

#1777

第七届“认证杯”数学中国

数学建模网络挑战赛 承 诺 书

我们仔细阅读了第七届“认证杯”数学中国数学建模网络挑战赛的竞赛规则。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们接受相应处理结果。

我们允许数学中国网站(www.madio.net)公布论文，以供网友之间学习交流，数学中国网站以非商业目的的论文交流不需要提前取得我们的同意。

我们的参赛队号为：#1777

参赛队员（签名）：

队员 1：王立志

队员 2：张国营

队员 3：曹媛莉

参赛队教练员（签名）：

参赛队伍组别：研究生组

#1777

第七届“认证杯”数学中国

数学建模网络挑战赛 编 号 专 用 页

参赛队伍的参赛队号：（请各个参赛队提前填写好）：

#1777

竞赛统一编号（由竞赛组委会送至评委团前编号）：

竞赛评阅编号（由竞赛评委团评阅前进行编号）：

#1777

2014 年第七届“认证杯”数学中国 数学建模网络挑战赛第一阶段论文

题 目 关于矢量化过程中边界提取问题的模型

关 键 词 矢量化；边缘跟踪与拟合；梯度检测法；参数方程

摘 要：

摘要：矢量图，作为图像的一种表示方法，以其缩放不失真、存储空间小等特性，越来越受到人们的青睐，广泛应用于电影、医疗等重要领域。本文针对由位图到矢量图这一栅格化的逆过程，简单分析了其去噪，二值化，等问题，重点依据题目提出的寻找边缘方程的问题进行详尽的描述。现按照次重点简单叙述如下：

首先，本文在综合了多种矢量图提取算法的基础上，结合图像矢量化编辑的特点，提出了一种用于快速提取边缘曲线的方法—基于参数方程法，基于该方法实现了对轮廓信息的快速提取。通过聚类的方法将检测出的轮廓信息（像素）进行分类，并对聚类后的曲线信息进行拟合，使之成为便于操作管理的 B-样条曲线。

其次，在完成题目要求的前提下，对获取矢量图过程中的主要步骤简单介绍，对矢量化涉及的消噪、二值化等问题进行解释，以加深对模型应用的理解。并对矢量化过程中的某些特定名词进行解释，以便于更好的了解和理解复杂的过程。

最后，对模型的优缺点加以总结，展望模型的应用领域。

关键词：矢量化；边缘跟踪与拟合；梯度检测法；参数方程

参赛队号： #1777

所选题目： B 题

参赛密码
(由组委会填写)

#1777

英文摘要（选填）

Abstract: vector, as a representation of image, with its scaling distortion, such as small storage space characteristics, more and more people of all ages, widely used in the important field of film, medical etc.. This paper by the bitmap to vector inverse process diagram of this grid, a simple analysis of the denoising problem, two values, and so on, focus on the detailed description of the foundation for edge equation the questions proposed. In accordance with the focus on simple narrative time:

Above all, this paper based on synthesizing a variety of vector extraction algorithm, combined with the characteristics of the image vector editing, a method for rapid extraction - characteristic curve edge curve method is presented, the method to realize the fast extraction of contour information based on. Contour information through clustering method will be detected (pixel) classification, and to fit the curve information after clustering, make it convenient for operation and management of the B-spline curve.

Secondly, on the premise to complete the requirements of the subject, the main steps for obtaining vector graph in the process of simple introduction, the vector to eliminate dryness, two values are explained, in order to deepen the understanding of the model application. And the specific term vector in the process of interpretation, in the process in order to better understand and understand complex.

Last but not least, the advantages and disadvantages of the model are summarized, application prospect model.

#1777

目录

目录

1.简介.....	1
1.1 背景介绍.....	1
1.2 本文的目的和意义.....	1
2.矢量化过程中存在的问题.....	1
3.模型的建立.....	2
3.1 模型中涉及的字母.....	2
3.2 灰度图像的二值化.....	2
3.2.1 阈值分割法简介.....	2
3.2.2 计算阈值的方法.....	3
3.2.3 最佳阈值的选择.....	4
3.2.4 二值化结果.....	5
3.3 二值图像的消噪.....	6
3.3.1 二值图像的消噪.....	6
3.3.2 孤立点消除.....	7
3.3.3 空穴的消除.....	8
3.3.4 小面积颗粒的清除.....	8
3.3.5 自动去除斑点噪声的方法.....	9
3.4 边缘检测.....	9
3.5 边缘跟踪.....	10
3.6 边缘曲线的分段拟合.....	11
3.6.1 尖点提取.....	11
3.6.2 算法思想.....	12
3.6.3 算法设计.....	12
3.6.4 实验结果及分析.....	13
3.6.5 曲线拟合.....	14
4.总结及展望.....	17
4.1 总结.....	17
4.2 展望.....	17
参考文献.....	18

#1777

1.简介

1.1 背景介绍

随着信息技术的发展，图像越来越成为人们获取和感知信息的主要来源。特别是数字图像技术的发展，已广泛应用于电影、医疗、工业和航空航天等重要领域，在经济的发展中起到了相当重要的作用。常见的计算机平面图形的表示方法有位图和矢量图。位图是基于像素点的图，由一系列像素点组成的。常见的位图有点阵图、光栅图以及像素图等，其存储格式有 jpg, gif, bmp 等。矢量图则是由一系列曲线信息和颜色信息组成的抽象图，通常被压缩存储为 XML 文件。矢量图从本质上只是使用曲线方程对图形进行的精确描述，在以像素为基本显示单元的显示器或打印机上是无法直接表现的。将矢量图转换成以像素点阵来表示的信息，再加以显示或打印，这个过程称之为栅格化（Rasterization），而栅格化的逆过程相对比较困难。

1.2 本文的目的和意义

本文的研究目的与矢量化过程密切相关，针对矢量化处理过程中所包含的图象前处理、矢量跟踪、图形识别、图形后处理以及图形格式转换输出这五个环节，本文主要对图象的去噪、二值化、细化特别是我们想要建立合理的数学模型，尽量准确地提取出图案的边界线条，并将其用方程表示出来。

应用本文所设计的模型，可以最快速的精确提取图像的边缘曲线，并对其进行拟合，方便地进行矢量化，对于工程制图及提高图像的质量有明显的意义。

2.矢量化过程中存在的问题

目前，光栅图象输入识别与矢量化系统的研究与开发是工程及制图领域内的一个研究方向，属于一个较新的课题，这个方向在我国的研究还处于一个开始阶段。该课题的研究具有重要的实际意义和理论价值，同时又具有较大的难度，理论上的研究成果与实际的需求还有比较大的距离。光栅图象输入识别与矢量化系统的研究与开发，需要花费一定的时间去进行编写调试，由于时间和个人能力的限制，我们的模型不可能在短时间内完成整个矢量化系统的工作，只能就课题所涉及的几个主要方面加以研究。对于文中已得出的结论，目前也只适用于理论阶段，方程的准确性及应用领域还有待考察。

#1777

3.模型的建立

3.1 模型中涉及的字母

字母	意义
T	阈值
Ave	灰度均值
$Q(k)$	分离指标
h_k	灰度为 k 的像素个数
$Phs(i)$	灰度级的概率函数
(i, j)	像素
$Gray$	灰度
S_x, S_y	梯度

3.2 灰度图像的二值化

灰度图像中目标图像与背景的分割即图像二值化。扫描图主要是由目标和背景组成，当然可能还有一些扫描图纸时所掺杂的噪声。一般人们所关心的那部分图像物体为目标，而把物体之外的区域称为背景，在人类视觉中目标与背景有着明显不同的界限，人们可以很容易的将其分离，而计算机则必须通过一定的判定规则才能将其分离。

扫描图像大多数为 256 级灰度图像，像素的灰度值是从 0~255 之间，灰度值 0 表示黑色像素，为目标像素，灰度值 255 表示白色像素，为背景像素，灰度值在 0 — 255 之间的像素可能属于目标像素，也可能属于背景像素。一般采用阈值分割法对目标和背景进行分割，把目标像素标记为 1，把背景像素标记为 0。

3.2.1 阈值分割法简介

阈值分割法是一种区域分割技术，使用该方法分割灰度图像时一般对图像有一定的假设。换句话说，是基于一定的图像模型的。最常用的模型可描述如下：假设图像由具有单峰灰度分布的目标和背景组成，在目标或背景内部的相邻像素间的灰度值是高度相关的，但在目标和背景交界处两边的像素在灰度值上有很大的差别。如果一副图像满足这些条件，它的灰度直方图基本上可看作是由分别对应目标和背景的两个单峰直方图混合而成。此时如果这两个分布大小(数量)接近且均值相距足够远，而且均方差也足够小，则直方图应是双峰的。

#1777

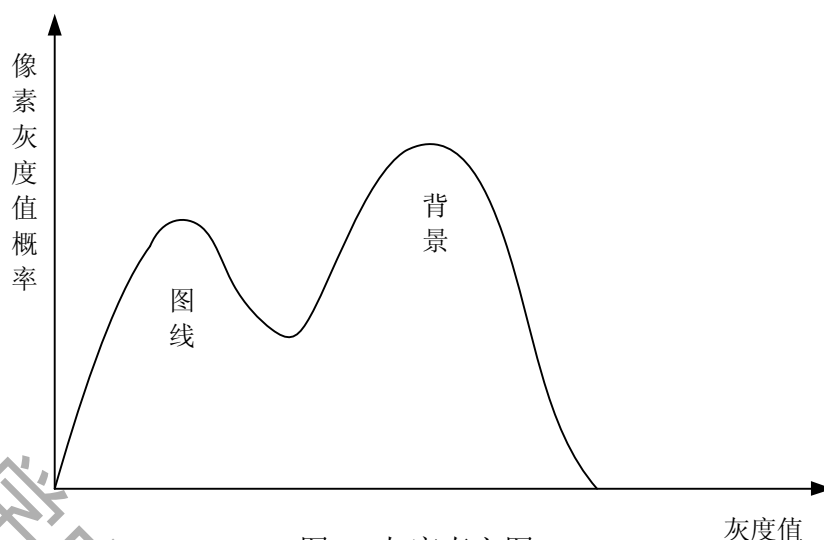


图3.1 灰度直方图

此类图像的灰度直方图曲线如图 3.1，其中横坐标轴表示图像中出现的灰度值，纵坐标轴表示图像中取某一灰度值出现的概率，即取该灰度值的像素占图像总像素的比例。当使用阈值方法来分割灰度图像时，首先确定一个灰度阈值 T ，然后将图像中每个像素的灰度值与阈值 T 相比较，所有灰度值小于或者等于阈值 T 的像素都被判属于目标，所有灰度值大于该阈值 T 的像素都被判断为背景。即

$$f(x, y) = \begin{cases} 1 & 0 \leq f(x, y) \leq T \\ 0 & T < f(x, y) \leq 255 \end{cases} \quad (3-1)$$

由上式可见，要从复杂的背景下把目标像素提取出来，阈值正确选取是关键。如果阈值选得过大，一些目标物将被误分为背景；如果取得太小，则会将一些背景转换成目标物，造成目标区的特征失真，形成错误分割。

3.2.2 计算阈值的方法

阈值计算的方法很多，常用的有整体阈值法、类判别分析法、迭代阈值法等。

1. 整体阈值法

整体阈值分割的具体方法是：首先计算整个图像的灰度直方图，然后获得整个图像的最佳分割阈值。整体阈值分割适合图纸质量比较好，整幅图像背景均匀，图线质量较好的情况。对于背景不均匀，背景与图线对比不一致的图像，不适合使用整体阈值分割的，可采用二次处理技术实现。首先，计算整图的灰度直方图，获得全局最佳阈值，进行第一次分割。然后将图像划分成 100×100 的像素块(根据图纸具体情况，可以调整分块大小)，在每一个块内部，按照整体阈值分割的方法获得最佳阈值。

2. 类判别分析法

该方法根据 Otsu 提出的类判别分析法寻找阈值[1]。计算阈值的具体步骤如下：

1) 计算输入图像的灰度级直方图(用灰度级的概率函数 $Phs(i)$ 来表示)。

2) 计算灰度均值(Ave)

$$Ave = \sum_{i=0}^{255} (i-1)Phs(i) \quad (3-2)$$

3) 计算灰度均值 $Aver(k)$ 以及直方图的和 $w(k)$

$$Aver(k) = \sum_{i=0}^k (i+1)Phs(i) \quad (3-3)$$

#1777

$$W(k) = \sum_{i=1}^k Phs(i) \quad (3-4)$$

4) 计算类分离指标

$$Q(k) = \frac{(Ave \times W(k) - Aver(k))^2}{W(k) \times (1 - W(k))} \quad (3-5)$$

5) 求使 Q 最大的 k 值。要计算的最佳阈值为： $T = k - 1$ 。

用该阈值得到的二值图像有较好的分离特性，并能尽可能多的图像的有用特征提取出来。

3. 迭代阈值

阈值也可以通过迭代计算得到，首先取图像灰度范围的中值作为初始阈值 T_0 (设共有 L 个灰度)，然后按下式进行迭代：

$$T_{i+1} = \frac{1}{2} \left\{ \frac{\sum_{k=0}^n h_k \times k}{\sum_{k=0}^n h_k} + \frac{\sum_{k=T_i+1}^{L-1} h_k \times k}{\sum_{k=T_i+1}^{L-1} h_k} \right\} \quad (3-6)$$

其中 h_k 是灰度为 k 的像素个数，迭代一直进行到 $T_{i+1} = T_i$ 结束，取结束时的 T_i 为分割阈值。迭代法不仅思路简单，实现容易，而且运行速度快。这种方法对于具有双峰直方图的图像效果很好，即使得多峰直方图的情况下也能获得合适的阈值。

3.2.3 最佳阈值的选择

目标像素与背景对比明显的图象，由于图线在图象整体中所占面积比较小，图线对应的峰面积较小，像素数量也较少；白色背景占据了图象中的绝大部分内容，面积较大。两个峰尖对应于目标及背景上的灰度值数量最多的点，两峰之间的谷对应于物体图线边缘附近相对较少数量的点。对于这种灰度变化均匀直方图曲线，一般采用整体阈值法。最简单做法就是在整个图象中将阈值设置为常数。只要选择了正确的阈值，使用一个固定的全局阈值一般会有较好的分割效果。

某些情况下，比如图象质量比较差，特别是噪声大，背景的灰度值并不恒定，而且图线和背景的对比如度在图象中变化比较大时，阈值就难以直接选取。因此，对于各种不同质量的图象，应有针对性地采用不同的处理方法，使二值化达到最理想的效果。根据图象的质量，一般可分为四类：

- (1) 清晰，即图形与背景分界明显且噪声小 (图 3.2(a))
- (2) 较为清晰，即图形与背景分界明显，但存在一定的噪声 (图 3.2(b))
- (3) 较为模糊，即图形与背景分界不很明显，也无较多的噪声 (图 3.2(c))
- (4) 模糊，即图形与背景分界不明显，且存在大量噪声数据 (图 3.2(d))

#1777

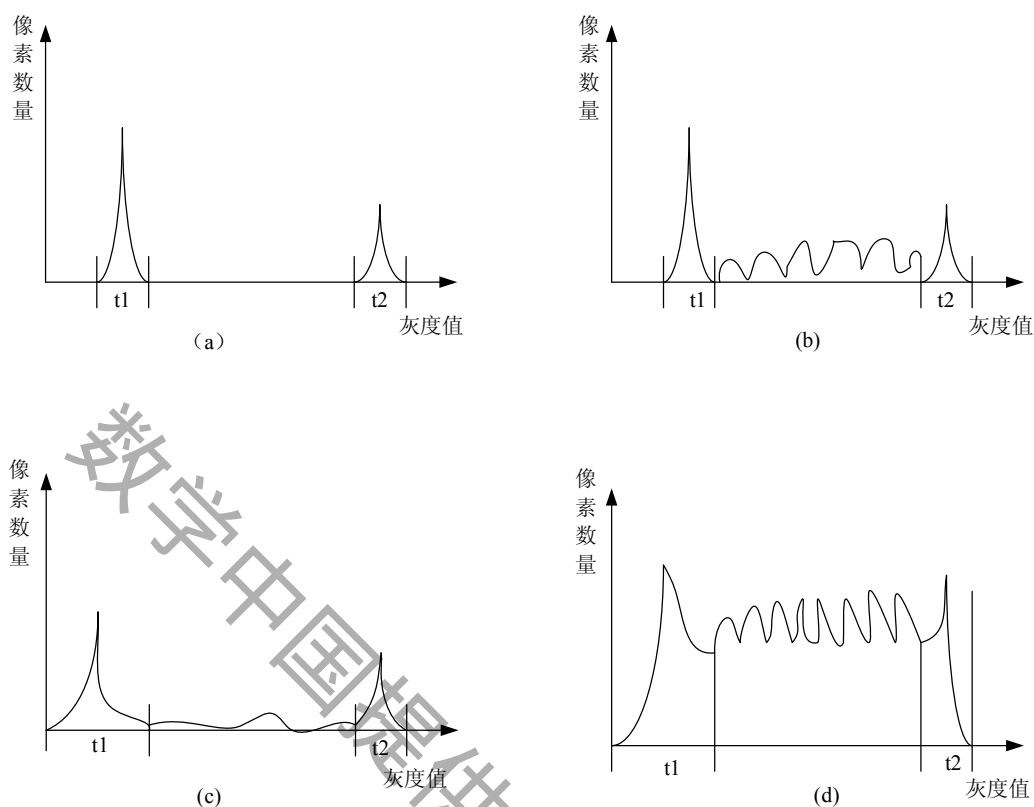


图 3.2 图像的分类

由于本文所处理的图像的噪声不多，属于图 3.2(a)的情况，因此采用整体阈值法进行处理。这种方法算法简单，运行速度快，也不会图像的某些部位产生伪影现象。具体方法是：首先考察整个图像的灰度直方图，寻找目标和背景的最高峰值，然后以两部分中的最高峰值的平均值对应的灰度值，作为整个图像的最佳分割阈值，对灰度图像进行分割，获得二值化图像。

由于白描图像中图线面积比较小，图线对应的波峰不明显。采用上述的整体阈值分割方法进行二值化处理，使用的阈值为 180，分割结果如图 3.4 所示。可以看出，绝大部分图线内容都得到了比较好的分割，背景区域比较干净，没有明显的噪声。

3.2.4 二值化结果

以下是实现灰度图像二值化的部分代码和结果：

```
Void CdfView::OnDraw(CDC*pDC)
{
    .....
    mybmp.Display2(pDC,0,0,widths,heights);
    color*pic=mybmp.BmpToPixel(mybmp.m-hBmP);
    if(pic)
    for(inti=0;i<heights;i++)
    for(intj=0;j<widths;j++)
    {
        if(!(pic[i*widths+j].Bb>180&&pic[i*widths+j].Bg>180&&pic [i*widths+j].Br>180))
            PDC ->SetPixel(j+500,i,00000000);
    }
}
}
```

#1777



图 3.3 灰度图像



图 3.4 二值化图像

3.3 二值图像的消噪

光栅图形一般是以像素或压缩的像素块为单位来表示图像，可以认为这种图形是由一个个的像素组成，彼此之间没有任何关系，也不考虑图像的内容，简单的说光栅图像就是一个点阵的概念。

矢量图采用的是完全不同的存储方式，它的存储单位不是单个像素或像素块，而是人们感兴趣的图像内容的特征。如在矢量图中要表示一个圆，只须存储圆的圆心、半径和颜色即可；对于一条直线，只须用直线的两端点坐标表示即可。

对于用矢量表示的图形而言，如果需要转化成光栅点阵的形式，由给出的图形参数描述，利用计算机图形学的基本方法，可以很方便的进行转化。但其逆过程，即从图形的光栅点阵形式转化成其参数的描述也就是采用矢量来表示的形式，就相当困难了。

本文的目的，就在于对这一逆过程进行探索，分析比较现有的几种矢量化算法，提出一种更实用、更高效的矢量化算法，将点阵图像转化为矢量图形。其大致流程如下图所示：

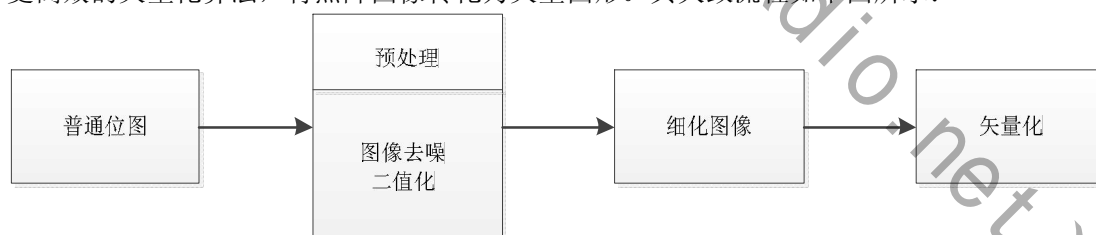


图 3.5 矢量化流程图

3.3.1 二值图像的消噪

噪声消除是矢量化预处理的一个重要环节。在二值化的过程中，由于所采用的整体阈值算法无法充分考虑图像的结构特征，二值化后的图像仍然会包括部分噪声，这些噪声会对后续的细化、矢量化等过程产生严重的影响，必须尽可能消除。

二值化后的图像中包含的主要噪声类型有[2]：

- (1) 孤立噪声：图纸背景中的孤立点、斑点、污点或者麻点。
- (2) 毛刺噪声：图纸中图线边缘不应该有的小毛刺。
- (3) 空洞、缺陷噪声：图纸中图线上出现的小洞或者边沿出现的缺口。
- (4) 断裂噪声：图纸中个别图线出现的轻微断裂。

下面将针对不同的噪声采用不同的方法消除[3]：

#1777

在介绍二值图消噪的方法之前,首先介绍一下二值图像的几个重要的定义。

(1) 4-邻域和 8-邻域

对于任一像素(i, j), 把包含该像素在内的一个集合称为像素(i, j)的邻域[4]。在栅格平面表示的数字图像中, 最常用的是“像素的 4-邻域”和“像素的 8-邻域”。

像素的 4-邻域(4-Neighbor): 对像素(i, j), 将它以及其上(i-1, j), 下(i+1, j), 左(i, j-1), 右(i, j+1)4 点构成的集合, 称为像素(i, j)的 4-邻域, 有时也称 4-近邻。

像素的 8-邻域(8-Neighbor): 对像素(i, j), 将它和其上(i-1, j), 下(i+1, j), 左(i, j-1), 右(i, j+1)4 个像素以及对角线的右上(i-1, j+1), 左上(i-1, j-1), 左下(i+1, j-1), 右下(i+1, j+1)4 个像素构成的集合, 称为像素(i, j)的 8-邻域(图 3.2), 有时也称 8-近邻。

x_8 (i-1,j-1)	x_2 (i-1,j)	x_4 (i-1,j+1)
x_4 (i,j-1)	x (i,j)	x_6 (i,j+1)
x_6 (i+1,j-1)	x_6 (i+1,j)	x_7 (i+1,j+1)

图 3.6 邻域图

(2) 连接数:

在某个“1”像素区中的连接数, 可以以这个像素的 8-邻域值 $f(x_0)$ 至 $f(x_7)$ (参照图 3.6) 按下式进行计算:

$$N_c = \sum_{k=0,2,4,6} [(1-f(x_k)) - (1-f(x_k))(1-f(x_{k+1}))(1-f(x_{k+2}))] \quad (3-7)$$

如果 $x_k = x_8$, 则令 $x_8 = x_0$

通过对 X 像素 8-邻域内一切可能存在的值, 按公式(3-1)进行计算, 其连接数总是取 0~4 之间的值。

(3) 像素间的距离

本次设计中, 像素间的距离采用阿基米德表示方法: 设像素 A, B 两点的坐标分别 (x_a, y_a) 和 (x_b, y_b) , 则其间的距离 d 可表示为:

$$|d|^2 = (x_b - x_a)^2 + (y_b - y_a)^2 \quad (3-8)$$

下面开始介绍具体的噪声的消除方法。

3.3.2 孤立点消除

目前, 孤立点的消除一般是采用 4-邻域清除方法或 8-邻域清除方法。如图 3.7 所示, 如果当前处理像素 $f(i, j)$ 为 1, 而其 4-邻域像素 $(i-1, j)$ 、 $(i+1, j)$ 、 $(i, j-1)$ 、 $(i, j+1)$ 均为 0 时。则将当前处理像素置为 0, X 为 0 或 1。在处理过程中发现, 4-邻域清除法虽然可以将孤立点清除, 但也有可能将交叉点、分支点或连接点一块除去。因此, 本文采用 8-邻域清除孤立点的方法。方法同 4-邻域清除方法类似, 如果当前处理像素 $f(i, j)$ 为 1, 判断它的 8-邻域像素上 $(i-1, j)$, 下 $(i+1, j)$, 左 $(i, j-1)$, 右 $(i, j+1)$, 右上 $(i-1, j+1)$, 左上 $(i-1, j-1)$, 左下 $(i+1, j-1)$, 右下 $(i+1, j+1)$ 。如果均为 0, 则将当前处理像素 $f(i, j)$ 置为 0。如图 3.8 所示。

X	0	X
0	1	0
X	0	X

→

X	0	X
0	0	0
X	0	X

#1777

图 3.7 4-邻域清除孤立点

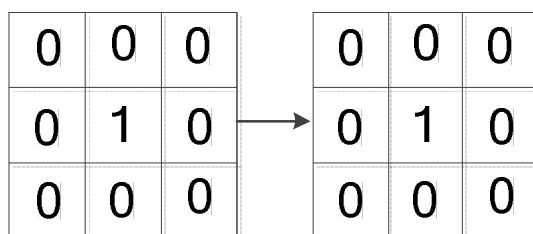


图 3.8 8-邻域清除孤立点

3.3.3 空穴的消除

空穴的消除与孤立点的消除正好相反，对应的消除方法也有 4-邻域和 8-邻域两种方法。如果当前处理像素 $f(i, j)$ 为 0，在 4-邻域消除空穴的算法中，只需判断 $f(i, j)$ 的四个邻点，而在 8-邻域消除空穴的算法中，则需要判断 $f(i, j)$ 的八个邻点。因此 4-邻域消除空穴的算法相对于 8-邻域算法的速度要快，所以本文采用了 4-邻域消除空穴的算法。

3.3.4 小面积颗粒的清除

利用 4-邻域法和 8-邻域法可以清除孤立像素点和空穴，但对于一些有若干个像素组成的孤立的小面积颗粒，则不能用这两种方法清除，可以用标记法求面积的方法除去。这里，面积指的是某一黑色区域中所有像素值为 1 的点的个数。

首先，对二值图按行、列进行扫描，将第一个扫描到的像素值为 1 的点进行标记。并且从标记的点开始，依次将与标记点相邻的像素值为 1 的点都作同样的标记，直到标记完所有与标记相连的为 1 的像素点。然后，计算出所有标记过的像素值为 1 的点的总数，即为该黑色块的面积。将该面积与预先设定的面积值相比较，如果大于给定面积值，则保留该块面积内的像素点。否则，就删除该块面积内的像素点。对整幅二值图作同样的处理，最后输出的二值图就是清除小面积颗粒后的图像，给定面积值可根据需要以及二值图中颗粒和目标的大小来设定。其具体操作可使用如下方式：在一次扫描图像的过程中即完成图像中连通区域的判定、区域像素点坐标的记录以及区域面积的计算。然后依据噪声面积阈值遍历图像中连通区域链表，将区域面积小于阈值的区域的像素点去除(置为背景色)。

在叙述图像中连通区域提取及面积计算算法之前说明一下定义的数据结构：定义了一个区域点坐标链表，表头结点记录区域标记号及区域面积值，其它结点记录区域内的点坐标值，再定义一个区域表头结点指针数组，用于记录图像中所有的连通区域信息。

C 语言描述如下：

```
typedef struct RegionPointNode{//区域点坐标结点
POINTpoint;//点坐标
Struet RegionPointNode*Nextpoint;//指向下一结点的指针
}*LPRegionPointNode,*RegionPointList;
TyPedef struet{
int RegionNo;//区域编号
int RegionArea;//区域面积
COLORREF RegionColor;//区域颜色
LPRegionPointNode next;
}*Region;
Region AllRegion[MAX];
```

假定二值图像目标像素为黑(下同)，且为了算法叙述的简洁，未考虑图像的边界问题，依据各边界特点作相应判断。

算法的主要步骤如下：

STEP1：初始化 AllRegion[MAX]，区域个数为 0；

STEP2：从上到下，从左到右扫描图像中的每个像素：

#1777

```
如果找到一个像素为黑且其左前!右上!正上!左上的点都不为黑，则{
新建一区域链表 RegionPointList 并初始化；区域个数增 1；
新建一坐标值结点 PointNode，赋值当前像素点坐标；
当前坐标值结点 PointNode 加入当前区域链表中；
修改区域链表头信息，区域编号为当前区域个数号，该区域面积置 1；
将区域表头指针加入图像区域数组中；}
```

```
否则{
```

```
根据其左前、右上、正上、左上的点的状态确定的巧种情况，决定当前点应该加入的区域链表或者
还需将两个链表合并；同时修改相应区域链表信息及图像区域数组信息。
```

```
}
```

以上算法即可求得图像中目标连通区域的个数，每一区域的面积以及该区域的像素点坐标。然后依据噪声面积阈值，遍历图像中的区域信息数组，将小于面积阈值的连通区域的像素点去除，即可达到去除噪声的目的。

3.3.5.自动去除斑点噪声的方法

自动去除噪声斑点的算法的基本思想是：先计算出图像中所有目标连通区域的面积，然后对区域面积进行排序,找出区域面积值出现突变的点,即可确定为噪声面积阈值。然后即可按上述方法进行去噪处理。例如，对一幅含有较多斑点噪声的线划图像进行了根据区域面积的去噪实验，通过交互式指定的某一较大斑点噪声面积为 37 个像素，系统自动确定的噪声面积阈值为 37 个像素，其区域面积及阈值如确定方法如图 3-5 所示，按此阈值对原始图像进行去斑点噪声后，可见在噪声面积与目标区域面积相差很大的情况下，本文采用的自动确定斑点噪声阈值的方法可行，可快速准确地消除输入图像的斑点噪声。而在噪声面积与部分目标线划面积相同的情况下，可适当减小噪声面积阈值，然后通过擦除工具擦除其它斑点噪声。

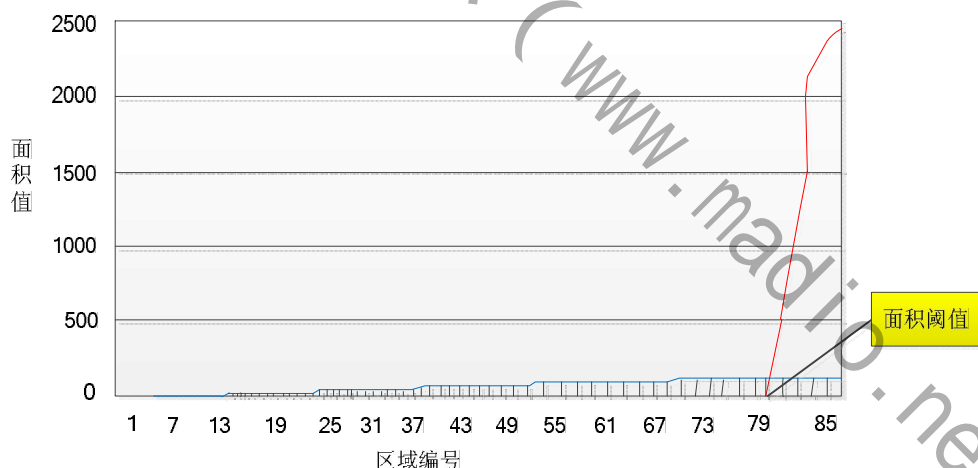


图 3-9 自动确定阈值示意图

3.4 边缘检测

边缘检测算法[5]作为图像处理领域和计算机视觉方面的一大重要技术，被广泛用来图像的识别，跟踪等。边缘检测的目的是标识数字图像中亮度变化明显的点，通过边缘检测可以大幅度的减少数据量，去除图像中可以认为不相关的信息,保留图像重要的结构属性。

依据检测算法的原理，边缘检测的方法可以分为一阶梯度检测和二阶梯度检测，常见的一阶梯测算子有 Canny 算子、Sobel 算子、罗盘算子、Prewitt 算子及 Roberts 算子等，常见的二阶梯度检测算子有 LOG、Marr-Hildreth 等。

由于 Canny 算子较好的兼容了图像的平滑操作，对平滑后的图像进行边缘检测时有较好的效果，

#1777

因此，本文基于 Canny 边缘检测算法的原理，提出了一种新的边缘检测的方法，该方法简单有效的提取出了图像的纹理信息。

首先，为了方便梯度信息的计算和使用，我们对图像进行灰度处理，考虑到实际应用，根据人眼的实际特点，我们选取：

$$Gray = 0.299R + 0.587G + 0.144B \quad (3-9)$$

接着，分别计算沿着 x、y 方向的梯度信息，梯度信息的计算使用一阶有限差分来进行近似，我们采用 Sobel 算子来进行梯度信息的计算，有卷积模板：

$$S_x = \begin{Bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{Bmatrix} \quad S_y = \begin{Bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{Bmatrix} \quad (3-10)$$

然后，对图像的梯度幅值进行非极大值抑制，图像梯度幅值矩阵中的元素值越大，说明图像中该点的梯度值越大，但这并不能说明该点就是边缘。在 Canny 算法中，非极大值抑制是进行边缘检测的重要步骤，通俗意义上是指寻找像素点局部最大值，将非极大值点所对应的灰度值置为 0，我们这里沿用了借鉴了 Canny 算法的思想，对灰度图进行了一个简单的非极大抑制中间处理。完成非极大抑制后，会得到一个二值图像，非边缘的点灰度值均为 0，可能为边的局部灰度极大值点可设置其灰度为 128。

最后，采用阈值法对图像的边缘信息进行连接，设定高低两个阈值，分别处理图像，这样就形成了一些边缘轮廓曲线。整个过程借鉴了 Canny 检测算法的优点，结合我们实际应用，相比 canny 检测，我们的方法一方面速度上更快，另一方面，由于采用较为精确的卷积模板进行梯度信息的计算，检测效果有较为明显的提高。

3.5 边缘跟踪

目标检测分离出来以后，就可以进行边缘跟踪。本设计中介绍用 8 方向 FREEMAN 链码对图像中的目标部分进行边缘跟踪。链码是对边界点的一种编码表示方法，其特点是利用一系列具有特定长度和方向的相连的直线段来表示目标的边界。因为每个线段的长度固定而且方向数目有限，所以只有边界的起点需要用(绝对)坐标表示，其余点都可只用接续方向来代表偏移量，最后通过起点的绝对坐标与各个偏移量计算出边缘中各点的绝对坐标。光栅图像一般是按固定间距的网格采集的，所以最简单的链码是跟踪边界并赋给每 2 个相邻像素的连线 1 个方向值。最常见的是 4 方向和 8 方向链码。它们的共同特点是线段的长度固定，方向数有限。

对于一个八连通的图像区域来说，由以上定义可知，区域中的每个像素，它的周围总存在 8 个像素与它连接，可以为这八个像素设定从 0 到 7 的方向编号，如图 3.10 所示。这样一旦确定了像素 P 的位置，以及某个邻接像素的编码，就可以知道邻接像素的位置。链码实质上是一串指向符的序列。链码表示就是从某点开始观察领域的指向来表示，结果形成一个序列，因此可以用链码来描述任意曲线或闭合的边界。

3	2	1
4	P	0
5	6	7

图 3.10 链码的 8 邻域方向

用链码对图像中的目标进行跟踪的过程如下[6]：对于整个图像进行从上到下、从左到右的扫描，如果发现了一个作为前景颜色的像素(黑点)，则对和这一点相关的图形进行外边缘的跟踪，跟踪完毕

#1777

后擦除掉跟踪得到的外边缘内部的所有黑像素，再对图像进行扫描跟踪，直到整个图像中已经不再有黑色像素。对于每个特定目标进行外边缘跟踪的过程如下：从第一个边界点开始，定义起始点的初始搜索方向为左上方；如果左上方的点是黑点，则为边界点，否则搜索方向顺时针旋转 45 度。直到找到第一个黑点为止，这个黑点就是新的边界点。新边界点的初始搜索方向为前一边界点到该点方向的反方向并顺时针旋转 45 度，用同样的方法搜索下一点，直到返回最初的边界点为止[7]。其算法实现如下：

```

Do
{
    n=(n+4)%8; //回到起始方向
do
{
    n=(n+7)%8; //turn arounds 顺时针旋转 45 度
    NewPoint=CutPoint 点在 n 方向上的点;
}while(NewPoint 不为目标点)
if(NewPoint=StartPoint)
}
本次跟踪结束;
}
else
{
    CutPoint=NewPoint
}while(l)

```

其中：n 表示当前方向数，CutPoint 表示当前点，NewPoint 表示在当前点的 n 方向上的点，StartPoint 表示起始点。

3.6 边缘曲线的分段拟合

图像分割以后，鞋样的边缘轮廓被表示成一个个分散的像素点，数据量非常大，离需要的图元对象还差很远，不利于进行存储和图形编辑。因此，通常的处理方式是用一系列首尾相连的曲线来拟合这些点序列，以达到最终矢量化的目的。轮廓曲线通常都采用内插曲线或者逼近曲线来实现拟合。在本系统里，一个数字化的鞋样边缘除了数据量大和存在毛刺以外，还具有以下三个特征：

- (1)存在曲率变化较缓慢的曲线段；
- (2)存在尖点；
- (3)鞋样边缘是闭合的；

综合以上特点，本文提出了一种分段拟合的方法对鞋样边缘进行矢量化。该方法主要分为两大步骤：

步骤一：尖点提取，即通过曲率方法提取出轮廓中的尖点，防止在下一步的拟合中将尖点平滑掉，曲线特征丢失。

步骤二：曲线拟合，即以尖点为分段点，对每两个尖点之间的曲线段采用样条曲线进行递归拟合，直到精度满足一定条件为止。

以下两节对这两个步骤分别进行详细介绍[8]。

3.6.1 尖点提取

尖点对应曲线中具有局部曲率极大值的点，即曲线的斜率变化发生突变的点。如图 3.11 中的以 ‘*’ 标示的点即为一个尖点。尖点一般位于两段光滑曲线的分段处，在拟合时无法用光滑的曲线表示出来，因此通常先提取出边缘中的尖点，然后把两个尖点之间的曲线作为一个拟合段。

#1777

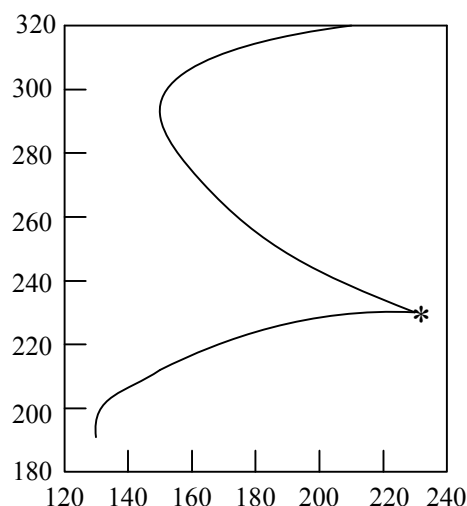
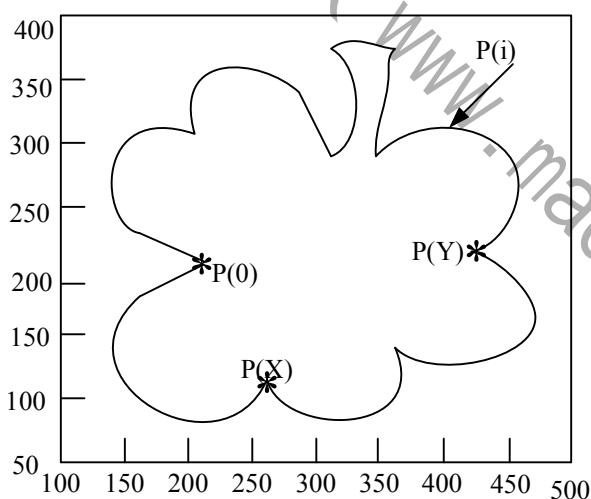


图 3.11 尖点示意图

在数字曲线中估算曲率中余弦法和弦长比方法都是直接利用曲线上点的坐标来计算曲率，计算比较复杂。本文针对鞋样边缘闭合的特点，提出了一种间接的算法，该算法相对以上提到的各种算法计算量要小、算法简单，并且适合任意形状的曲线，能快速准确的提取出复杂轮廓上的尖点。

3.6.2 算法思想

设 $P(i) = [X(i), Y(i)] (i = 0, 1, 2, \dots, n-1)$ 为一闭合的图像边缘轮廓点序列(见图 3.12)，用 ‘*’ 标示的 $P(0)$ 点为该序列的起始点，随着 i 的增大， $P(i)$ 沿逆时针方向行进一周后又回到 $P(0)$ 点。由于在闭合曲线上， x 和 y 之间是非函数关系，因此，函数中曲率的计算公式不能直接用在此处。

图 3.12 闭合图形轮廓点序列 $P(i)$

在图 3.12 上可以看到， $P(i)$ 点的坐标由 $X(i)$ 和 $Y(i)$ 共同决定，因此，可将 $P(i)$ 分解为两条一维离散曲线 $X(i)$ ， $Y(i)$ ，(如图 3.13)， $X(i)$ 随 i 的变化过程即为 $p(i)$ 在水平方向上的变化， $Y(i)$ 则为 $P(i)$ 在竖直方向上的变化，因此，只要分别找出曲线 $X(i)$ ， $Y(i)$ 上的尖点，综合起来，就能得到 $P(i)$ 上的尖点集合。例如在图 3.12、图 3.13 中，由点坐标的变化趋势可以看出，曲线 $X(i)$ 上的尖点 P_X 和曲线 $Y(i)$ 上的尖点 P_Y (‘*’ 标示)，分别对应曲线 $P(i)$ 上的尖点 P_X 和 P_Y (‘*’ 标示)。由于 $X(i)$ ， $Y(i)$ 都是关于 i 的一元离散函数，因此曲率的估算简单明了。

3.6.3 算法设计

根据以上的曲率近似值即 q 值求曲线尖点的算法思想如下：以相邻且其 q 值全大于 0 的点序列作为一个局部曲线段，找出该段曲线中 q 值最大的点，若该点的 q 值大于某一阈值 T ，则将该点标

#1777

记为一个尖点，否则则为非尖点。具体设计如下，其中 \max 用来记录当前段中的最大 q 值， k 用来记录当前段中具有最大 q 值的点的序列号， T 为阈值：

(1) 求曲线 $X(i)$ 上的尖点

计算 $X(i)$ 上某一点对应的 $q(i)$ 值并进行判断；

```

if  $q(i) > 0$ 
{
    if  $q(i) \geq \max$ 
    {  $\max = q(i)$ ;  $k = i$ ; 计算  $q(i+1)$  并重新开始判断; }
    else
    {
        计算  $q(i+1)$  并重新开始判断;
    }
}
else
{
    if  $\max > T$ 
    { 将第  $k$  点标记为尖点;  $\max = 0$ ; 计算  $q(i+1)$  并重新开始判断; }
    else
    {  $\max = 0$ ; 计算  $q(i+1)$  并重新开始判断; }
}

```

(2) 求曲线 $Y(i)$ 上的尖点具体过程与求 $X(i)$ 上的尖点相同。

(3) 综合 $X(i)$ 与 $Y(i)$ 上的尖点，求原曲线 $P(i)$ 上的尖点集。

对于 $X(i)$ 上的任意特征点 P_X 和 $Y(i)$ 上的任意尖点 P_Y ，设其点的序列号分别为 j, k ，

若 $|j - k| \leq 3$ ，则 P_X 和 P_Y 对应 $P(i)$ 上的同一个尖点。

若 $|j - k| > 3$ ，则 P_X 和 P_Y 对应 $P(i)$ 上两个不同的尖点。

3.6.4 实验结果及分析

以下图是用 MATLAB[9] 语言实现该算法并对图 3.12 中的边缘提取尖点得到的结果，(图中 ‘*’ 代表尖点)。图 3.14(a) 是在曲线 $X(i)$ 上提取的尖点的情况，图 3.14(b) 是 $X(i)$ 中的尖点在原图中的对应，其中原图中的 $P_X(1)$ 点对应于 $X(i)$ 中的第一个尖点，其后的以逆时针方向与 $X(i)$ 中的尖点一一对应。图 3.15(a) 与图 3.15(b) 则是 $Y(i)$ 中的尖点提取及其在原图中的对应情况。

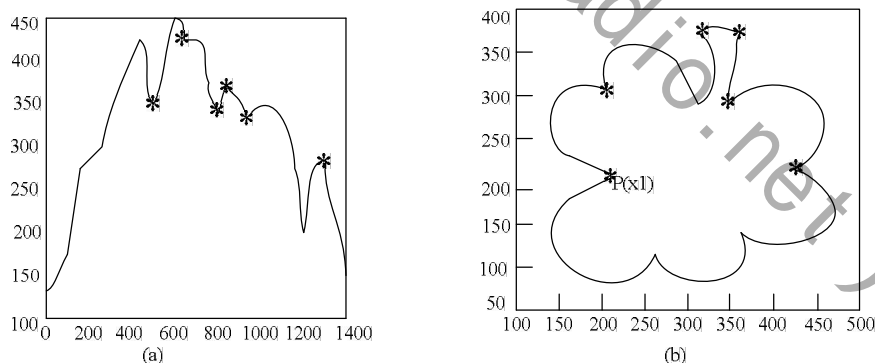


图 3.14 曲线 $X(i)$ 上的尖点及其在原理中的应用

#1777

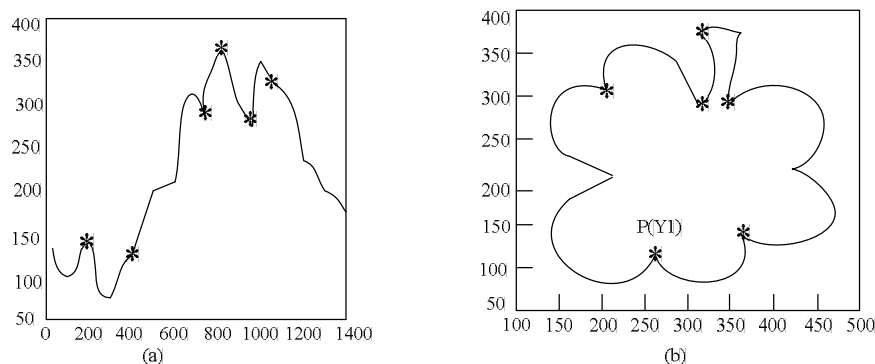
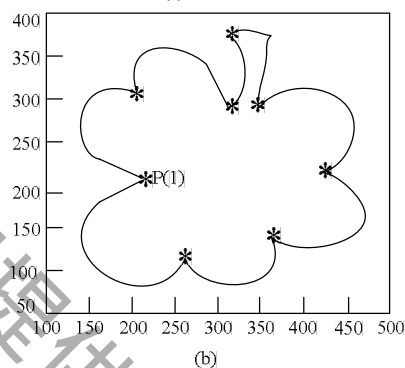
图 3.15 曲线 $Y(i)$ 上的尖点及其原理图

图 3.16 综合两个方向上的尖点后的结果

从以上结果可以看出该算法对尖点的定位是比较准确的，且适合各种形状。在算法效率上，本算法在两个方向上点的曲率时，分别只用到了三次加减法运算，与其他方法比较，计算量大大减小，效率高，以下是该算法在计算某点的曲率时其计算量与余弦法和弦长比法的一个比较：

表 3.1 各种曲率算方法计算量的比较

方法 \ 操作	加减	乘除	开方运算
余弦法	7	7	2
弦长比法	6	7	0
本方法	6	0	0

本节中提出的间接方法实际上是一种基于参数方程的方法，在该方法中我们引入了一个新的参数，即点序列号 i ，利用该参数将点序列 $P(i)$ 的坐标变为参数方程 $X(i)$, $Y(i)$ 的方法来表示

$$\begin{cases} X(i) = X_i \\ Y(i) = Y_i \end{cases} \quad i = 0, 1, 2, \dots, n; \quad (3-11)$$

其中 $X(i)$ 和 $Y(i)$ 是关于 i 的一维离散函数，因此可以对其用函数的方法来处理。

3.6.5 曲线拟合

1. 抛物样条曲线

通过不在同一直线上的三点： P_1, P_2, P_3 ，定义一条抛物样条曲线的表达形式如下：

$$P(t) = A_1 + A_2 t + A_3 t^2 \quad 0 \leq t \leq 1; \quad (3-12)$$

该抛物样条曲线过 P_1, P_2, P_3 三个点，并且有：

(1) 抛物线段以 P_1 为起始点。即当参变量 $t=0$ 时，曲线过 P_1 点。

#1777

(2)抛物线段以 P_3 点为终点，即当参变量 $t=1$ 时，曲线过 P_3 点。

(3)当参变量 $t=0.5$ 时，曲线过 P_2 点,且切矢量等于 $P_3 - P_1$ 。

由以上三个条件得到方程组如下：

$$\begin{cases} A_1 = P_1 \\ A_1 + A_2 + A_3 = P_3 \\ A_1 + A_2 + A_3 = P_2 \end{cases} \quad (3-13)$$

解得：

$$\begin{cases} A_1 = P_1 \\ A_2 = 4P_2 - P_3 - 3P_1 \\ A_3 = 2P_1 + 2P_3 - 4P_2 \end{cases} \quad (3-14)$$

将式(3-14)代入式(3-12)得：

$$P(t) = (2t^2 - 3t + 1)P_1 + (4t - 4t^2)P_2 + (2t^2 - t)P_3$$

由式(3-14)可得，当 P_1, P_2, P_3 在同一条直线上时，即 $2P_2 = P_1 + P_3$ 时， $A_3 = 0$ ，曲线方程变成一条直线。因此在本文中的拟合基元实际上是直线和抛物样条曲线。 $P(t)$ 是一个点向量，在二维平面上它包含了两个坐标值 $[x(t), y(t)]$ 。对该条抛物线进行存储和编辑时，只需对 P_1 、 P_2 和 P_3 进行操作。无需像数字曲线那样对每一个点都进行操作。在屏幕上显示该条曲线时，可以根据参变量 t 的取值，计算出位于曲线上的数据点，然后顺次连线绘出图形。

2.拟合算法--递归算法

本文提出了一种递归的拟合方法对各个尖点之间的曲线段分别进行拟合。该算法思想如下：以当前离散曲线段的起点、中点和末点构造一条抛物样条曲线对当前曲线段进行拟合并进行误差计算，若本段拟合的误差小于给定的阈值，则返回；否则以当前曲线段的中点将曲线分为两部分再用同样的方法分别进行拟合，算法设计如下，其中：

first 表示当前曲线段的起始点。

last 表示当前曲线段的终点。

T 为误差阈值，单位为像素，可以根据不同需要进行设定

Recursive (int first,int last)

bew=first 和 last 的中点；

构造依次过 first,bew 和 last 三点的抛物样条曲线；

Len=当前段的拟合误差；

if(Len>=T)

{

Recursive(first,bew);

Recursive(bew,last);

}

else

{返回;}

}

用这种方法依次拟合各尖点之间的曲线段，就能得到总的拟合结果。可以看出该算法对每段曲线只需用两个简单的递归语句就能实现拟合；相对于遗传算法中各种复杂的遗传、变异及交叉等操作，非常简单且易于实现。

3.算法分析及试验结果

由前面的分析可知，评价一个矢量化算法的根本标准是其精确性和其数据压缩率。以下就从这两个方面对递归算法进行分析：

(1) .精确性

递归算法中递归的中止条件是某段曲线的拟合误差小于误差阈值 T，若条件不满足，则可以对曲线进行进一步的细分，以进行更精确的拟合，直到满足精度条件为止。由于 T 可以根据实际需要进

#1777

行设定，因此理论上递归算法能达到任意大小的拟合精度。但精度越大，递归次数越多，计算量也越大。

(2) 数据压缩率

拟合后的鞋样边缘可以表示为一系列首尾相连的抛物样条曲线。由前面的分析可知一条抛物样条曲线可以由不在同一直线上的三个点决定，因此对它们进行存储和编辑时无需直接操作曲线，只需存储或编辑各段曲线的分界点及其在原始曲线上的中点，在最后显示或切割时再用这些分界点及中点作为参数恢复出各条曲线。因此本算法的压缩率就是原始点序列个数与分界点及中点个数的比。比率越大，则算法的压缩效率越大。在递归算法中，最后得到的分界点的数量也与误差阈值 T 的选取有关， T 越小，拟合的精度越高，则对原曲线的分段就越细，分界点数量越多，压缩率就越小。同样， T 越大，则压缩率也越大。因此实际应用中我们应当根据需求选择合适的阈值，这样可在满足精度的条件下减小递归次数，避免不必要的计算，同时还可以提高数据压缩率。



图 3.17 拟合后的图像

以下各组图片显示了用该算法对各种形状进行拟合，在误差阈值 T 分别取 5 和 12 的时候得到的结果(图中线条有分叉的地方表示存在一定的拟合误差，无分叉的地方则表示拟合的精度较高，圆圈表示各段抛物样条曲线间的分界点)。

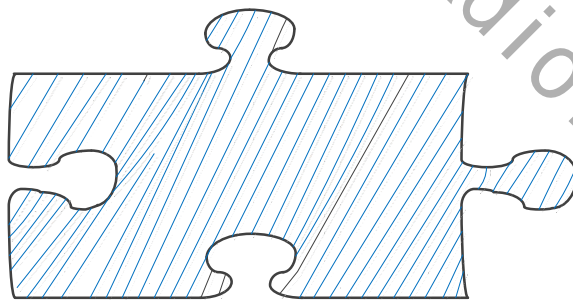


图3.18 原始形状1

#1777

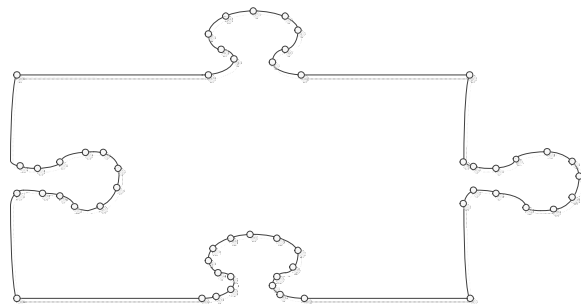


图3.19 T=5时的拟合情况及分界点

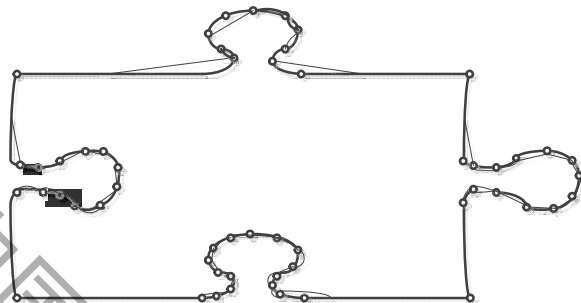


图3.20 T=12时曲线的拟合情况及分界点

4.总结及展望

4.1 总结

图像矢量化技术独有的缩放不失真性，存储空间少等特点使其越来越受到研究人员的青睐，特别是被广泛应用于移动终端设备。不管是用来作为图像处理的一种技术手段还是用来作为海量图像数据的一种压缩方式，都有非常长远的应用前景。本文针对题目要求，对栅格化的逆过程——即矢量化中的快速检测边缘建立了相应的数学模型，并给出了其方程。

模型中涉及的主要问题及解决方案简单列举如下：

- 针对题目要求，运用边缘检测算法--Canny 算子以最快的速度对边缘进行灰度处理、卷积处理、阈值算法，得到边缘轮廓曲线，并用递归算法对其进行拟合，总体上达到题目要求。
- 在完成题目要求的前提下，对矢量化涉及的消噪、二值化等问题进行解释，以加深对模型应用的理解。
- 对涉及的有关概念、专业术语也有相应的简单解释，方便初学者对于矢量化过程的学习。

模型可以适用于大多情况，在出现严重外界干扰的非正常状况，模型会出现较大偏差，对于其他未来相对稳定的矢量化过程均可作为参考。另外，模型还可以推广应用于其他工业领域，而非单纯用于理论阶段。

由于时间有限及作者水平限制，本文提出的模型在构造上有不尽如人意之处，在理论的丰富及模型的完善上有进一步探索的空间。

4.2 展望

◆构建模型实现的功能

#1777

本文针对矢量化过程中的主要步骤——边缘检测给出其方程，期望高效实现提取图像的边缘曲线，将其快速检测、提取、拟合，实现图像的处理，提高图像显示效果，并可以对工程应用有一定的指导意义。

◆对模型的展望

光栅图像的矢量化系统的开发与研究是一个相当复杂和具有一定难度的研究课题，开发比较实用完善的光栅图像输入与自动识别软件需要很大的工作量和丰富的图象处理及模式识别等方面的知识，需要编写、调试和测试大量的程序代码。因此本文所描述的算法仍不可避免的存在很多缺陷，对后续的展望主要表现在以下几个方面：

◆特征曲线的提取

特征曲线是矢量化编辑和渲染的重要依据，也是整个过程的开始，本文的特征曲线提取算法涉及大量的图像处理操作。我们知道，图像的处理次数越多，丢失的信息就越严重，引入的外界干扰就越大。然而，图像处理的次数少了，提出的特征信息就不够准确，因此选择合适的处理力度是后续需要关心的一个问题。

◆模型中涉及的光栅矢量化提取图像较为单调，由此生成的可视图，必然没有丰富、逼真的色彩和内容。后续可以考虑加入纹理合成技术，以得到更加逼真的图案。

◆对于检测的速度及人际界面的设计都存在一定得可提升空间，在后续实际应用及模型的改进都可以考虑。

参考文献

- [1]吴一全,朱兆达.图像处理中阈值选取方法 30 年的进展[J],数据采集与处理.1993.8:268-281.
- [2]Kenneth R. Castle~著.朱志刚石定机等译.数字图像处理[M].北京:电子工业出版社,2002,368-370.
- [3]陈金都,鹿凯宁,丁润涛.工程图纸噪声滤除的数学形态方法[J],电子测量与仪器学报.1996.10:23-27.
- [4]靖稳峰.应用数学形态学理论提高扫描二值图像质量的方法[J].西安工业学院学报.2000.1:31-33
- [5]王植,贺赛先.一种基于 Canny 理论的自适应边缘检测方法[J].中国图形学学报,2004,9(8):957-961.
- [6]Ray N., Cavin X., Levy B. 2005. Vector texture maps on the GPU. Technical Report ALICE-TR-05-003.
- [7]L. Demaret, N. Dyn and A. Iske, .Image Compression by Linear Splines over Adaptive Triangulations.Signal Processing, vol. 86,no. 7,pp. 1604-1616,Jul 2006.
- [8]X. Hilaire and K. Tombre, .Robust and Accurate Vectorization of Line Drawings.Proc. IEEE Trans. Pattern Anal. Mach. Intell, 2006,26(6): 890-904.
- [9]孙兆林.MATLAB.6x 图像处理.清华大学出版社.2002,266-278