

CEIP: 用于电脑刺绣的图象 预处理系统

陈世福 罗秋清 潘金贵

(南京大学计算机科学系, 210008)

CEIP: THE IMAGE PREPROCESSING SYSTEM FOR COMPUTER CONTROLLED EMBROIDERY

Chen Shifu, Luo Qiuqing and Pan Jingui

(Computer Science Department of Nan jing University, 210008)

ABSTRACT

This paper describes the design and implementation of CEIP (Computer Embroidery Image Preprocessing) of Computer Embroidery Programming System. CEIP uses scanner to input embroidery design and creates the graphs suitable for embroidery by recognition and processing. Using effective image processing algorithms and techniques, the system makes embroidery improved.

摘 要

本文描述了用于电脑刺绣编程系统的图象预处理系统CEIP的设计方法和实现技术。CEIP系统采用平板扫描仪输入刺绣画稿, 经过图象预处理系统的识别和处理, 生成适合刺绣的画稿。本系统所使用的图象处理算法和技术, 明显地提高了刺绣样板的生产效率和绣品质量。

§ 1. 引言

如何把高新技术应用于生产实践, 改造传统产业, 提高工业生产自动化及智能化程度, 乃是一项非常紧迫的课题。

1990年3月19日收到, 1990年5月25日定稿。

刺绣是我国的一项传统手工艺，其历史悠久，制作精美，刺绣品倍受人们青睐。国内外市场需求量大。然而，由于它的制作是手工的，其产量受到限制。虽然，目前国内外有为数不少的电脑刺绣机问世，但还有很多问题尚未解决。其一，目前的电脑刺绣机并不能很好地生产出具有中国特色的绘画式绣品，即艺术表现能力差。其二，其图案、图象的输入大都是通过计算机交互输入，或者是数字化仪输入的，速度慢，效果差。其三，针法的编制自动化程度低。为此，我们开展了智能刺绣编程系统的研制，把人工智能技术用于刺绣生产，解决自动编针问题，并且刺绣图案由平板扫描仪输入，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，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。该系统适用于国内外多种型号的电脑刺绣机，经一年多的实际使用，明显地提高了刺绣样板的生产效率和绣品质量。

本文描述了智能刺绣编程系统中的图象预处理系统CEIP 的设计方法和实现技术。

§ 2. 系统结构

智能刺绣编程系统的工作流程如图1 所示。

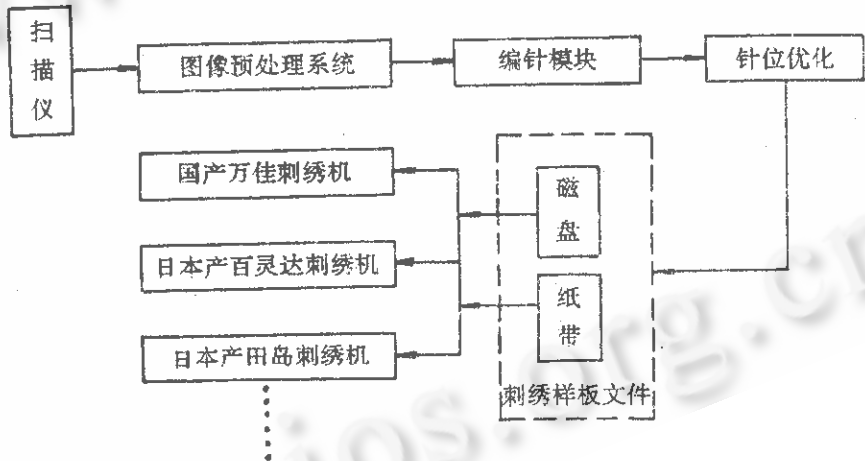


图1 智能刺绣编程系统工作流程

电脑自动刺绣工艺过程如下：由扫描仪接收的原始图象画稿，经图象预处理系统的处理产生适合刺绣的画稿，再经编针系统和针位优化系统处理，最后产生针位文件，即所要刺绣图案的刺绣样板，并具有磁盘和纸带两种介质，可适用于国产万佳刺绣机，日本产的百灵达和田岛刺绣机等其它型号的多种刺绣机。

2.1 系统功能

图象预处理系统的主要功能有：图象文件的输入、输出，提供平滑处理，边界判定，去噪声、膨胀、腐蚀、轮廓、修饰等匹配操作，并可把边界点转换为线条，进行最小闭合区域的识别、区域标记，以及根据图象文件的特点对其进行自动处理。

2.2 数据结构

图象预处理系统所处理的对象主要是TIFF 格式的图象文件。

图象文件有两种：二值图象文件和多值灰度图象文件，它们均由文件头和数据区两部分组成。文件头部分存贮有关图象的一些特性值，如水平长度、垂直长度、水平分辨率、垂直分辨率、图象类型(二值图，多级灰度图象)，以及每个点的信息所占的位数等等。数据区则顺序存放每点的信息，二值图象文件中是用0或1来表示该象素点是黑还是白，而多级灰度图象文件中则依次给出各点的灰度级数。

2.3 系统结构

图象预处理系统的结构如图2所示。

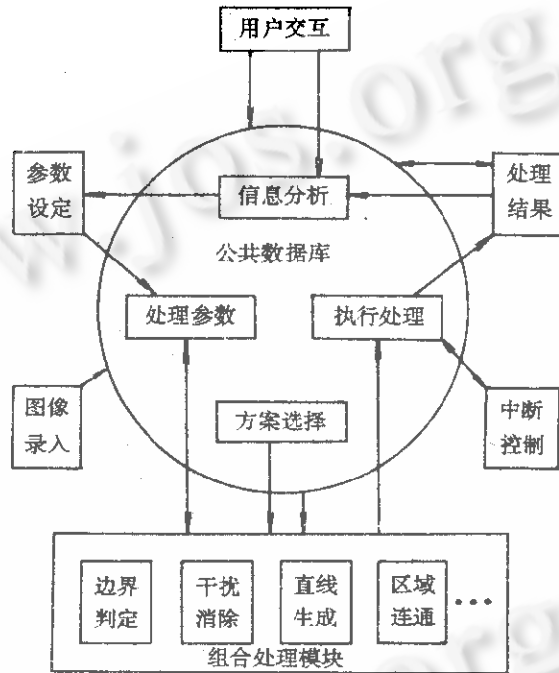


图2 CEIP 系统结构

该系统处理图象的流程如下：

第一步：录入图象并转换成易于处理的数据格式，存放在公共数据库内。

第二步：设置适当的参数，由系统选择执行方案以选择组合处理模块。

第三步：执行处理，在处理过程中进行中断控制。

第四步：对处理结果和反馈信息进行分析并允许用户干预，其交互是贯穿整个系统运行的。若处理结果不佳，再回到第二步，直至产生满意的处理结果。

由此，不难看出该系统具有交互性、重复性及中断控制等特点。

§ 3. 实现技术和算法

该系统的核心是组合处理模块, 它包括边界判定、干扰消除、直线生成和区域连通等子模块, 下面将分别予以阐述. 其它部分, 因篇幅有限本文将省略.

3.1 边界判定和生成

边界判定的基本任务是找出图象的轮廓点. 当图象由扫描仪输入后, 首先分析图象文件格式, 然后把数据按照适当的和便于处理的形式组织起来进行边界判定. 边界判定采用3×3模板的方法, 对模板中的各个元素(点), 又称象素, 赋予一定的权值, 来计算判定某一点是否为边界点.

模板中点的名称如图3(a)所示, 其中间的点表示为(i, j), 而相应的P(i, j)则表示点(i, j)的象素值.

(i-1, j-1)	(i-1, j)	(i-1, j+1)
(i, j-1)	(i, j)	(i, j+1)
(i+1, j-1)	(i+1, j)	(i+1, j+1)

(a)

0	1	0
1	-4	1
0	1	0

(b)

图3 3×3模板及权值

在二值图中有: P(i, j)=1, 当(i, j)为黑时; P(i, j)=0, 当(i, j)为白时.

在灰度图中P(i, j)为相应的灰度值, 而在彩色图中P(i, j)则表示相应的颜色值. 算法如下: 令Laplas=P(i-1, j)+P(i, j-1)+P(i, j+1)+P(i+1, j)-4×P(i, j), 若abs(Laplas) > δ, 则可判定(i, j)为边界点, 否则(i, j)为非边界点.

例如: 以图3(b)各象素赋以的相应权值, 计算点(i, j)是否为边界点, 在实践中我们发现对大多数图象用图3的模板及权值计算(i, j)的边界情况, 效果较好, 而且速度明显提高.

图象通常由色块和线条组成. 对于色块, 用上述算法可判断出边界点; 对于仅由线条组成的图象可跳过边界判定这一步. 对于既有色块又有线条的图象, 用上述算法处理就不那么理想, 而且非收敛.

在处理兼有色块和线条的图象时, 采取符号比较方法代替上面的数值运算, 本文称之为补救式判别法. 下面仅对判定垂直边界进行讨论, 水平边界情况大体相同.

在一个3×3的模板中:

如果P(i, j)=P(i, j+1), 则P(i, j)不是边界点, 除非已被预置;

如果P(i, j) <> P(i, j+1), 则P(i, j)是边界点; 同时

如果P(i, j-1)=P(i, j+1), 则P(i, j-1)被重置为非边界点; 否则

如果P(i, j-1) <> P(i, j), 则P(i, j+1)被预置为边界点.

垂直边界判定的补救式算法用类Pascal语言描述如下:

```

if p(i, j)=p(i+1)
then if p(i, j) 被预置
      then p(i, j) 保持不变
      else p(i, j)=0
else p(i, j)=2
      if p(i, j-1)=p(i, j+1)

```

then $p(i, j-1)=0; p(i, j)=1$

else if $p(i, j) <> p(i, j-1)$

then $p(i, j)=1, p(i, j+1)$ 被预置为2

处理结果表明, 对于较复杂的图象采用该方法处理效果有明显改善. 该算法与一般的边界判定算法的区别在于它不仅可判定边界, 还可区分单线与块, 并且该算法是收敛的.

3.2 干扰消除

图象制作或输入时, 如果图象上的某些点的值不理想, 就会造成直线的不连续, 色块中有污点等现象. 干扰消除要删除这些干扰现象: 现以二值图象为例描述所采用的干扰消除方法.

我们采用了模式匹配技术找出干扰点并给予消除. 通常伴有直线间断问题的修复和粗直线转化为细直线的细化等处理. 修复处理采用如图4所示的一组标准模板.

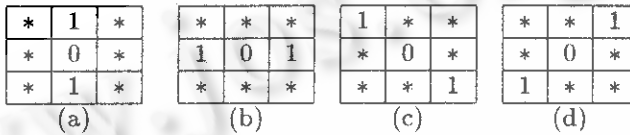


图4 修复处理的标准模板

如果图象的点阵值与模式匹配, 则把 $P(i, j)$ 置为1, 否则为0, 其中*表示0或1.

细化处理采用如图5所示的一组标准模板, 如果象素矩阵与模式匹配, 则 $P(i, j)$ 置为“0”.

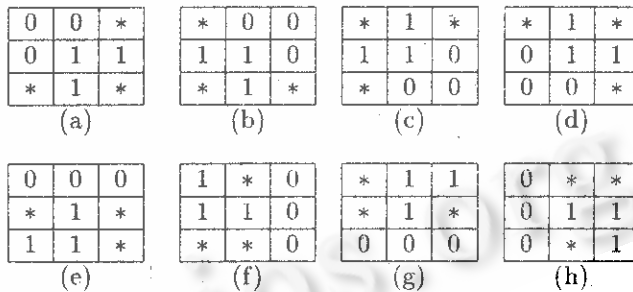


图5 细化处理标准模板

上述 3×3 模板的处理效果是非常明显的. 但对于 3×3 模板其功能是有限的, 还会留有许多干扰无法消除, 如果增大模板, 处理的时间和处理的复杂程度又会成倍增加, 故本系统没有增大模板, 而是把某些干扰消除的任务放在直线生成或区域连通之中去完成.

3.3 直线生成算法

前面讨论的处理均是在点一级进行的. 用点来表示一幅图象不仅占用的存储空间大, 耗费的处理时间多, 而且对于某些处理来说也是极不方便的. 一个基本而有效的方法是生成直线, 即把前面生成的边界点有效地组织起来, 用线段来表示图象.

直线生成的基本方法是找出一串相连的点, 把它们首尾连起来, 如果它不能近似地表示这些点就把距离该直线最远的点找出来, 并用此点与原来的首尾分别相连, 再分别

对这两根线段作上述的处理, 直到满意为止.

由一串点生成直线的步骤是显然的, 但关键的是如何以最佳方式找出这一串点. 当然, 我们可先按一定的顺序, 如从左到右、自上而下地搜索整个象素矩阵, 先找出一个边界点, 记下来并把它擦除, 然后再在它的周围8个点上按任意的顺序寻找下一个边界点, 依此类推. 这种处理虽很简单, 但速度不快而且不符合人们的观察习惯. 为了提高速度, 改善其效果, 本系统在实现时, 我们引入了惯性法则和当前方向优先法则. 所谓当前方向优先法则, 即在传统方法的基础上加上一条启发式信息. 该法则明显提高了处理速度且消除干扰的效果良好. 如图6所示, 如果当前点为(i, j), 而当前方向为从左到右, 即前一个找到的点为(i, j-1), 则寻找下一个边界点的顺序如图6(b)所示, 权值越小者越先寻找, 而对于图7(a), 显然不会出现图7(b)的情况, 只可能产生图7(c)的直线. 假如约定: 若相连点数小于 α ($\alpha = 3$) 则作为污点除去, 则可达到消去干扰的作用.

为了避免加上“ α ”限制后, 有时会产生丢失点错误的的可能性, 我们对上述算法作了改进, 即在找到第一点之后记下它的位置, 在搜索结束之后, 回到第一点, 寻找其它可能的边界点.

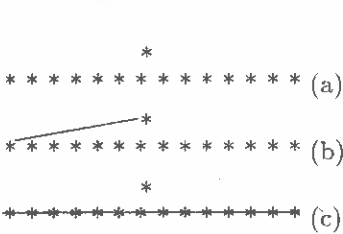
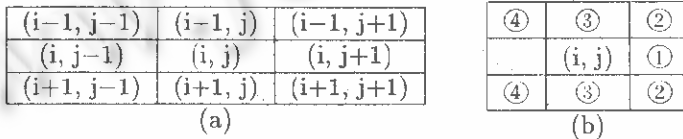


图7

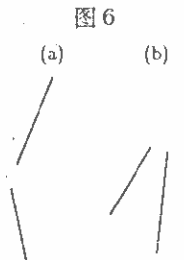


图8 间接类型



图9 连接方式

为了消除直线间断的问题, 我们采用了所谓的“惯性法则”的启发式信息. 亦即记下向同一方向寻找边界点的次数, 若在当前位置周围不存在边界点时, 就根据已寻找次数的大小决定是否按该方向或向前寻找或寻找多少点. 显然次数越大, 向前寻找的点就可相应的多一些. 如果找到一点是边界点, 并且它非孤立点, 而在该方向上也可继续找出边界点, 则把刚才越过的那些非边界点也置为边界点并记下, 这样从一定程度上可以解决直线的间断问题.

3.4 区域连通

当生成了直线之后, 公共数据区内存放的也仅是一些无规律的线段. 为了把它们有效地组织起来, 以获得图象信息, 改进图象, 为以后的处理提供必要的准备, 我们必须进行区域连通.

当然, 对于相连线段的处理是简单的. 但复杂的是不相连线段的处理, 需要考虑下列问题: (1) 对不同类型的间断如何进行最适当的连接, (2) 在与一条线段相邻的多条线段中, 选择哪一条; (3) 如何确定允许的间断长度等. 下面将描述在系统实现中所采用的

方法.

对于图8中所给出的两种不同的间断类型, 有两种连接方式: (1) 延长不平行的两线段使它们相交, 这样线段总数不变; (2) 把两线段的最近点用一线段相连, 这样线段总数增加. 对于(a)型间断, 两种连接方式区别不大, 对(b)型间断用两种方式连接, 则可产生图9中的两种情况, 即通过用户接口让用户选择某种连接方式. 如果要系统自动完成, 则根据两个最近点间的距离及其相交点的距离之间的关系, 以及该图的全局信息, 决定究竟选择哪一种连接方式. 当然, 所给出的间断类型和连接方式的区分并不一定是最完备的, 但我们认为该方法较为实用且易于实现.

当一条线段有可能与两条或多条其它的线段连接时, 我们的选择标准有两种方法: (1) 根据线段间的距离; (2) 根据两条线段的位置关系. 对第一种方法是毋庸说明的, 而对第2种就图10而言, 线段(a)与(b)的可能性最大, 其次是(c), 接下来是(d), 最后才是(e). 当然除了上述两种情况外, 还需考虑图象的全局信息, 如相连时是否与其它线段相交, 线条型线段与区域型线段等. 在某些情况下, 如果没有用户干预, 就需在上述两种方法中进行权衡选择.

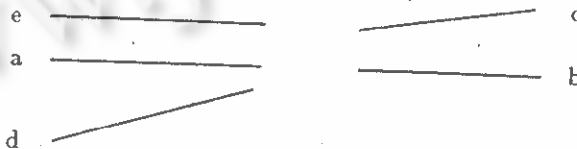


图10

相距多远的线段进行相连, 预先确定是不合适的, 要根据要求而定, 系统提供了两种解决方法: 第一种方法是对每一幅图象由用户指定一个阈值; 第二种方法是让这个阈值从小到大自动循环迭代. 具体做法是先确定一个较小的值, 进行一次区域连通, 当完成后增大这个阈值, 再进行一次区域连通. 这时参加连通的就不再是线段了, 而是上一次区域连通后生成的区域作为一个整体进行, 如此反复. 当然, 这个阈值的增加是有上限的, 对于仅由色块组成的图象来说, 其边界必定是闭合的, 所以其阈值是使其各区域闭合为止. 对于其它情况, 则可由用户确定, 也可由系统内的缺省值决定.

区域连通又可以归属于区域标记或区域划分, 而区域划分中还需考虑线条的走向以及线条的复用, 即一线条同时为两个区域的边界. 为此我们提出了一种“区域标记算法”以及“区域划分算法”. 限于篇幅, 这里不再描述.

§ 4. 处理实例

下面将给出系统如何处理图象“禄”字的实例, 以使读者对该系统及其处理流程有一直观的了解. 首先用扫描仪输入画稿, 可得到如图11的图象, 经过前面所描述的边界判定、细化、消除干扰并通过直线生成与区域连接处理将得到图12的处理结果. 对图12所给出的图象所生成的文件即可交付自动刺绣编针模块进行自动编针, 得到如图13的绣品.



图 11



图 12



图 13

§ 5. 结束语

该系统是在PC386 机上用Turbo C2.0 编程实现的, 处理速度快, 可扩充性、可移植性好. 该系统提供的人机接口友善, 具有多种交互方式, 例如, 通过多级菜单可随意调用各类操作, 提供的各种模拟输入, 可使用户方便地输入各种参数、橡皮条和游动光标、确定处理区域等. 另外, 该系统还设计了图象修饰功能, 允许用户在任何情况下, 方便地增减点或线段, 并提供了联机帮助文件, 指导用户使用该系统. 系统对用户透明, 除了显示各阶段图象的处理结果外, 还显示必要的操作信息, 帮助用户进行必要的干预.

该系统的实现使得在电脑刺绣领域用扫描仪输入图象成为可能, 它大大简化了迄今使用的人工方式或由数字化仪输入刺绣画稿的操作, 明显提高了刺绣样板的生产效率, 已被实际应用, 并生产出较为理想的绣品. 整个系统已经商品化, 并已打入国际市场.

本系统所采用的一些图象处理方法和算法以及灵活的交互方式, 具有一定的通用性, 可以较好地处理一些中等复杂程度的图象, 不仅适用于电脑刺绣的图象处理, 也适用于其它领域的图象处理.

参考文献

- [1] W.K.Pratt, DIGITAL IMAGE PROCESSING, John Wiley & Sons Inc., 1978.
- [2] D.H.Ballard C.M., Brown, COMPUTER VISION Prentice-Hall, Inc., 1982.
- [3] ERNESTL HALL, Computer Image Processing and Recognition Academic Press 1979.
- [4] Peter Stucki, ADVANCES IN DIGITAL IMAGE PROCESSING Plenum Press 1979.
- [5] V.A.Kovalevsky, Image Pattern Recognition Springer-Verlag, 1980.
- [6] R.H.J.M.Otten and L.P.P.P.van Ginneken, "The Annealing Algorithm", Kluwer Academic Publishers, 1989.