, такие как распознавание графических образов в САПР. Разработанный алгоритм обладает всеми необходимыми характеристиками для успешной интеграции в САПР.

Литература

1. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход: Пер. с англ. – М.: Вильямс, 2004. – 928 с.
2. Москаленко С.В., Гатчин Ю.А. Волновой алгоритм векторизации линейных растровых изображений // Научно-технический вестник СПбГУ ИТМО. – 2008. – № 51.
3. Feng Su, Jiqiang Song, Shijie Cai. A Vectorization System for Architecture Engineering Drawings // GREC. – 2005. – P. 11–22.
4. Tombre Karl. Analysis of Engineering Drawings: State of the Art and Challenges, Selected Papers from the Second International Workshop on Graphics Recognition, Algorithms and Systems. – 1997, August 22–23. – P. 257–264.
5. Dori D., Liu W. and M. Peleg. How to Win a Dashed Line Detection Contest // In: Graphics Recognition: Methods and Applications, Lecture Notes in Computer Science, Springer, 1996. – V. 1072.
6. ГОСТ 2. 303-68. Линии. – Введ.01.01.71. – М.: Изд-во стандартов, 1968. – 6 с.
7. Bresenham J.E. Algorithym for Computer Control and Digital Plotter // IBM Syst. J. – 1965, April. – V. 4. – № 1. – P. 25–30.

Гатчин Юрий Арменакович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, доктор технических наук, профессор, gatchin@mail.ifmo.ru

Москаленко Станислав Владимирович

– Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, stan.moskalenko@gmail.com