一种区域边界的识别和区域标记算法与应用

陈世福 潘金贵 胡滨 罗秋清
(南京大学计算机科学系，南京 210008)

ALGORITHMS FOR REGION EDGE RECOGNITION AND REGION MARK AND THEIR APPLICATION

Chen Shifu, Pan Jingui, Hu Bin and Luo Qiuqing
(*Department of Computer Science, Nanjing University, Nanjing* 210008)

Abstract In the case of region edge recognition when several edges intersect or there are some edges which belong to several different regions, the algorithm REMR (Recognition of the Edges of the Minimal Region) can find out the whole edges of the minimal region by line tracing. Then algorithm EBRM (Edge-Based Region Mark) marks the region with the recognized edges. These two algorithms have been applied to the image preprocessing system for computer-controlled embroidery.

摘要 本文介绍了一种区域边界的识别算法和一种区域标记的算法. 前者在有若干区域的多条边界相交及存在边界公用的情况下，通过直线生成的方法能寻找出每一个最小区域的完整的区域边界；后者则能在识别出区域边界的基础上进行区域的标记，把区域分割出来. 这两个算法已用于电脑刺绣编程系统的图象预处理系统，效果良好. 文中并通过实例说明了这两个算法的特点.

§1. 引 言

目前的电脑刺绣编程系统其图案大多是通过计算机交互输入，或是通过数字化仪输入的，速度慢，效果差. 为此我们采用了平板扫描仪输入刺绣画稿，通过图象处理技术自动提取轮廓信息，输入的速度快，精度高，并可进行局部修改、区域识别等处理，解决了人机交互方式或由数字化仪输入方式的速度慢、灵活性差等瓶颈问题.

提高刺绣编程自动化的关键技术是区域的识别. 区域的识别是通过直线生成的方法，对图象中的点信息进行向量化，以寻找出区域的边界，并在此基础上进行区域的标记，把区域分割出来. 在实现这一技术时，我们引入了当前方向优先法则和惯性法则，明显地提高了处理速度，并起到了消除干扰的良好效果，在一定程度上解决了直线的间断问题. 但是，当有若干个区域的多条边界相交，以及存在边界公用的情况时，通过直线生成的方法来寻找出一系列最小区域的完整的区域边界是非常困难的. 这就要求我们对算法进行完善，并能解决上述问题，以满足自动化程度较高的要求. 本文描述的最小区域边界识别算法及基于边界的区域标记算法不仅提高了电脑刺绣编程系统的可靠性和效率，而且使系统可以根据图象的具体特点自动选取针法和自动进行处理.

本文1991年4月定稿. 作者陈世福，教授，主要研究领域为人工智能，知识工程和图象处理. 潘金贵，副教授，主要研究领域为知识工程与软件工程. 胡滨，硕士研究生，主要研究领域为人工智能和图象处理. 罗秋清，工程师，硕士毕业于南京大学，主要研究领域为人工智能和图象处理.

§2. 区域边界的识别算法

图象经过图象预处理系统的提取轮廓、细化等处理后，常常会出现如图1(a)所给出的线条型图象；该图象由区域的边界组成，如果按照一般的算法处理[11]，可得图1(b)所示的结果，其中 * 表示线条的端点，即首先从A-B-C-A提取一条边界，再由A-D-B提取第二条边界，最后是D-C；显然并未得出图1(a)中的α、β以及γ这三个区域的独立完整的边界.当然我们可以采取在直线生成时，一旦遇分叉点则停止搜索的办法，这样只能得到一些更零散的线条，如图1(c)所示，虽然可以由它们组合成完整的区域边界，如把AD、DB以及BA连接起来可以形成区域的边界，但这样操作很复杂，而且还存在区域间边界公用的问题.为此考虑到图象分割可靠性和经济性的要求，我们设计了一种既简单又高效的最小区域边界识别算法REMR (Recognition of the Edge of the Minimal Region)，该算法对图1(a)处理之后，可以直接得出区域α的边界A-D-B-A，区域β的边界A-D-C-A以及区域γ的边界D-B-C-D.

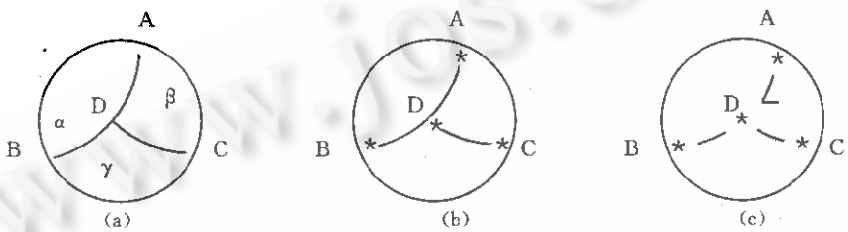


图1

假设图象经过局部修改、消除干扰、提取轮廓、细化、去短枝等处理后，能满足：(1) 图象得到充分的细化；(2) 图象中不存在无意义的线条，包括任何非区域边界的线条.在满足以上两个条件的情况下，REMR算法如下(为便于叙述，使用了下列符号：scan_line 当前扫描行；region_count 闭合边界数；start point 起始点；init point 初始点；scan_direction 当前搜索方向；fork_buffer[max_fork]分叉点数组；follow_buffer[max_fork]分叉点后继结点；fork_points 分叉点数；edge_buffer[max_edge]边界点数组；edge_up_pointer 边界点上指针；edge_down_pointer 边界点下指针；P1 当前边界点；P2 后继边界点；follow 有/无后继边界点(true/false))：

REMR 算法：

```
Step 1：scan_line：=0；
        region_count：=0；
Step 2：while 当前行无边界点 do
          begin
            scan_line：=scan_line+1；
            if scan_line>max_line then
              算法结束
          end；
        P1：= 最左边的一个边界点；
Step 3：edge_up_pointer：=(max_edge)/3-1；
        edge_down_pointer：=(max_edge)/3；
        fork_points：=0；
        scan_direction：=右；
        start_point：=P1；
        init_point：=P1；
Step 4：根据当前方向优先法则寻找 P2；
          if 存在 P2 then
            follow：=true
          else
            follow：=false；
Step 5：if follow=false then
          if scan_direction= 右 then
            goto Step 7
          else
            goto Step 9
        else
          if is_fork(P1)=true then
            goto Step 8；
Step 6：if scan_direction=右 then
          begin
            edge_buffer[edge_down_pointer]：=P1；
            edge_down_pointer：=edge_down_pointer+1
```

```
        end
      else
        begin
          edge_buffer[edge_up_pointer]:=P1;
          edge_up_pointer:=edge_up_pointer-1
        end;
      P1:=P2;
      goto Step 4;
Step 7:scan_direction:=左;
      start_point:=P1;
      P1:=init_point;
      goto Step 4;
Step 8:if P1 在 fork_buffer 中出现 then
        goto Step 10
      else
        begin
          if scan_direction=右 then
            按右手法则选择 P2
          else
            按左手法则选择 P2;
          fork_buffer[fork_points]:=P1;
          follow_buffer[fork_points]:=P2;
          fork_points:=fork_points+1;
          goto Step 6
        end;
Step 9:if 可以构成闭合边界 then
        begin
          生成区域边界;
          region_count:=region_count+1;
        end;
      if fork_points=0 then
        在图象数组中删除 edge_buffer 中的点
      else
        在图象数组中删除从 init_point 到其前后第一个
        分叉点之间的边界点;
      goto Step 2;
Step 10:把 edge_buffer 中两相同分叉点之间的一串边界
        点生成边界;
        region_count:=region_count+1;
        if 这两个相同分叉点的后继点相同 then
        begin
          在图象数组和 edge_buffer 中删除这一串边界
          点;
          fork_points:=fork_points-1
        end;
      goto Step 4;
```

上述算法用到了右手法则和左手法则，其定义如下：

定义 1：右手法则 在搜索后继边界点时从右向左，即按逆时针方向寻找.

如在图 2 中，如果当前点为 (i，j)，前一个点为 P，则后继点的搜索次序应为 1，2，3，4，5，6，7.

7	6	5
P	(i,j)	4
1	2	3

图2

定义 2：左手法则 在搜索后继边界点时从左向右，即按顺时针方向寻找.

如在图 2 中，如果当前点为 (i，j)，前一个点为 P，则后继边界点的搜索次序应为 7，6，5，4，3，2，1.

由于 REMR 算法的扫描次序为从上到下，从左到右，所遇到的初始点必为某区域的最上面一点，所以一开始把当前方向定义为右是合理的. 在图 3 中，初始点为 A，由于当前方向为右，所以在分叉点 B 处，按右手法则即可定出其搜索路径为 A-B-F，而非 A-B-D；显然可得到 A-B-F-C-E-A，这样就可得到一条完整的区域 α 的边界. 左手法则类同，就不再赘述了.

从图 3 可知，边 B-F-C 为区域 α 与 β 的公用边界，因而在生成区域 α 的边界之后，在图象数组中还不能完全删除这些边界点，必须先判断出在已生成的区域边界中究竟哪些是公用边界点，哪些不是公用边界点，然后把非公用边界点从图象数组中删除，保留公用边界点.

显然初始点必定不是当前区域的公用边界点，若是公用边界点的话，则该点已被使用过一次，所以无保留价值，可以删除. 那么该点的前一点和后继点，如果是非分叉点，也具有相同的属性. 即可得到如下结论：从初始点起，分别向左、向右到第一个分叉点为止，这一串边界点必为非公用边界点，可以从图象数组中删除；如果这个区域边界上无分叉点，就可以认为该区域与其他区域无公用边界，所以可以从图象数组中删除.

不难看到，本算法在进行边界跟踪时，并未立即删除搜索到的边界点，这样对于有环的情况就可能出现死循环，所以算法要在 Step 8 和 Step 10 检查分叉点：一旦搜索到分叉点则检查该分叉点是否遇到过，如果遇到过就表示出现了闭合环，需要生成该闭合区域边界，并从图象数组中删除该边界点.

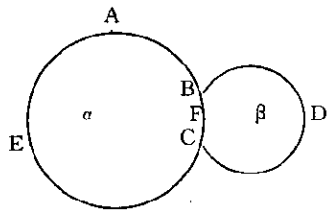


图3

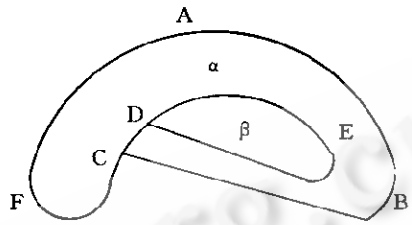


图4

对于图 4 所示的区域自相交的情况，当按从 A-B-C-D-E-D 顺序搜索时，D 作为分叉点已出现了两次，这样能找出区域 β 的边界 D-E-D；若立即删除 D-E-D 则会丢失区域 α 的边界，因此在 Step 8 中用数组 follow_buffer 保存分叉点的后继点，而在 Step 10 中则根据相同分叉点的后继结点是否相同，来决定是否删除闭合边界点.

上述算法应用结果表明具有较好的自适应性和准确性，能够正确地识别出比较复杂的轮廓图象中最小闭合区域的完整的区域边界.

§3. 区域标记算法

电脑刺绣编程系统不仅要求能识别出每一个最小区域的完整边界，而且还要求能够提供有关区域的整体信息，即一个区域有哪些边界以及区域的内外边界等.

目前对区域进行标记使用得最多的是区域生长的方法，一般可归纳为以下几类：(1) 局部法：根据象素本身的特性或其邻域的特　将其放置到某一区域. (2) 整体法：根据分布在整个图象上大量象素的特性将象素分成各个独立的区域. (3) 分裂合一法：利用表示区域和边界的图结构来合并或分裂区域.

以上各种方法均是在图象一级对象素点阵加以处理的，分别需要经过区域定位、区域搜索、区域标记等一系列的处理；这样不仅耗费较多的 CPU 时间，而且控制起来也颇为复杂. 为了提高效率，我们设计了一种基于边界的区域标记算法 EBRM (Edge-Based Region Mark)，把处理的对象从象素点阵级上升到轮廓线条级，这样不仅能够大大节约 CPU 时间，而且还简化了算法的控制.

EBRM 处理对象是 REMR 算法生成的区域边界，因此可以断定在区域标记过程中所遇到的闭合折线均属于某一个区域的边界.

关于两个区域边界的包含关系定义如下：

定义 3：a，b 为两条区域的边界 (闭合折线)，a 包含 b (a ⊃ b) 当且仅当对 b 上任一点 A 均在 a 的内部.

EBRM 算法步骤因篇幅所限，不再赘述，其主要思想是：首先对边界环按其最高点的值从高到低进行排序，然后逐一加上标志；在每一条边界环加上(外边界)标志时，要判断是否有包含于该环的其他边界环，若有的话则对其加上(内边界)标志；如此逐步循环，直至所有边界环均被加上标志为止.

下面进一步说明该算法，首先介绍一下区分内外边界的方法，给出标记函数 margin_type(a) 如下：

$$margin\_type(a)=\begin{cases}0 & \text{当 } a \text{ 为区域的外边界时}\\ 1 & \text{当 } a \text{ 为区域的内边界时}\end{cases}$$

其中 a 表示一条区域边界. 显然有如下定理：

定理 1：如果边界 a,b 有 a⊃b，且不存在边界 c，使得 a⊃c，c⊃b，则必有 margin_type(b) = 1-margin_type(a).

由定理 1 及区域的定义[11]，可以得到如下结论：如果有边界 a,b，且 a⊃b，并且不存在边界 c 使得 a⊃c，c⊃b，且有margin_type(a)=0，即 a 为外边界，则必有 a,b 属于同一区域.

上面已经介绍如何区分内、外边界，以及如何标记属于同一区域的各个边界，下面将介绍如何判定边界的包含关系.

根据图 5 所给出的坐标关系，可以定义下列一组函数：

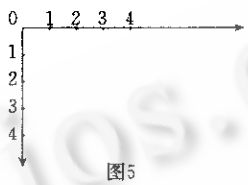
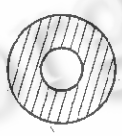


图5


(a)


(b)

图6

定义 4：令 a 为一区域边界，$A_i(X_i, Y_i)$，$A_j(X_j, Y_j)$为 a 上的点，则：

$$\begin{aligned}&most\text{-}up(a)=Y_i && \text{当且仅当} && Y_i<=(\forall\ A_j\in a\ Y_j)\\ &most\text{-}down(a)=Y_i && \text{当且仅当} && Y_i>=(\forall\ A_j\in a\ Y_j)\\ &most\text{-}left(a)=X_i && \text{当且仅当} && X_i<=(\forall\ A_j\in a\ X_j)\\ &most\text{-}right(a)=X_i && \text{当且仅当} && X_i>=(\forall\ A_j\in a\ X_j)\end{aligned}$$

则判断 a⊃b 的公式如下：

公式 1：most_up(a)<=most_up(b) and most_down(a)>=most_down(b) and most_left(a)<=most_left(b) and most_right(a)>=most_right(b)

当然要判别区域边界 a，b 的包含关系仅有公式 1 是不够的，因为它只能判断图 6(a)的情况，而对于图 6(b)的情形则会出错，所以公式 1 仅仅是判别区域边界包含关系的必要条件，而非充分条件，它仅可用作初步判定，即不满足公式 1 的 a，b 必不存在 a⊃b 的关系.

当满足公式 1 时，要确定是否有 a⊃b，必须作进一步的判定，算法如下：

Step 1：在区域边界 b 上选择一点 $B(X_0, Y_0)$，使得 X_0=most_left(b)；

Step 2：作直线 $Y=Y_0$，它与区域边界 a 在 $X>X_0$处的交点为 $C_1, C_2, \cdots, C_k$；若无满足条件的交点则 k=0；

Step 3：若 k 为偶数，则 a⊃b；否则 a⊃b.

该判定算法可由下面的定理[10]进行验证：

定理 2：过不在闭合折线上一点 P 作射线，若它与闭合折线有奇数个交点，则该点在闭合折线内部，否则该点在闭合折线外部.

§4. 处理实例

应用 REMR 算法和 EBRM 算法，可以自动生成适合刺绣的画稿. 下面给出经上述算法处理的一个实例，如图 7 所示. 图 7(a)是一个“福”字的原始图象，经过图象预处理系统的局部修改、消除干扰、提取轮廓、充分细化、去短枝等处理后，得到图 7(b)的图象；再经 REMR 算法生成边界和 EBRM 算法标记区域后，得到图 7(c)所示的区域标记，其中的二元组为标记信息，第一个参数表示边界所属的区域号，第二个参数表示边界的属性，即是外边界还是内边界，0 表示外边界，1 表示内边界. 电脑刺绣编程系统根据标出的区域信息，能够进行自动处理，选取各种针法进行自动编针.

结论：REMR 算法和 EBRM 算法在 PC/386 机上用 Turbo C 2.0 编程实现，用于电脑刺绣编

程系统的图象预处理系统. 该系统使得电脑刺绣编程系统能根据图象文件的特点对其进行自动处理，选取各种针法进行自动编针，明显地提高了刺绣样板的生产效率，并生产出较为理想的绣品，通过实际的应用，说明了这两个算法的有效性和高效性.

(a)　　(b)　　(c)

图 7　处理实例

REMR 算法和 EBRM 算法具有一定的通用性，不仅适用于电脑刺绣的图象处理系统，也适用于其他领域的图象处理.

参考文献

[1] R. H. J. M. Otten and L. P. P. P. van Ginneken, The Annealing Algorithm ,Kluwer Academic Publisher, 1989.

[2] W. K. Pratt, Digital Image Processing , John Wiley & Sons Inc., 1978.

[3] D. H. Ballard and C. M. Brown, Computer Vision , Prentice－Hall Inc., 1982.

[4] Ernest L. Hall, Computer Image Processing and Recognition , Academic Press, 1979.

[5] Peter Stucki, Advances in Digital Image Processing , Plenum Press, 1979.

[6] V. A. Kovalevsky, Image Pattern Recognition , Springer－Verlay, 1980.

[7] Michael P. Ekstrom, Digital Image Processing Techniques , Academic Press Inc., 1984.

[8] Edward R. Dougherty, Ph. D., Charles R. Giardina, Ph. D., Matrix Structured Image Processing , Prentice－Hall Inc., 1987.

[9] Refael C. Conzalez Paul Wintz, Digital Image Processing , Addison－Wesley Publishing Company Inc., 1977.

[10] 希尔伯特，《几何基础》，北京科学出版社，1959.

[11] 罗秋清，用于电脑刺绣的图象预处理系统，南京大学计算机系硕士论文，1991. 6.