维普资讯 http://www.cqvip.com

一种基于累计角度偏差的圆弧拟合方法

周 翔 王崇骏 伍 静 陈世福

(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘 要 矢量化是图像自动编针中很重要的步骤。为了改进现有的圆弧拟合方法,我们提出了基于累计角度偏差的圆弧拟合方法。并且已成功运用于ECAD中的图像自动编针子系统中,获得了良好的效果。

关键词 Bezier曲线,矢量化,圆弧拟合,图像自动编针,刺绣

An Approach of Fitting-Bezier Based on Totaling Deviations

ZHOU Xiang WANG Chong-Jun WU Jing CHEN Shi-Fu

(National Key Laboratory of New Software Technology, Nanjing University, Nanjing 210093)

Abstract Vectorization is a crucial step in the automatic punching. To overcome the shortcoming of the current arc fitting algorithm, we propose a vectorization method based on totaling deviations to fit circular arcs. Furthermore, the method is implemented in the embroidery computed aided design system and the experimental results indicate that the method is more effective and efficient.

Keywords Bezier curve, Vectorization, Arc fitting, Automatic punching, Computer embroidery

1 引言

图像自动编针(Automatic Punching)是刺绣CAD系统(Embroidery Computed Aided Design System,简称ECAD)中重要的功能之一,其结果在很大程度上决定了该CAD系统的智能化水平,因此,对图像自动编针的研究引起了很多学者的兴趣。在ECAD中,处理流程如图1所示。

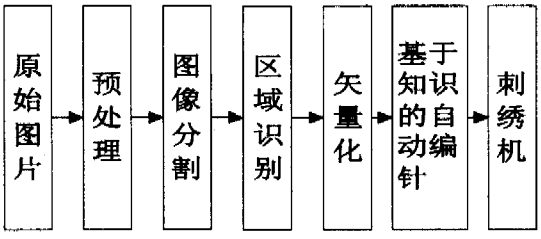

图1 图像自动编针流程

首先,通过图像分割算法将图像分割为不同的区域,并进行区域识别,然后将各个区域的边界进行矢量化,这样每个区域就成为由几何曲线环绕的封闭几何图形并作为编针的输入。由此可见,图像分割和矢量化是最基本的环节之一[1],对编针结果起重要影响作用。

刺绣图像区域分割的最终目的是对区域选择某种算法进行编针,而编针的计算都是对矢量化边界进行的。因此我们需要对分割出的区域识别其线、带、面类型,并为后续的自动编针提供矢量化边界轮廓。

本文对其中的矢量化环节展开研究,根据刺绣图像的特点以及用户操作的方便,考虑到业界对曲线有"3点成圆"的需求,改进了现有的圆弧拟合方法[2],提出了基于累计角度偏差的圆弧拟合方法,该改进算法可以明显提高精度。

另外,在刺绣CAD中,一般曲线造型是采用Bezier曲线[3]。因此,上述关于拟合结果还没有达到最终编针的输入需求,因此还要做一些后续处理,本文对这部分内容也给予了介绍和分析,并给出了实验结果。

2 相关的技术和方法

2.1 圆弧拟合

定义1 解析几何中圆的定义。其中(x_0,y_0)为圆心,r为半径。

$$(x-x_0)^2+(y-y_0)^2=r^2$$

在边界轮廓被分段后我们用等长的直线段对每一段边界轮廓进行拟合,得到等距分布的点序列,以次点序列作为待拟合点,进行圆弧拟合。

定义2 圆弧拟合是对直线分段点序列的处理步骤,若干个相邻的分段点在满足一定条件是可以识别为圆弧曲线,从而进一步减少存储空间并准确地逼近边界轮廓。

定义3 圆弧拟合的标准最大误差。

$$\varepsilon=\frac{\max_i|d_i|}{r}$$

其中r为拟合圆弧的半径,d_i为轮廓上第个i被拟合的轮廓点到拟合圆弧的距离。ε为误差。

2.2 圆弧拟合方法

要达到对一个等距点序列进行圆弧拟合的目的,常用的方法是从等距点序列的起始四点开始,设置窗口,并顺次移动此窗口,直到某四个点构成的三条线段不满足构成一个圆弧的条件。

判断条件描述如下:

设四点构成的三条等长直线段为L_1,L_2,L_3;并且L_1与L_2,L_2与L_3,L_1与L_3构成的夹角分别为α,β,γ,如果满足$|\alpha-\beta|\leqslant T_w$,($T_w$为角度阈值)且$(\gamma<\alpha)and(\gamma<\beta)$,则这三条直线属于同一个圆的一条圆弧。

但是,这种圆弧拟合方法存在弊端,例如当我们要拟合的圆弧呈波浪线时,有可能相邻的两条原本不属于同一圆形的圆弧会被识别成为一条圆弧。

维普资讯 http://www.cqvip.com

类似于图 2 的波浪形曲线应当被识别成为不同圆弧的拼接，而已有算法无法做到这一点。待拟合的原始图像如图 2 所示，实验结果如图 3 所示。

图 2　待检测的波浪形圆弧

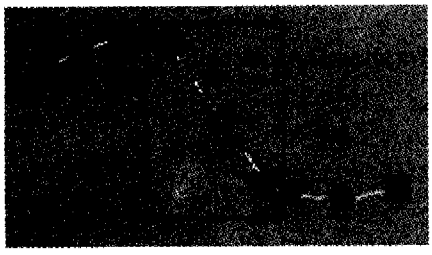

图 3　圆弧拟合方法的圆弧识别

与图 2 所示的原始图相比，在图 3 的中段（如图中方框处所示），由于遗漏了必要的控制点，使拟合出的曲线不够圆滑，明显丢失了曲线的细节信息。

所以，我们对文[2]中的圆弧拟合方法进行改进，提出一种基于累计角度偏差的圆弧拟合方法。

3　基于角度偏差的圆弧拟合方法

3.1　算法设计思想

针对已有算法不能对波浪形圆弧准确识别的缺陷，我们做了以下改进：在我们利用得到的线段夹角信息进行判别时，当累记的角度差值超过预定的阈值时，则当前的圆弧识别完毕，并开始新的圆弧识别。因为在检测呈波浪形的两条圆弧时，角度偏差值会迅速增大，从而不会被识别成为一条圆弧。弥补了上一节中算法的缺陷。

在边界轮廓被分段后，我们用等长的直线段对每一段边界进行拟合，得到等间距分布点的序列，再以这些等间距点为待拟合点，进行圆弧拟合。

同时在该算法执行之后，增加了圆弧延长和相邻圆弧合并算法，达到在精度允许范围内尽可能识别圆弧的目的。

整个圆弧识别的基本思路如下：

(1)使用累计角度偏差检测出初始圆弧段；

(2)检查每一个初始圆弧段，若满足条件，作适当的延长处理；

(3)检查相邻的圆弧段，若满足条件，作适当的合并处理。

定义 4　角度问题，当入射线段到出射线段成逆时针方向旋转时，角度取值范围为[0～π]，否则为(π～2π)。

定义 5　输出点的属性问题，输出的点分为直线点和曲线点。两个直线点之间为直线段，两个曲线点中间加上一个曲线点表示一段圆弧，属于此三点构成的圆。

3.2　算法描述

检测初始圆弧段运用了基于角度偏差的圆弧拟合方法，具体的算法描述如下：

输入：一组等距点的集合 P_i，$i=1\sim n$。

输出：标明了点的属性的控制点的序列，即是直线点还是曲线点。

取输入序列中的前三个点 P_1，P_2，P_3，$difsum_{thres}$ 为累计角度偏差阈值，$difsum$ 为累计角度偏差，θ_0 为初始角度，θ_{thres} 是预定的角度阈值(可取 3/4π)，ε 是预定的小角度值(取 π/36)

STEP1　WHILE(P_1，P_2，P_3 皆小于 n)
　　BEGIN
STEP2　P_1 属性取直线点压入控制点序列
STEP3　计算线段 P_1P_2 与 P_2P_3 所成夹角为 θ。
STEP4　IFθ 的值在
　　$[\theta_{thres}, \pi-\varepsilon)or(\pi+\varepsilon, 2\pi-\theta_{thres}]$
　　范围内，GOTO STEP6
STEP5　ELSE
　　P_1，P_2，P_3 都往前移动一位，GOTO STEP1
STEP6　调整参数 $\theta_0=\theta$，$difsum=0$，
　　$p_1=P_1$，$p_2=P_2$，$p_3=P_3$
STEP7　WHILE($difsum<difsum_{thres}$ and θ 的值在$[\theta_{thres}, \pi-\varepsilon)$ or $(\pi+\varepsilon, 2\pi-\theta_{thres}]$范围内)
　　BEGIN
STEP8　游标 $p_1p_2p_3$ 按输入序列的顺序往下移一位，且须保证不超出输入序列的范围，如果超出，则立刻 BREAK。
STEP9　计算当前三点的直线段所成夹角 θ
STEP10　调整参数 $difsum=difsum+(\theta-\theta_0)$
　　END
STEP11　IF 内层循环只执行了一次，P_1，P_2，P_3 都往前移动一位，GOTO STEP1
STEP12　ELSE 取 P_1 和 p_2 的中点，按照属性值为曲线点压入输出控制点序列
STEP13　P_1，P_2，P_3 都往前移动一位
　　END
STEP14　最后剩余的一个点作为直线点压入输出序列。
　　算法完毕

在识别出一段圆弧后，我们需要计算出圆弧所属圆的圆心和半径，这只需要运用简单的平面几何知识即可做到[4]。

3.3　圆弧检测之后的一些处理

3.3.1　圆弧段的延长　在使用累计角度偏差识别出初始圆弧后，圆弧段的长度较短，可以适当把两端点两侧的直线点归并到圆弧段来，从而达到优化目的。依据的条件是引入点到圆心的距离与半径的比值在可以接受的精度误差之内[5]。当然，在完成了圆弧延长之后需重新计算圆弧的圆心半径。

3.3.2　相邻圆弧的合并　对于前两步的结果，初始圆弧段得到了适当的延长，但依然存在一些本属于同一圆中的相邻圆弧被一些细微的噪声分隔开的情况。当这两段圆弧同属于一个圆的时候可以被合并。判断条件为两圆弧有相同或相似的圆心和半径，或者有相同的圆心且在衔接处有相同或相似的切线斜率。我们采取了第二种判断条件，公式描述如下：

设两个圆弧在 p 处衔接，它们的半径分别为 r_1，r_2，在点 p 处的切线方向分别为 θ_1，θ_2。若满足条件 $|r_1-r_2|<r_{thre}$ && $|\theta_1-\theta_2|<\theta_{thre}$，其中 r_{thre}，θ_{thre} 分别是预设定的半径和角度的阈值，则我们将此相邻圆弧合并。

3.2.3　Beizer 曲线拟合　在获得了曲线的控制点之后，我们开始构造 Bezier 曲线，利用轮廓控制点计算出两相邻轮廓点之间的形状控制点(Bezier 曲线的内点)[6]。

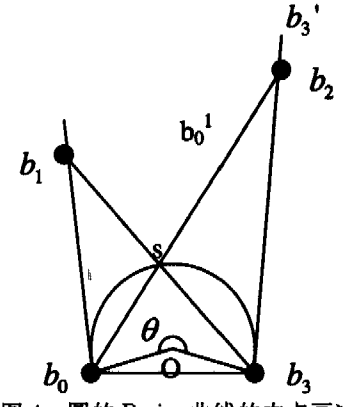

图 4　圆的 Bezier 曲线的内点画法

维普资讯 http://www.cqvip.com

对逼近圆弧的一段曲线，获得 Bezier 曲线内点的方法如下：

(1)取圆弧上两点 b_1，b_3，做切线 b_0b_0'，b_3b_3'。

(2)在圆弧上任选一点 s，连接 b_0s，b_3s。

延长与切线 b_0b_0'，b_3b_3'相交于 b_1，b_2 两点。

(3)b_1，b_2 即所求 Bezier 曲线的内点如图 4 所示。

4 算法的实验结果与分析

4.1 两种方法的效果比较

图 5 是使用基于角度的圆弧识别和基于累计角度偏差的圆弧识别方法的对图 2 进行识别的实验结果，明显可以看出图 5 的圆弧中段处本应被识别成为不同圆弧的部分没有正确识别，原因是已有算法会把波浪形圆弧误识别成为一段圆弧的缺点。

图 5 基于角度的拟合结果

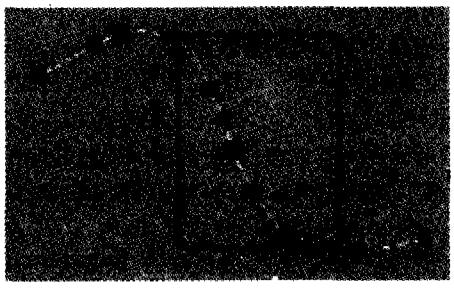

图 6 基于角度偏差的拟合结果

图 6 是使用基于角度偏差的圆弧识别方法对图 2 进行识别的实验结果，由于加入了更多的控制点，完好地保留了原始图的信息，准确地检测到波浪轮廓中间的衔接形状(对比两图中的方框处)。

4.2 圆拟合结果

我们使用近似于圆形的简单图形进行识别，其效果如图 7 所示。

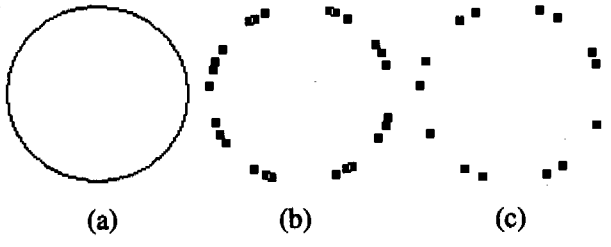

图 7 对圆拟合的实验结果

图 a 为原图，图 b 为使用了基于角度累计方法识别出的结果，图 c 是进行了圆弧延长，合并处理后的结果，最终以较少的控制点准确地识别出初始圆弧段的轮廓，得到较好的圆弧拟合效果(图中方形点为直线点，星形点为曲线点)。

4.2 Beizer 拟合结果

图 8，图 9 是根据求出轮廓控制点的曲线确定 Bezier 内点的效果图。图中实心点是轮廓控制点，空心点是 Bezier 内点。最终达到电脑刺绣图像自动编针的要求。

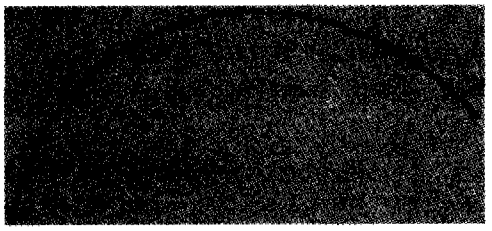

图 8 对一段圆弧取得控制点

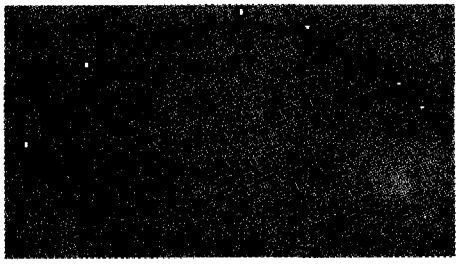

图 9 做出控制点对应的 Bezier 内点

结束语 本文介绍了一种基于角度累计的圆弧识别方法，是对原有圆弧拟合方法的改进，弥补了已有方法不能准确识别波浪形圆弧的缺陷。并成功地运用于 ECAD 中的图像自动编针子系统中，获得良好的效果。

针对前面提到的关于 ERP 的几个特点，在具体实施 ERP 软件开发中有如下几个技术应该被采纳：这些技术包括构件技术、软总线技术、中间件技术、异构数据库技术、安全机制等。

参 考 文 献

1 Parker J R. Extracting vectors from raster images. Computer & Graphics, 1988, 9

2 宋晓宇，王永会. 工程图自动矢量化算法的设计与实现. 中国图像图形学报，2000，5A(1)

3 孙家广. 计算机图形学. 北京：清华大学出版社，2001. 301～307

4 李庆华. 图像的矢量化方法预算法研究. 计算机应用研究，1995

5 王刚，林贻佳，宋乐乐. 直线及圆弧矢量化的统计算法. 中国图像图形学报，1998，3(1)

6 伍静. 刺绣图像自动编针中图像分割和矢量化技术的研究：[南京大学研究生学位论文]. 2005. 30～37