

О поиске прямоугольников на векторной модели изображения с помощью преобразования Хоха

Алексей В. Губанов, Олег Е. Галкин, Николай В. Ильинов.

Нижегородский филиал фирмы "TESIS-Software BVBA"

Нижний Новгород, Россия

Аннотация

Работа посвящена описанию возможных постановок задачи и алгоритма поиска прямоугольников на векторной модели изображения с помощью одного из вариантов преобразования Хоха.

Распознавание прямоугольников может потребоваться, например, при автоматическом чтении чертежей, так как в них широко используются графические конструкции типа прямоугольников, при отыскании информационной области на рентгеновских снимках, при переводе карт или планов местности в электронную форму и в других областях машинного зрения.

Приведенный вариант преобразования Хоха [1-3] сопоставляет отрезкам контуров некоторые подмножества пятимерного пространства параметров, определяющих прямоугольники. Основанный на нем алгоритм поиска прямоугольников позволяет учитывать погрешность, с которой построены элементы векторной модели изображения, и дает возможность производить поиск прямоугольников, часть границ которых отсутствует или искажена. В процессе работы алгоритм определяет области, в которых могут находиться параметры искомого прямоугольника.

В работе приводятся результаты применения алгоритма в реальной системе, где контуры являются кусочно-линейной аппроксимацией границ присутствующих на изображении прямоугольников (границы могут быть искажены шумом и дискретизацией), и построены с использованием библиотеки функций векторной обработки изображений [4].

Описываются планы развития алгоритма.

Ключевые слова: Поиск прямоугольников, Векторная модель изображения, Преобразование Хоха.

1. ВВЕДЕНИЕ

1.1 Цель и содержание работы

В работе описаны возможные постановки задачи и два алгоритма поиска прямоугольников на векторной модели изображения с помощью преобразования Хоха.

Распознавание прямоугольников полезно, например, при автоматическом чтении чертежей (см. [5], [6]), при поиске прямоугольной информационной области на рентгеновских снимках, при распознавании карт и планов местности (см. [6-8] и библиографию к [6]).

Приведенные варианты преобразования Хоха (о преобразовании Хоха см., например, [1-3]) сопоставляют отрезкам контуров некоторые подмножества пятимерного пространства параметров, определяющих прямоугольники. Описаны два алгоритма поиска прямоугольников, основанных на этом преобразовании. Они позволяют учитывать погрешность, с которой построены элементы векторной модели изображения, и дают возможность производить поиск прямоугольников, часть границ которых отсутствует или искажена. В процессе работы алгоритмы находят области, которым могут принадлежать параметры искомого прямоугольника. Возможно использование информации об ограничениях на местоположение, размеры или форму искомого прямоугольника.

В работе приводятся результаты применения одного из алгоритмов на векторной модели полутонового изображения, состоящей из контуров, являющихся кусочно-линейной аппроксимацией границ присутствующих на изображении объектов. Для построения контуров использовалась библиотека функций векторной обработки изображений (см. [4]).

Описываются также планы развития приведенных алгоритмов.

1.2 Векторная модель изображения

Векторной моделью изображения или *векторизованным изображением* (см. [9]) будем называть набор плоских замкнутых контуров K_i , $i = 1, \dots, N$, каждый из которых задан последовательностью координат своих узлов: $K_i = \{ (x_j^{(i)}, y_j^{(i)}) \mid j=1, \dots, N_i \}$. Мы предполагаем, что векторизованное изображение каким-либо способом получено из растрового, например, путем бинаризации растрового изображения с постоянным или адаптивным порогом или методом многоуровневой бинаризации, и последующего построения и сглаживания контуров объектов (описание соответствующих алгоритмов можно найти, в частности, в [4] и [9]).

1.3 «Классическое» преобразование Хоха

Преобразование Хоха было описано им в [1]. Его можно задать следующим образом.

Пусть на плоскости $\mathbf{R}^2 = \{ z = (x, y) \mid x, y \in \mathbf{R} \}$ задано некоторое семейство кривых Γ_p , зависящих от параметра p , изменяющегося в некоторой области P пространства параметров $\mathbf{R}^d = \{ p = (p_1, \dots, p_d) \mid p_i \in \mathbf{R}, i = 1, \dots, d \}$. Предположим, что каждая кривая описывается неявным уравнением $F(z; p) = 0$, то есть $F(x, y; p_1, \dots, p_d) = 0$, $p \in P$. Преобразование Хоха сопоставляет каждой точке $z \in \mathbf{R}^2$

множество P_z параметров, соответствующих кривым Γ_p , проходящим через точку z . Таким образом, мы получаем в пространстве параметров семейство множеств $P_z = \{p \in P \mid F(z, p) = 0\}$, $z \in \mathbf{R}^2$. Отождествляя точки p области P с отвечающими им кривыми Γ_p , можно сказать, что каждой точке $z \in \mathbf{R}^2$ сопоставляется множество кривых Γ_p , проходящих через нее.

1.4 Применение преобразования Хоха к поиску кривых из заданного семейства на черно-белом растровом изображении

Этому вопросу посвящено большое число работ (см., например, [3], [10] и имеющиеся в [3] ссылки). Применить преобразование Хоха к решению указанной задачи можно следующим образом.

Пусть черно-белое изображение состоит из точек z_j , $j = 1, \dots, M$. Сначала для каждой точки z_j находим в пространстве параметров множество P_{z_j} , являющееся ее преобразованием Хоха. Затем ищем те точки $p \in P$, вблизи которых расположено наибольшее число полученных множеств (понятие близости можно вводить по-разному в зависимости от конкретной задачи). Присутствующими на изображении считаются кривые Γ_p исходного семейства, соответствующие этим точкам пространства параметров.

2. ВОЗМОЖНЫЕ ПОСТАНОВКИ ЗАДАЧИ

Нашей целью было разработать вариант преобразования Хоха, переводящий элементы векторной модели изображения во множества прямоугольников, и построить на основе этого преобразования алгоритм поиска прямоугольников на векторизованных изображениях.

Вопрос о поиске прямоугольников может ставиться по-разному в зависимости от имеющейся предварительной информации о прямоугольниках и от того, какие именно семейства прямоугольников мы ищем. Можно предложить целую серию задач поиска, которые позволяют также комбинировать их друг с другом. Например:

1. Найти прямоугольники

- имеющие угол наклона, периметр или площадь в заданных диапазонах;
- с заданными минимальными и максимальными длинами сторон;
- отношение сторон которых лежит в заданных пределах;
- расположенные в заданной области изображения;
- на сторонах которых лежит не менее заданного числа отрезков контуров;
- близкие к нескольким контурам, количество которых не меньше заданного;
- на которых лежит не менее заданного числа вершин контуров;
- у которых не меньше заданного число их вершин, совпадающих с вершинами контуров.

2. Найти максимальный набор прямоугольников

- имеющих общий для всего набора угол наклона, периметр или площадь;
- у которых длина одной (или обеих) сторон — общая для всего набора;
- у которых отношение сторон является общим для всего набора;
- центры или стороны которых лежат на общей для всего набора прямой.

В данной работе речь будет идти, в частности, об алгоритме, поддерживающем запросы 1a)-1e) из перечисленных выше. При этом мы будем предполагать, что на трех сторонах искомого прямоугольника лежит хотя бы по одному отрезку контуров. Допускается, чтобы отрезки не точно лежали на сторонах, а были близки к ним в некотором смысле.

3. ОСНОВНЫЕ РЕЗУЛЬТАТЫ

Ниже описывается преобразование Хоха для отрезков контуров векторизованных изображений и основанные на нем алгоритмы поиска прямоугольников, не менее чем к трем сторонам которых близки какие-либо отрезки.

3.1 Описание пространства параметров

3.1.1 S-параметр прямой

Каждой прямой L сопоставим величину S , равную расстоянию от начала координат O до этой прямой, взятому со знаком плюс, если точка O лежит ниже прямой L (или правее, если прямая вертикальна), и со знаком минус, если точка O лежит выше прямой (или левее, если прямая вертикальна). Будем называть эту величину *S-параметром* прямой.

3.1.2 S-параметр и T-параметры отрезка

S-параметром отрезка J будем называть *S-параметр* прямой, проходящей через этот отрезок.

T-параметрами отрезка J будем называть величины T_1 и T_2 , равные *S-параметрам* прямых, проходящих через концы этого отрезка перпендикулярно ему (см. Рис. 1).

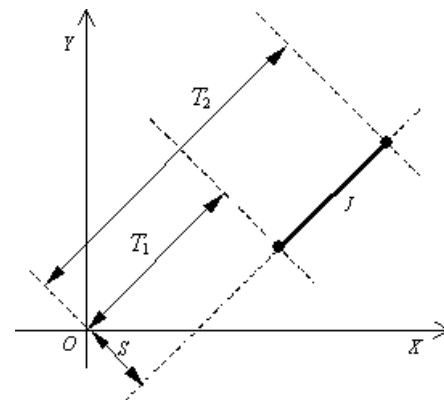


Рис. 1

3.1.3 Описание параметров прямоугольника

Каждый прямоугольник будем задавать набором $p = (\varphi, S_1, S_2, T_1, T_2)$ из пяти параметров, где (см. Рис. 2):

φ — угол наклона двух из четырех сторон прямоугольника, $0 \leq \varphi < 90^\circ$;

S_1, S_2 — S -параметры сторон прямоугольника с углом наклона $\varphi, S_1 \leq S_2$;

T_1, T_2 — T -параметры сторон прямоугольника с углом наклона φ (они равны S -параметрам сторон прямоугольника с углом наклона $\varphi + 90^\circ$), $T_1 \leq T_2$.

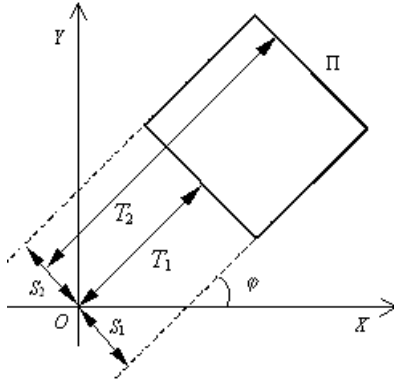


Рис. 2

3.1.4 Различные способы параметризации прямоугольников

В этом пункте мы обсудим различные способы параметризации прямоугольников и объясним, почему мы выбрали именно способ, описанный выше.

В качестве параметров прямоугольника можно брать, например, следующие его характеристики:

1. φ — угол наклона прямоугольника (см. п. 3.1.3);
2. (S_1, S_2, T_3, T_4) — расстояния от начала координат до прямых, проходящих через стороны прямоугольника (см. п. 3.1.3);
3. (X_C, Y_C) — координаты центра прямоугольника;
4. $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), (X_4, Y_4)$ — координаты вершин прямоугольника, в порядке их обхода по часовой стрелке;
5. A, B — длины сторон прямоугольника, $A \geq B$;
6. $Q = B/A$ — отношение длин сторон;
7. D — длина диагонали прямоугольника;
8. ψ — угол между диагоналями прямоугольника.

Описать прямоугольник можно, в частности, с помощью следующих наборов из пяти параметров:

1. $\varphi, S_1, S_2, T_1, T_2$;
2. φ, X_C, Y_C, A, B ;
3. $\varphi, X_1, Y_1, X_3, Y_3$;
4. X_C, Y_C, X_1, Y_1, A или B ;
5. X_1, Y_1, X_3, Y_3, A или B ;
6. φ, X_1, Y_1, A, B ;
7. φ, X_1, Y_1, A, Q ;
8. $\varphi, X_C, Y_C, D, \psi$.

Если $A \neq B$, то наборы 2, 4, 5 и 8 соответствуют двум прямоугольникам, а наборы 6 и 7 — восьми прямоугольникам.

Нас привлек первый из приведенных выше способов параметризации, потому что при этом каждый набор задает один прямоугольник, и нам удобнее описывать преобразование Хоха. Однако этот способ имеет и недостаток: при повороте прямоугольника вокруг своего центра на небольшой угол величины параметров S_1, S_2, T_1 и T_2 могут значительно изменяться.

3.2 Описание преобразования Хоха для отрезков контуров векторизованных изображений.

3.2.1 Предварительный вариант преобразования Хоха

Пусть J — отрезок контура с углом наклона $0 \leq \alpha < 180^\circ$, S -параметром S^J и T -параметрами T_1^J и T_2^J . Преобразованием Хоха назовем преобразование, сопоставляющее этому отрезку два множества $P^i_J, i=1,2$ содержащих его прямоугольников. Если прямоугольник содержит отрезок J , то

а) при $0 \leq \alpha < 90^\circ$ определены параметры φ и S_1 или φ и S_2 этого прямоугольника и заданы границы для параметров S_2, T_1 и T_2 или S_1, T_1 и T_2 (см. Рис. 3);

б) при $90^\circ \leq \alpha < 180^\circ$ определены параметры φ и T_1 или φ и T_2 этого прямоугольника и заданы границы для параметров S_1, S_2 и T_2 или S_1, S_2 и T_1 .

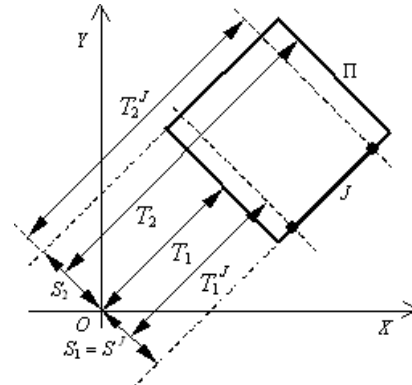


Рис. 3

Поэтому

а) при $0 \leq \alpha < 90^\circ$

$$P^1_J = \{p = (\varphi, S_1, S_2, T_1, T_2) \mid \varphi = \alpha, S_1 = S^J, S_2 \geq S^J, T_1 \leq T_2^J, T_2 \geq T_1^J\};$$

$$P^2_J = \{p = (\varphi, S_1, S_2, T_1, T_2) \mid \varphi = \alpha, S_1 \leq S^J, S_2 = S^J, T_1 \leq T_2^J, T_2 \geq T_1^J\};$$

б) при $90^\circ \leq \alpha < 180^\circ$

$$P^1_J = \{p = (\varphi, S_1, S_2, T_1, T_2) \mid \varphi = \alpha - 90^\circ, S_1 \leq T_2^J, S_2 \geq T_1^J, T_1 = S^J, T_2 \geq S^J\};$$

$$P^2_J = \{p = (\varphi, S_1, S_2, T_1, T_2) \mid \varphi = \alpha - 90^\circ, S_1 \leq T_2^J, S_2 \geq T_1^J, T_1 \leq S^J, T_2 = S^J\}.$$

3.2.2 Первый вариант — преобразование Хоха с параметрами

Учитывая, что угол наклона, S -параметр и T -параметры отрезка определены не точно, а также то, что прямоугольник не может содержать отрезки с близкими, но различными углами наклона, введем в рассмотрение погрешности: ε_φ — для угла наклона и ε_S — для S - и T -параметров прямоугольника. Эти погрешности являются *параметрами преобразования Хоха*, которое будет теперь действовать так: каждому отрезку J с углом наклона α , S -параметром S^J и T -параметрами T_1^J и T_2^J сопоставляются два множества прямоугольников: $P^1_{J(\varepsilon_\varphi, \varepsilon_S)}$ и $P^2_{J(\varepsilon_\varphi, \varepsilon_S)}$, к одной из сторон которых близок отрезок J :

а) при $0 \leq \alpha < 90^\circ$

$$P^1_{J(\varepsilon_\varphi, \varepsilon_S)} = \{ p = (\varphi, S_1, S_2, T_1, T_2) \mid \alpha - \varepsilon_\varphi \leq \varphi \leq \alpha + \varepsilon_\varphi, \\ S^J - \varepsilon_S \leq S_1 \leq S^J + \varepsilon_S, S_2 \geq S^J + \varepsilon_S, T_1 \leq T_2^J - \varepsilon_S, T_2 \geq T_1^J + \varepsilon_S \};$$

$$P^2_{J(\varepsilon_\varphi, \varepsilon_S)} = \{ p = (\varphi, S_1, S_2, T_1, T_2) \mid \alpha - \varepsilon_\varphi \leq \varphi \leq \alpha + \varepsilon_\varphi, \\ S_1 \leq S^J - \varepsilon_S, S^J - \varepsilon_S \leq S_2 \leq S^J + \varepsilon_S, T_1 \leq T_2^J - \varepsilon_S, T_2 \geq T_1^J + \varepsilon_S \};$$

б) при $90^\circ \leq \alpha < 180^\circ$

$$P^1_{J(\varepsilon_\varphi, \varepsilon_S)} = \{ p = (\varphi, S_1, S_2, T_1, T_2) \mid \beta - \varepsilon_\varphi \leq \varphi \leq \beta + \varepsilon_\varphi, \\ S_1 \leq T_2^J - \varepsilon_S, S_2 \geq T_1^J + \varepsilon_S, S^J - \varepsilon_S \leq T_1 \leq S^J + \varepsilon_S, T_2 \geq S^J + \varepsilon_S \};$$

$$P^2_{J(\varepsilon_\varphi, \varepsilon_S)} = \{ p = (\varphi, S_1, S_2, T_1, T_2) \mid \beta - \varepsilon_\varphi \leq \varphi \leq \beta + \varepsilon_\varphi, \\ S_1 \leq T_2^J - \varepsilon_S, S_2 \geq S_1^J + \varepsilon_S, T_1 \leq S^J - \varepsilon_S, S^J - \varepsilon_S \leq T_2 \leq S^J + \varepsilon_S \},$$

где $\beta = \alpha - 90^\circ$.

В данном случае тот факт, что отрезок J близок к одной из сторон прямоугольника, означает принадлежность этого прямоугольника множеству $P^1_{J(\varepsilon_\varphi, \varepsilon_S)}$ или $P^2_{J(\varepsilon_\varphi, \varepsilon_S)}$.

3.2.3 Второй вариант преобразования Хоха

Здесь, чтобы не усложнять изложение, мы даем описание второго варианта преобразования Хоха в несколько упрощенной форме.

Поскольку при повороте отрезка J вокруг своего центра на небольшой угол значения параметров S_1 , S_2 , T_1 и T_2 содержащего его прямоугольника могут значительно изменяться (величина изменения зависит от удаленности отрезка от начала координат), мы будем считать, что если отрезок J близок к одной из сторон прямоугольника, то середина отрезка должна располагаться от этой стороны на расстоянии, не большем ε_S . Кроме того, во втором варианте преобразования Хоха мы будем учитывать максимальный A_{max} и минимальный B_{min} размер прямоугольника (см. пункт 3.4.1). Более точно, пусть середина отрезка J имеет координаты (X_{Jc}, Y_{Jc}) , а его начало и конец — координаты (X_{Jb}, Y_{Jb}) и (X_{Je}, Y_{Je}) соответственно. Тогда преобразование сопоставит отрезку J два множества: $Q^1_{J(\varepsilon_\varphi, \varepsilon_S)}$ и $Q^2_{J(\varepsilon_\varphi, \varepsilon_S)}$, которые будут определяться так:

а) при $0^\circ \leq \alpha < 90^\circ$

$$Q^1_{J(\varepsilon_\varphi, \varepsilon_S)} = \{ p = (\varphi, S_1, S_2, T_1, T_2) \mid \alpha - \varepsilon_\varphi \leq \varphi \leq \alpha + \varepsilon_\varphi,$$

$$Y_{Jc} * \cos \varphi - X_{Jc} * \sin \varphi - \varepsilon_S * \cos(\varphi - \alpha) \leq S_1 \leq Y_{Jc} * \cos \varphi - X_{Jc} * \sin \varphi + \varepsilon_S * \cos(\varphi - \alpha),$$

$$Y_{Jc} * \cos \varphi - X_{Jc} * \sin \varphi + B_{min} * \cos(\varphi - \alpha) \leq S_2 \leq Y_{Jc} * \cos \varphi - X_{Jc} * \sin \varphi + A_{max} * \cos(\varphi - \alpha),$$

$$X_{Jb} * \cos \varphi + Y_{Jb} * \sin \varphi - A_{max} * \cos(\varphi - \alpha) \leq T_1 \leq X_{Je} * \cos \varphi + Y_{Je} * \sin \varphi - \varepsilon_S * \cos(\varphi - \alpha),$$

$$X_{Je} * \cos \varphi + Y_{Je} * \sin \varphi + A_{max} * \cos(\varphi - \alpha) \geq T_2 \geq X_{Jb} * \cos \varphi + Y_{Jb} * \sin \varphi + \varepsilon_S * \cos(\varphi - \alpha) \},$$

$$Q^2_{J(\varepsilon_\varphi, \varepsilon_S)} = \{ p = (\varphi, S_1, S_2, T_1, T_2) \mid \alpha - \varepsilon_\varphi \leq \varphi \leq \alpha + \varepsilon_\varphi,$$

$$Y_{Jc} * \cos \varphi - X_{Jc} * \sin \varphi - \varepsilon_S * \cos(\varphi - \alpha) \leq S_2 \leq Y_{Jc} * \cos \varphi - X_{Jc} * \sin \varphi + \varepsilon_S * \cos(\varphi - \alpha),$$

$$Y_{Jc} * \cos \varphi - X_{Jc} * \sin \varphi - B_{min} * \cos(\varphi - \alpha) \geq S_1 \geq Y_{Jc} * \cos \varphi - X_{Jc} * \sin \varphi - A_{max} * \cos(\varphi - \alpha),$$

$$X_{Jb} * \cos \varphi + Y_{Jb} * \sin \varphi - A_{max} * \cos(\varphi - \alpha) \leq T_1 \leq X_{Je} * \cos \varphi + Y_{Je} * \sin \varphi - \varepsilon_S * \cos(\varphi - \alpha),$$

$$X_{Je} * \cos \varphi + Y_{Je} * \sin \varphi + A_{max} * \cos(\varphi - \alpha) \geq T_2 \geq X_{Jb} * \cos \varphi + Y_{Jb} * \sin \varphi + \varepsilon_S * \cos(\varphi - \alpha) \};$$

б) при $90^\circ \leq \alpha < 180^\circ$ в определениях множеств $Q^1_{J(\varepsilon_\varphi, \varepsilon_S)}$ и $Q^2_{J(\varepsilon_\varphi, \varepsilon_S)}$ нужно заменить α на $\beta = \alpha - 90^\circ$, а также поменять местами S_1 с T_1 и S_2 с T_2 .

3.3 Описание первого алгоритма.

3.3.1 Входные параметры первого алгоритма

Алгоритму необходимы следующие входные параметры:

- 1) набор отрезков контуров $J_n, n = 1, \dots, M$;
- 2) погрешности: ε_φ — для угла наклона и ε_S — для S -параметров и T -параметров прямоугольника.

3.3.2 Выходные параметры первого алгоритма

На выходе алгоритм дает:

- 1) набор множеств значений параметров, каждое из которых с учетом заданных погрешностей соответствует прямоугольнику, к каждой стороне которого близок хотя бы один отрезок;
- 2) для каждого из найденных прямоугольников — набор индексов отрезков, близких к сторонам этого прямоугольника.

3.3.3 Первый алгоритм

Первый алгоритм использует первый вариант преобразования Хоха (см. п. 3.2.2) и содержит следующие шаги:

Шаг 1. Для каждого отрезка J_n строим множество $P^i_{J_n(\varepsilon_\varphi, \varepsilon_S)}, i = 1, 2, n = 1, \dots, M$.

Шаг 2. Для каждой пары отрезков $(J_m, J_n), m, n = 1, \dots, M$, ищем пересечения $P^{ij}_{J_m, J_n(\varepsilon_\varphi, \varepsilon_S)}$ всех соответствующих

им множеств $P^i_{J_m(\varepsilon_\varphi, \varepsilon_S)}$ и $P^j_{J_n(\varepsilon_\varphi, \varepsilon_S)}, i, j = 1, 2$.

Например, в случае, когда отрезки J_m и J_n имеют углы наклона α_m и α_n , S -параметры S_m и S_n , и T -параметры $T_{1,m}, T_{2,m}$ и $T_{1,n}, T_{2,n}$, которые подчинены условиям $0^\circ \leq \alpha_m \leq \alpha_n < 90^\circ, S_m \leq S_n$ и $T_{1,m} \leq T_{1,n}, T_{2,m} \leq T_{2,n}$, множество

$P^{11}_{J_m, J_n(\varepsilon_\varphi, \varepsilon_S)}$ можно описать следующим образом. Если

$\alpha_n - \alpha_m > 2\varepsilon_\varphi$ или $S_n - S_m > 2\varepsilon_S$, то множество $P^{11}_{J_m, J_n(\varepsilon_\varphi, \varepsilon_S)}$

пусто, иначе оно имеет вид:

$$P^{11}_{J_m, J_n(\varepsilon_\varphi, \varepsilon_S)} = \{ p = (\varphi, S_1, S_2, T_1, T_2) \mid$$

$$\alpha_n - \varepsilon_\varphi \leq \varphi \leq \alpha_m + \varepsilon_\varphi, S_n - \varepsilon_S \leq S_1 \leq S_m + \varepsilon_S,$$

$$S_2 \geq S_n + \varepsilon_S, T_1 \leq T_{1,m} + \varepsilon_S, T_2 \geq T_{2,n} - \varepsilon_S \}.$$

Шаг 3. Строим граф G , в котором вершинами служат множества $P^i_{J_m(\varepsilon_\varphi, \varepsilon_S)}, i = 1, 2, m = 1, \dots, M$, и пара вершин

$P^i_{J_m(\varepsilon_\varphi, \varepsilon_S)}$ и $P^j_{J_n(\varepsilon_\varphi, \varepsilon_S)}$ соединена ребром в том и

только в том случае, когда пересечение $P^{ij}_{J_m, J_n}(\varepsilon_\varphi, \varepsilon_S)$ этих множеств не пусто.

Шаг 4. Находим в графе G максимальные клики, то есть максимальные полные подграфы (об алгоритме поиска максимальных клик в графе см., например, работы [11-13]).

Шаг 5. Находим пересечения множеств, входящих в максимальные клики графа G . Все эти пересечения будут не пусты и будут соответствовать множествам прямоугольников, к сторонам которых близко наибольшее количество отрезков $J_n, n = 1, \dots, M$.

Шаг 6. Из найденных пересечений выбираем те, в которых области изменения всех параметров ограничены. Такие пересечения будут содержать только прямоугольники, к каждой из сторон которых близок хотя бы один отрезок. На этом работа алгоритма заканчивается.

3.4 Описание второго алгоритма

3.4.1 Входные параметры второго алгоритма

Алгоритму необходимы следующие входные параметры:

- 1) набор отрезков контуров $J_n, n = 1, \dots, M$;
- 2) погрешности: ε_φ — для угла наклона и ε_S — для S -параметров и T -параметров прямоугольника;
- 3) минимальный и максимальный допустимый угол наклона прямоугольника;
- 4) границы заданной области изображения, в которой будет производиться поиск;
- 5) способ поиска прямоугольников (поиск прямоугольников с заданными размерами сторон, с заданной площадью, с заданным периметром или с заданным отношением сторон);
- 6) один из следующих наборов параметров, задающих размеры прямоугольника:
 - a) A_{min}, A_{max} — минимальная и максимальная длина большей стороны прямоугольника;
 - b) B_{min}, B_{max} — минимальная и максимальная длина меньшей стороны прямоугольника;
 - c) диапазон значений площади прямоугольника;
 - d) диапазон значений периметра прямоугольника;
 - e) Q_{min}, Q_{max} — минимальное и максимальное значение отношения меньшей стороны прямоугольника к большей.

3.4.2 Выходные параметры второго алгоритма

Второй алгоритм имеет такие же выходные параметры, как первый.

3.4.3 Второй алгоритм

Второй алгоритм использует второй вариант преобразования Хоха (см. п. 3.2.3) и, кроме того, отличается от предыдущего варианта алгоритма способом нахождения тех точек пространства параметров, вблизи которых расположено наибольшее число множеств $Q^i_{J_n}(\varepsilon_\varphi, \varepsilon_S), i = 1, 2$. Здесь идея состоит в том, что мы отдельно обрабатываем каждый параметр, строя для каждого из них проекции множеств $Q^i_{J_n}(\varepsilon_\varphi, \varepsilon_S)$ на ось значений этого параметра, и находя затем точки оси, принадлежащие наибольшему числу этих

проекций. Это позволяет ускорить работу данного алгоритма по сравнению с первым.

Второй алгоритм содержит следующие шаги:

Шаг 1. Для каждого отрезка J_n строим множества $Q^i_{J_n}(\varepsilon_\varphi, \varepsilon_S), i = 1, 2, n = 1, \dots, M$.

Шаг 2. Проектируем все множества $Q^i_{J_n}(\varepsilon_\varphi, \varepsilon_S)$ на ось φ , в результате получаем некоторый набор интервалов. Упорядочиваем левые и правые границы этих интервалов. Находим максимальные группы из не менее чем трех интервалов, имеющих непустое пересечение. Назовем эти группы φ -наборами.

Шаг 3. Для каждого φ -набора строим проекции всех множеств $Q^i_{J_n}(\varepsilon_\varphi, \varepsilon_S)$, соответствующих интервалам из этого набора, на ось S_1 , в результате получаем новый набор интервалов. Упорядочиваем левые и правые границы этих интервалов. Находим максимальные группы из не менее чем трех интервалов, имеющих непустое пересечение. Назовем эти группы (φ, S_1) -наборами. Проверяем выполнение ограничений, определяемых входными параметрами.

Шаг 4. Для каждого (φ, S_1) -набора строим проекции всех множеств $Q^i_{J_n}(\varepsilon_\varphi, \varepsilon_S)$, соответствующих интервалам из этого набора, на ось S_2 , в результате снова получаем набор интервалов. Упорядочиваем левые и правые границы этих интервалов. Находим максимальные группы из не менее чем трех интервалов, имеющих непустое пересечение. Назовем эти группы (φ, S_1, S_2) -наборами. Проверяем выполнение ограничений, определяемых входными параметрами.

Шаг 5. Для каждого (φ, S_1, S_2) -набора строим проекции всех множеств $Q^i_{J_n}(\varepsilon_\varphi, \varepsilon_S)$, соответствующих интервалам из этого набора, на ось T_1 , получая в результате очередной набор интервалов. Упорядочиваем левые и правые границы этих интервалов. Находим максимальные группы из не менее чем трех интервалов, имеющих непустое пересечение. Назовем эти группы (φ, S_1, S_2, T_1) -наборами. Проверяем выполнение ограничений, определяемых входными параметрами.

Шаг 6. Для каждого (φ, S_1, S_2, T_1) -набора строим проекции всех множеств $Q^i_{J_n}(\varepsilon_\varphi, \varepsilon_S)$, соответствующих интервалам из этого набора, на ось T_2 , в результате чего опять получаем набор интервалов. Упорядочиваем левые и правые границы этих интервалов. Находим максимальные группы из не менее чем трех интервалов, имеющих непустое пересечение. Назовем эти группы $(\varphi, S_1, S_2, T_1, T_2)$ -наборами. Проверяем выполнение ограничений, определяемых входными параметрами.

Шаг 7. Просматриваем все $(\varphi, S_1, S_2, T_1, T_2)$ -наборы и удаляем те из них, которые вложены в какой-либо из предыдущих $(\varphi, S_1, S_2, T_1, T_2)$ -наборов или совпадают с ним.

Шаг 8. Для каждого из оставшихся $(\varphi, S_1, S_2, T_1, T_2)$ -наборов строим пересечение всех множеств $Q^i_{J_n}(\varepsilon_\varphi, \varepsilon_S)$, соответствующих интервалам из этого набора. Все эти пересечения будут не пусты и будут соответствовать множествам прямоугольников, к сторонам которых близко наибольшее количество отрезков $J_n, n = 1, \dots, M$.

Шаг 9. Из найденных пересечений выбираем те, в которых области изменения всех параметров ограничены. Такие пересечения будут содержать только прямоугольники, к каждой из сторон которых близок хотя бы один отрезок.

Шаг 10. Из оставшихся пересечений берем те, в которых ограничены области изменения угла и еще трех параметров, и восстанавливаем значение пятого параметра, используя входные данные. Затем запоминаем наборы индексов отрезков, близких к сторонам каждого из найденных прямоугольников. На этом работа алгоритма завершается.

3.5 Результаты анализа второго алгоритма

Описанные выше алгоритмы были реализованы на языке С и протестированы в реальной системе, предназначенной для определения местоположения электронных деталей и компонентов на полутоновых изображениях по стилизованному шаблону. Мы приводим результаты тестирования только второго алгоритма, поскольку он работает значительно быстрее первого.

Второй алгоритм позволяет реализовать запросы широкого класса, например: осуществлять поиск прямоугольников заданной области изображения по заданным сторонам, диапазону углов наклона прямоугольников, по заданной площади, периметру или отношению сторон.

Алгоритм обладает рядом преимуществ перед другими алгоритмами поиска прямоугольников (например, см. [14]). Он позволяет находить прямоугольники как в ситуациях, когда один или несколько прямоугольников принадлежат одному контуру, так и когда один прямоугольник образован несколькими контурами. Алгоритм дает возможность определять положение прямоугольников на векторной модели даже в случае, когда углы наклона отрезков контуров, соответствующих прямоугольнику, сильно отличаются от углов наклона сторон прямоугольника. Успешный поиск возможен также в случаях, когда на векторной модели имеются только части сторон, и когда полностью отсутствует одна из них.

Надежность алгоритма определяется качеством сглаживания контуров в векторной модели изображения. Если границы контуров сильно изрезаны, то есть отрезки, образующие контура, имеют малую длину и большой разброс углов наклона, результат работы алгоритма сильно зависит от погрешности измерения углов наклона отрезков ε_φ . Если задать эту величину большой (например, $\varepsilon_\varphi > 15^\circ$), то алгоритм предложит большее количество кандидатов в прямоугольники, часть которых может быть построена на контурах, имеющих слабую степень сходства с прямоугольниками. Если же величина ε_φ мала (например, $\varepsilon_\varphi < 5^\circ$), то возможна потеря некоторых кандидатов.

В качестве примера использования алгоритма приведем результаты поиска микросхем на полутоновом изображении электронной платы (Рис. 4).

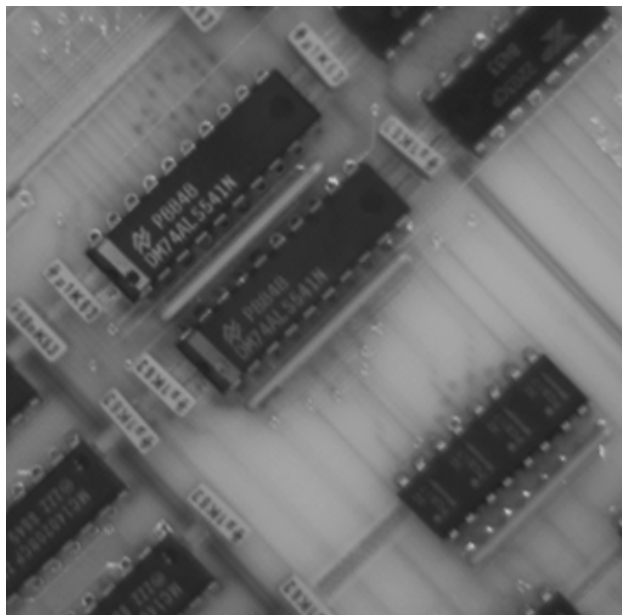


Рис. 4

На Рис. 5 представлена векторная модель данного изображения (контура выделены черным цветом) и показаны результаты поиска прямоугольников для значения $\varepsilon_\varphi = 5^\circ$ (два найденных прямоугольника выделены жирными серыми линиями). На Рис. 6 приведены результаты поиска прямоугольников для $\varepsilon_\varphi = 10^\circ$. Во втором случае число найденных прямоугольников (три) больше, что служит иллюстрацией к сказанному выше

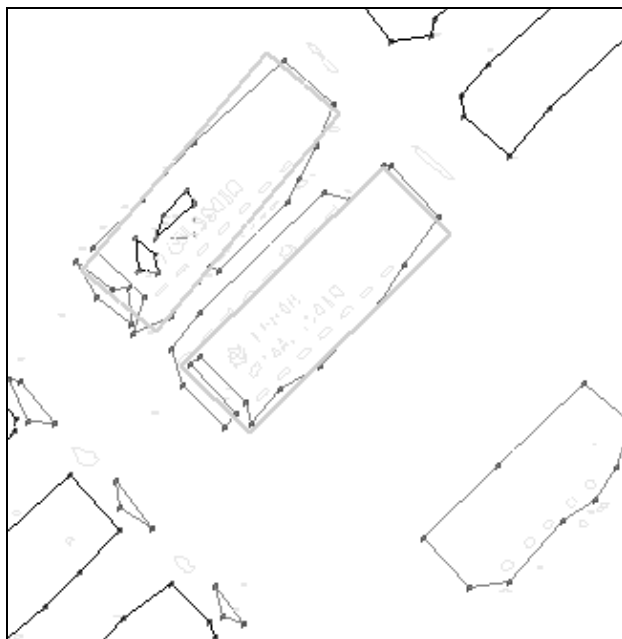


Рис. 5

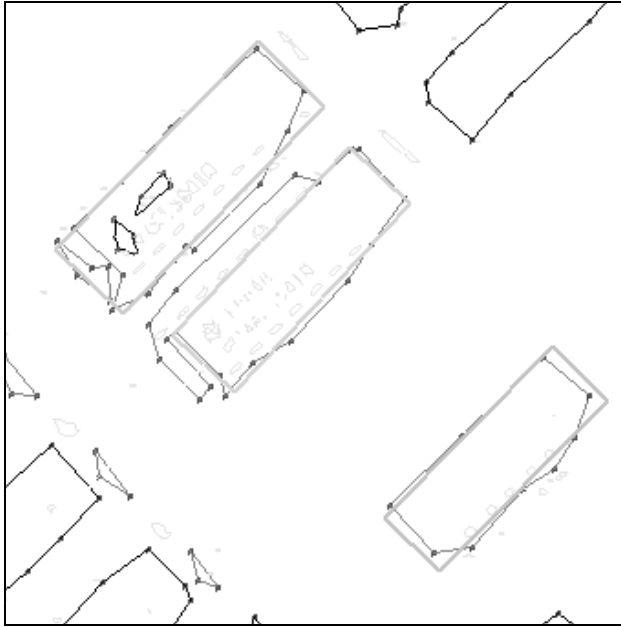


Рис. 6

4. ЗАКЛЮЧЕНИЕ

В работе описано применение преобразования Хоха для распознавания прямоугольников на векторной модели изображения. Аналогично приведенным алгоритмам могут быть построены алгоритмы поиска других многоугольников.

Один из разработанных алгоритмов позволяет легко реализовать широкий набор параметров и условий распознавания входных изображений и имеет ряд преимуществ по сравнению с другими алгоритмами поиска прямоугольников (см., например, [14]). В частности, он допускает существенные вариации входных данных: значительное отклонение углов наклона отрезков контуров, отсутствие одной стороны, отсутствие частей некоторых сторон, принадлежность нескольких прямоугольников одному контуру, или наоборот, принадлежность одного прямоугольника нескольким контурам.

Проведенные эксперименты подтвердили упомянутые свойства алгоритма.

В заключение мы выражаем благодарность Олегу Александровичу Башкирову за полезные обсуждения и помощь в рецензировании работы.

5. ПЛАНЫ ПО РАЗВИТИЮ АЛГОРИТМОВ

В дальнейшем мы предполагаем расширить возможности алгоритмов так, чтобы обеспечить поиск прямоугольников, к сторонам которых близки не менее двух отрезков, и добиться поддержки оставшихся запросов 1e)-2d), перечисленных в пункте 2.

Кроме того, при построении множеств, состоящих из прямоугольников, к одной из сторон которых близок отрезок

J , мы можем придавать им некоторый вес, который будет затем учитываться при нахождении пересечений этих множеств. Величину веса можно взять, например, пропорциональной длине отрезка J .

Можно также ввести вес для пересечений таких множеств, например, придавая больший вес пересечениям тех множеств, которые соответствуют отрезкам, принадлежащим одному контуру.

Видимо, стоит рассмотреть также и другие варианты преобразования Хоха, например, определенные на наборах из отрезка и трех точек.

6. БИБЛИОГРАФИЯ

- [1] Hough P.V.C. *Method and means for recognizing complex patterns*. // US Patent 3069654. Dec. 18 1962.
- [2] Duda R.O., Hart P.E. *Use of Hough transformation to detect lines and curves in pictures*. // Commun. ACM. V. 15, № 1. Jan. 1972. P. 11-15.
- [3] Aguado A.S., Montiel E., Nixon M.S. *Invariant characterization of the Hough transform for pose estimation of arbitrary shapes*. // Electronic proceedings of the eleventh british machine vision conference University of Bristol 11-14 september 2000.
- [4] Бовырин А.В., Губанов А.В., Колесов А.И., Курякин В.Ф., Родюшкин К.В., Чудинович Б.М. *Библиотека функций векторной обработки видео изображений в реальном времени*. // IX конференция «Математические методы распознавания образов», тез. док. М. 1999. С. 145-147.
- [5] Бабулин Н.А. *Построение и чтение машиностроительных чертежей*. М. «Высшая школа». 1987. 319 с.
- [6] Абламейко С.В., Лагуновский Д.М. *Обработка изображений. Технология, методы, применение*. Минск. «Амалфея». 2000. 304 с.
- [7] Васин Ю.Г., Башкиров О.А., Чудинович Б.М., Гервагин Ю.Г. *Подсистема растрового ввода-вывода и обработки графической информации РАСТР I*. // Автоматизация обработки сложной графической информации. Межвузовский сборник. Издание Горьковского государственного университета им. Лобачевского. Горький, 1984. С. 235-254.
- [8] Васин Ю.Г. *Автоматизированные системы и технологии создания и обработки цифровых карт и планов городов*. // Тез. докл. 5-й конф. "Методы и средства обработки сложной графической информации". Н. Новгород. 1998. С. 97.
- [9] Васин Ю.Г., Башкиров О.А., Рудометов С.Б. *Математические модели структурированного описания графических изображений*. // Автотметрия. № 1. 1988. С. 50-54.
- [10] Illingworth J., Kittler J. *A survey of the Hough transform* // Computer vision, graphics and image processing. 1988. V. 44, № 1. P. 87-116. Библиогр.: p. 111-116 (136 назв.).
- [11] Берж К. *Теория графов и ее применение*. М. «Иностранная литература». 1962.
- [12] Bron C., Cerbosch J. *Finding all cliques of an undirected graph* // Comm. ACM. 1973. V. 16, № 9. P. 575-577.

[13] Кристофидес Н. *Теория графов. Алгоритмический подход*. М. «Мир». 1978. 432 с.

[14] P. Dewaele, M. Ibison, P. Vuylsteke *A trainable rule-based network for irradiation field recognition in AGFA's ADC system*. // Proc. medical imaging, SPIE. 1996. V. 2708. P. 72-80.

Об авторах

Алексей В. Губанов, Нижегородский филиал фирмы “TESIS-Software BVBA”.

E-mail: gubanov@tesis.nnov.ru

Олег Е. Галкин, Нижегородский филиал фирмы “TESIS-Software BVBA”.

E-mail: galkin@tesis.nnov.ru

Николай В. Ильинов, Нижегородский филиал фирмы “TESIS-Software BVBA”.

E-mail: iljinov@tesis.nnov.ru