

## 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，我们将受到严肃处理。

我们允许数学中国网站([www.madio.net](http://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：1519

参赛队员（签名）：

队员 1： 朱勇

队员 2： 胡学峰

队员 3： 刘雅倩

参赛队教练员（签名）：杨桂元

参赛队伍组别：本科组

## 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）：1519

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）：

# 2014 年第七届“认证杯”数学中国 数学建模网络挑战赛

## 题 目 位图矢量化的模型与推广

关 键 词 位图，贝塞尔曲线，双线性插值，误差分析，极限替代法，

matlab

### 摘 要

本文针对简单图案的边界线条提取问题，以题中所给的黑白图案为切入点，综合分析了图案轮廓特征点的坐标值数据，分别构建了基于贝塞尔曲线的图形边界线条提取模型，图形边界线条提取修正模型和简单彩色图形边界线条提取模型，运用 Excel、Photoshop、Matlab 软件提取出了图形的边界线条，并用 Opencv 软件对拟合的线条进行了检验。最后，将模型结果和实际相结合，对模型进行了进一步的改进和推广。

针对问题一，首先，运用 Photoshop 软件对图像进行噪点去除；其次，将图像切割成四部分，对每个部分的光滑区域进行一维插值，非光滑区域采取双线性插值；最后，运用了贝塞尔曲线的原理建立了模型 I——基于贝塞尔曲线的图形边界线条提取模型。

针对问题二，首先，在对模型 I 修正的基础上建立了模型 II——图形边界线条提取修正模型；其次，对于被处理图像的直线区域采取长度约束的分段法和坐标微调法进行数据的插值和剔除，同时为了提高图像的拟合优度，在进行轮廓点切割时，依据转折点切割原则，将原轮廓点切割成六部分，并分别拟合，最后，用方程将图案的边界线条准确表示出来，且运用了极限替代法进行了拟合优度的检验。

针对问题三，我们建立了模型 III——简单彩色图形边界线条提取模型，对简单彩色图形进行边界线条提取。首先，我们选择了一个色调分布和形状相对简单的彩色图形，并利用 Matlab 软件将该彩色图像分离出 R, G, B 三个通道的伪灰度图像，并采取中值滤波法对图像进行了滤波处理，剔除噪点，其次，对三个图像分别进行边缘检测，提取 R, G, B 三个通道的边缘线条，最后，采取边缘轮廓特征点并点法和非边缘点剔除法将三个通道的边缘线条信息融合，从而得出最终的彩色图像边界线条。采用 opencv 软件对边界线条提取结果进行检验。

本文还对模型的误差进行了定性分析；对模型的优化提出了针对性的切实的改进方向，对问题模型作出了理性中肯的评价。最后，我们又对空间图形的轮廓提取提出了初步想法，把模型从黑白图像推广至彩色图像进而到空间图像的边界线条的处理。本文分析思路清晰，切入点独到，运用多种插值方法，分析全面，特色鲜明。

参赛队号 1519

所选题目 B

参赛密码 \_\_\_\_\_  
(由组委会填写)

## Abstract

This paper is aimed at extracting the boundary line of simple graphics. we select black and white pattern as the breakthrough point, and analyze the coordinates point data of the outline feature comprehensively. build the Bessel graphic black and white border line extraction model, boundary line extraction correction model and simple graphical color graphic boundary line extraction model, by use of Excel, Photoshop, Matlab software to extract the boundary line of the graphics, and Opencv software for the inspection of the line fitting. Finally, we combine the model results with reality.

For the first question, firstly, using Photoshop software to remove image noise in interference; Secondly, cut the image into four parts, do one dimensional interpolation to the smooth part, do bilinear interpolation to nonsmooth region; Finally, establish the model - graphical boundary line extraction model based on bezier curve.

For the second question, first of all, model II is based on the model One- graphic boundary line extraction correction model. Secondly, as for the linear area, we use length limited piecewise method and Coordinate fine-tuning method. In order to improve the goodness-of-fit of the image, we use the turning point cutting principle to divide the image into six parts, and fitting respectively, and finally, use equation to simulate the boundary line of accurate representation, and use the ultimate alternative method for the test of goodness of fit.

For the third question, we build a model III - simple color graphic boundary line extraction model. First, we chose a simple color figure, whose color distribution and shape are simple, and we use Matlab software to isolate the three gray images of R, G, B channel. We take the median filtering method to filter image and eliminate noise. Secondly, we do the edge detection for the three images and extract the three edge lines of R, G, B channel. Finally, taking different methods according to different edge distribution, and we get the final boundary line of the color image. Last but not least, we use the Opencv software to test the results.

This article also carry on the qualitative analysis to the errors. And we put forward the improvement direction about the optimization, made rational evaluation to the problems. Finally, we put forward the model from the black and white images to color images and then to the space pattern. This article is clarity and original, and using a variety of interpolation methods.

## § 1 问题的重述

### 一、背景知识

#### 1. 位图

位图图像，亦称为点阵图像或绘制图像，是由称作像素（图片元素）的单个点组成的。这些点可以进行不同的排列和染色以构成图样。当放大位图时，可以看见赖以构成整个图像的无数单个方块。扩大位图尺寸的效果是增大单个像素，从而使线条和形状显得参差不齐。然而，如果从稍远的位置观看它，位图图像的颜色和形状又显得是连续的。

#### 2. 矢量图

矢量图是根据几何特性来绘制图形，矢量可以是一个点或一条线，矢量图只能靠软件生成，文件占用内在空间较小，因为这种类型的图像文件包含独立的分离图像，可以自由无限制的重新组合。它的特点是放大后图像不会失真，和分辨率无关，适用于图形设计、文字设计和一些标志设计、版式设计等。

#### 3. 位图的处理

这里位图的处理主要考虑位图的矢量化。矢量化是数字图像处理中的一个重要问题，是一个综合了计算机视觉，计算机图像处理，计算机图形学和人工智能等各个学科的交叉课题。目前位图矢量化的处理软件有：Flash、Ai、Coreldraw、AutoCAD，这里我们考虑用算法来实现位图矢量化。

#### 4. 栅格化及栅格化逆过程

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）。栅格化的逆过程这里即为将位图处理为矢量图。

#### 5. 灰度图像

灰度图像是每个像素只有一个采样颜色的图像，灰度图像是一种具有从黑到白 256 级灰度色域或等级的单色图像。图像颜色由 RGB(R, G, B) 三个通道组成，那么一个图像可以分割成三个灰度图像，其中每个图像的每个点的 R, G, B 值是相等的，红绿蓝三个通道的缩览图都是以灰度显示的，并且不同的灰度色阶表示红绿蓝三色在图像中的比重。通道中的纯白，代表了该色光在此处为最高亮度，亮度级别为 255，黑色的亮度级别为 0。灰度图像的表现形式为黑白图像，只有黑白两色。

#### 6. 位图的矢量化处理优点及应用

位图矢量化处理有很多优点：首先，矢量图像由简单的几何图元组成，表示紧凑，所占存储空间小；其次，矢量图像易于进行编辑，对矢量图像进行编辑的时候，如进行旋转、拉伸、平移等操作时仅需要修改相应几何图元的参数信息；第三，用矢量表示的对象易于放大或者压缩，而且不会降低其在计算机中的显示质量，矢量图像的放缩能够保持边角的尖锐等特性，不会出现模糊影响显示质量。位图矢量化的应用：气象传真图、CAXA 数控线切割 XP、激光雕刻技术、电子地图等等。

### 二、相关数据

1. 题中轮廓特征点坐标值（详见附录表[1]）；

### 三、要解决的问题

1. 问题一：对题目中给出的已经进行栅格化的位图（见图一）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。

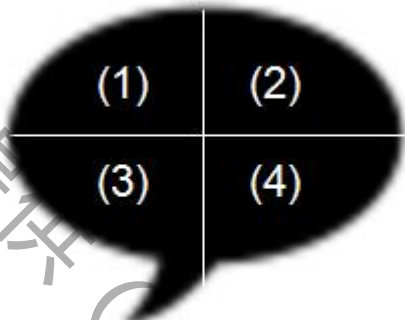


图一 已栅格化图

2. **问题二：**由对问题一中已经建立的模型进行改进和推广
3. **问题三：**简单彩色图像的轮廓提取方法

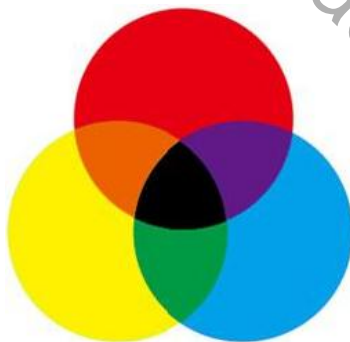
## § 2 模型的假设

1. 本文建立的模型针对形状较为简单的图标(这里我们将简单图形定义为转折点数量较少且颜色分布简单的图形)
2. 我们假设将图形分成四个部分：(1), (2), (3), (4)



图二 图形切割图

3. 假设所获取的图片图像水平清晰度较高，噪点影响程度低，图像本身质量较高。
4. 由于题目没有提供彩色图像，这里我们假设以下图(文档形式看不出图像颜色，请看电子版)作为处理对象。



图三 彩色图

5. 所有数据来源准确、可信、稳定、科学。

## § 3 名词解释与符号说明

### 一、名词解释

1. **贝塞尔曲线：**贝塞尔曲线(Bézier curve)，又称贝兹曲线或贝济埃曲线，是应用于二维图形应用程序的数学曲线。一般的矢量图形软件通过它来精确画出曲线，贝兹曲线由线段与节点组成，节点是可拖动的支点，线段像可伸缩的皮筋。

**2. 双线性插值：**双线性插值，又称为双线性内插。在数学上，双线性插值是有两个变量的插值函数的线性插值扩展，其核心思想是在两个方向分别进行一次线性插值。

**3. 噪点：**主要是指图像中的粗糙部分，也指图像中不该出现的外来像素，通常由电子干扰产生。如果将原图像放大，那么就会出现本来没有的颜色(假色)，这种假色就是图像噪音。

## 二、符号说明

序号	符号	符号说明
1	$t$	时间变量，又叫参数方程的变量
2	$B(t)$	$t$ 时间内关于坐标 $(x, y)$ 的参数方程
3	$(x_i, y_i)$	原始数据的坐标点 $(i=0,1,2,3\cdots)$
4	$(\hat{x}_i, \hat{y}_i)$	拟合过后的坐标点 $(0,1,2,3\cdots)$
5	$X(t)$	贝塞尔曲线拟合后对应的 $x$ 点参数方程
6	$Y(t)$	贝塞尔曲线拟合后对应的 $y$ 点参数方程
7	$R^2$	模型拟合后检查方程拟合程度的检验数
8	$\eta$	某一坐标对应其他多个坐标的点的微调程度
9	$\alpha_i$	三维坐标轴中用于垂直截取某一坐标轴的平面

## § 5 模型的建立与求解

### 一、问题一的分析与求解

#### 1.对问题的分析

将问题拆解成两部分：第一，图形边缘线的提取，第二，对提取的边缘线矢量化。首先，通过 matlab 对题中已给定的图形进行轮廓特征点提取，得到轮廓特征点的坐标位置之后，对这些特征点的位置进行插值并拟合，其次，考虑到插值之后的图像不够圆滑，为了减小误差，将图像等分切割为四部分，然后利用贝塞尔曲线的思想，对已插值后的轮廓特征点分别进行贝塞尔曲线的曲线矢量化，从而得到四个部分矢量化轮廓曲线图，最后将四部分的曲线图结合起来形成一个整体轮廓图。

#### 2. 问题的求解

##### 模型 I 基于贝塞尔曲线的图形边界线条提取模型

##### (1) 模型的准备

①去除噪点干扰：由于对于扫描后的图像，其经过栅格化之后图片上存在许多噪点，在矢量化的预处理阶段，必须先将其剔除，以消除对矢量化的干扰。在一定的尺寸范围内，噪声、文字符号和一些较小的图形元素都是孤立图元，在初步提取工作完成后，还须采取一些必要步骤将它们有效地区分开。这里我们利用 photoshop 对图片进行噪点剔除(见参考文献[1])。

②轮廓特征点坐标值提取(见程序 1)：用 matlab 分别提取出四组图形

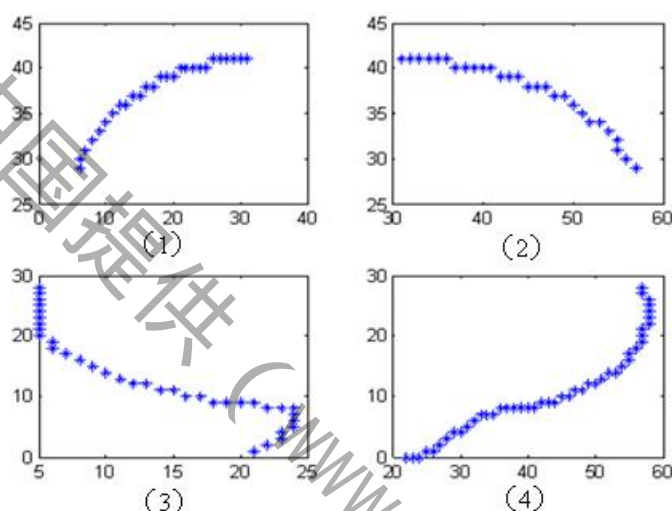
(1), (2), (3), (4)的轮廓特征点，这些点的坐标见附录表一。

③图像分割：将曲线分割成四个部分理论依据：

第一，如果采取对整个图形的轮廓特征点进行拟合的话，数据过少，误差太大，我们采取将轮廓特征点连成的图像分割成四个部分的办法（见图一）；

第二，一条曲线可在任意点切割成两条或任意多条子曲线，每一条子曲线仍是贝塞尔曲线；

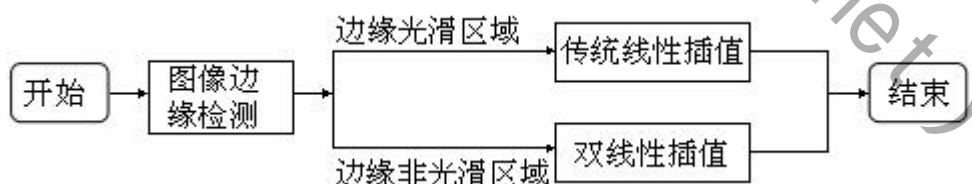
第三，一些看似简单的曲线（如圆）无法以贝塞尔曲线精确的描述，我们可以将其分成四个部分，当每个内部控制点对单位圆上的外部控制点水平或垂直的距离为 $4(\sqrt{2}-1)/3$ 时，分成四段的贝塞尔曲线，可以小于千分之一的最大半径误差近似于圆，同理我们类推到一些不规则的封闭曲线当中，我们将其切分为四个部分。



图四 图形轮廓特征点的散点图

④轮廓特征点坐标插值处理：

由于提取的轮廓特征点可能过于稀少，导致重构的轮廓失真，这里我们采取插值的办法。处理轮廓特征点的流程图：



图五 轮廓特征点插值处理流程图

第一、光滑区域一维插值处理：对于很光滑区域，即没有尖点的情况下我们采取一维插值法。即在已知的点集  $x$  上对应数值  $y$ ，构造一个解析函数，通过这些点能够求出在这些点之间的值。这里我们对切割的四块图片中的第（1）、（2）、（4）块图（见图一）都使用了一维插值法。

第二、非光滑区域双线性插值处理：双线性插值可以有效地消除锯齿现象，插值质量高。这里我们应用双线性插值的思想对图像（3）（图一）进行类双线性

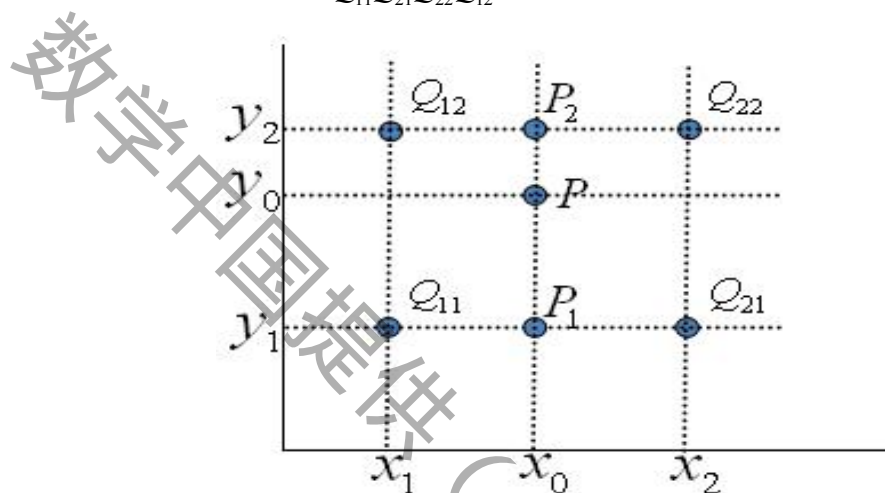


插值处理。如下：



图六 图像(3)边缘非光滑区域双线性处理流程图

具体步骤如下：假设有  $Q_{12} = (x_1, y_2), Q_{21} = (x_2, y_1)$ ，这两点由于处在非光滑区域，即认定这两点不是共线，那么我们另外补充进两点  $Q_{11} = (x_1, y_1), Q_{22} = (x_2, y_2)$ ，使得这四点成为一个矩形  $Q_{11}Q_{21}Q_{22}Q_{12}$ 。如图：



图七 边缘特征点补充图

接下来进行第一步：X方向的线性插值：

$$f(P_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}), P_1 = (x, y_1)$$

$$f(P_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}), P_2 = (x, y_2)$$

第二步：做Y方向的点  $P_1$  和  $P_2$  插值，由  $R1$  与  $R2$  计算  $P$  点：

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(P_1) + \frac{y - y_1}{y_2 - y_1} f(P_2), P = (x_0, y_0)$$

这样对每两个点都做如上处理，用 matlab 编程辅助，可以对边缘的非光滑区域做出很好的插值。

④一些看似简单的曲线（如圆）无法以贝塞尔曲线精确的描述，或分段成贝塞尔曲线（虽然当每个内部控制点对单位圆上的外部控制点水平或垂直的距离为1时，分成四段的贝塞尔曲线，可以小于千分之一的最大半径误差近似于圆）。

## (2) 模型的建立

线性贝塞尔曲线给定点  $P_0, P_1$ ， $B(t)$  代表经过  $t$  时间的坐标轴位置，线性贝塞尔曲线只是一条两点之间的直线。这条线由下式给出：

$$B(t) = P_0 + (P_1 - P_0)t = (1-t)P_0 + tP_1, t \in [0, 1]$$

二次方贝塞尔曲线二次方贝塞尔曲线的路径由给定点  $P_0, P_1, P_2$  的函数  $B(t)$  追

踪：  $B(t) = (1-t^2)P_0 + 2t(1-t)P_1 + t^2P_2, t \in [0,1]$

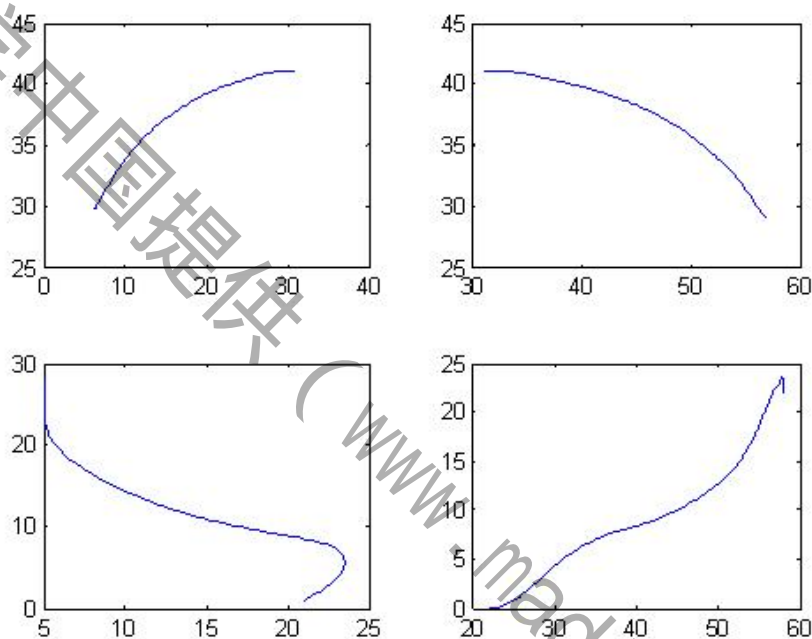
则  $n$  阶贝塞尔曲线可如下推断。给定点  $P_1, P_2, \dots, P_n$ ，其贝塞尔曲线即曲线的参数形为：

$$B(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} P_i = \binom{n}{0} P_0 (1-t)^n t^0 + \binom{n}{1} P_1 (1-t)^{n-1} t^1 + \dots + \binom{n}{n} P_n (1-t)^0 t^n, t \in [0,1],$$

我们将这个  $n$  阶的贝塞尔曲线作为我们的理论模型。

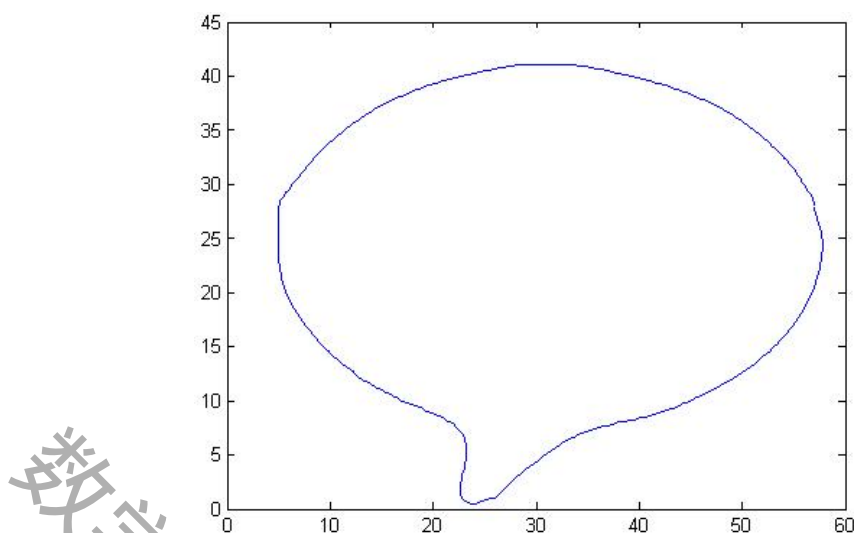
### (3) 模型的求解

利用上面求出的模型通过 matlab 软件编程求出四块图形的分别拟合情况，如图：



图八 图形轮廓特征点拟合图

将图像虽然切割成四部分，但是四部分的像素点的坐标还是不变的，所以根据坐标我们再将这四块拟合曲线组合在一起，形成如下的图像边缘线提取图：



图九 图案的边界线条图

#### (4) 模型的检验

- ①使用软件检验：Flash、Ai、Coreldraw、AutoCAD 等
- ②使用其他算法检验：Sobel 算子、Roberts 算子、Prewitt 算子等

### 二、问题二的分析与求解

#### 1. 对问题的分析

首先，因为模型 I 进行四部分切割之后发现原图与拟合后的的矢量曲线不能很好的进行重合，考虑到在取点时由于对某些不是圆滑的点直接就进行模拟，而贝塞尔曲线本身模拟出来的是一条曲线，这导致原图出现棱角的地方会由于曲线拟合而出现棱角磨平，不符合原图。

为此我们想出以下切割点的规则，即转折点切割原则：

转折点切割原则：

对于一切位图文件转化为矢量文件时，轮廓点采点过后

第一，根据已知的位图文件判断有哪些转折点，并进行标记；

第二，对已读出的轮廓点我们根据位图文件以特殊的转折点作为切割点，对由于噪点而产生的转折点丢失给予补点，并以此为端点。

使用转折点切割原则进行轮廓点切割后，应出现  $n$  组轮廓点，每组轮廓点的粗略连线应该是一条光滑的圆弧，再利用贝塞尔曲线进行拟合。

其次，模型 I 中面对  $x$  轴相等的点的数量有很多，这里我们将重点思考直线段与曲线段的矢量化问题。

#### 2. 问题的求解

##### 模型 II 图形边界线条提取修正模型

##### (1) 模型的准备

①直线区域与曲线区域的检测：当相邻 3 点的斜率相等(直线, 斜线)。其余情况用圆弧曲线区域。

②对于直线区域处理方法：由于直线区域的存在会使得提取的图形边界线条过于粗糙生硬，并且会对下一步的曲线拟合造成麻烦，也会使得图片失真，所以对于直线区域，我们要采取必要的处理。

当相邻 3 点的斜率相等(即为直线, 斜线)时候，我们采取以下两种方法：

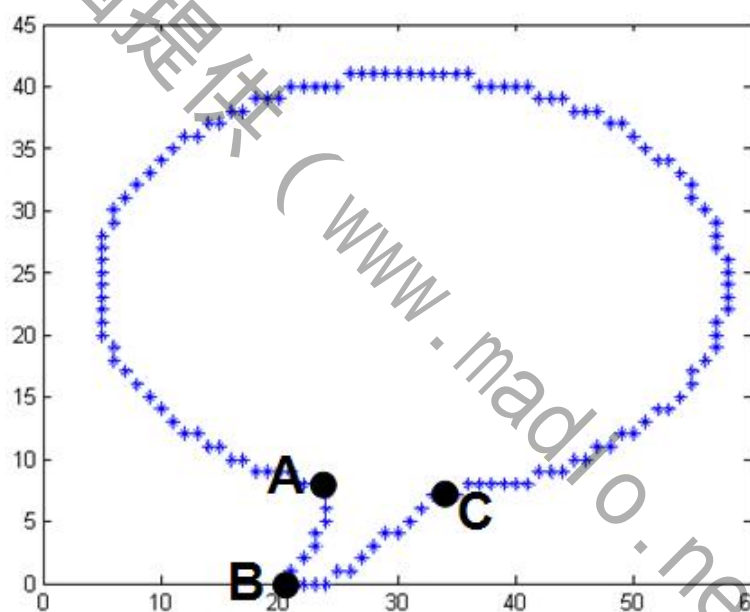
长度约束的分段方法：即为在此直线区域上每隔一定距离取得一点作为分段点，按序插入到直线区域两端点之间即可；

坐标微调法：于一个坐标可能出现多个坐标(如图中的点有多个)的情况，便需要对插值点的选取做一定微调，我们采取  $(x_i + \eta(\eta \neq 0 \text{ 且 } \eta \rightarrow 0), y_i)$  对于  $(x_i, y_i)$  做修改后带入，再进行 spline 插值拟合。

③对于圆弧曲线区域处理办法：处理方法采取模型一中一样的插值办法。

## (2) 模型的建立

首先用 matlab 提取出该图像的轮廓特征点位置，再对图像轮廓线的依照转折点重新切割成线段 AB, BC, AC 三种情况，如图：



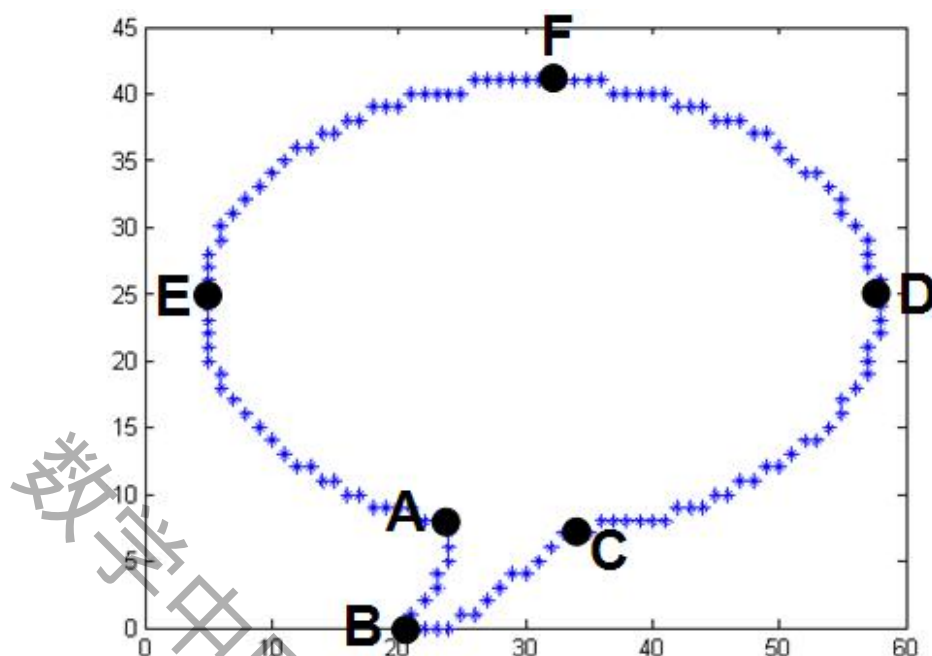
图十 图像轮廓切割图

仍然按照贝塞尔曲线拟合的办法，得出曲线的方程拟合模型：

$$B(t) = \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} P_i = \binom{n}{0} P_0 (1-t)^n t^0 + \binom{n}{1} P_1 (1-t)^{n-1} t^1 + \cdots + \binom{n}{n} P_n (1-t)^0 t^n, t \in [0,1]$$

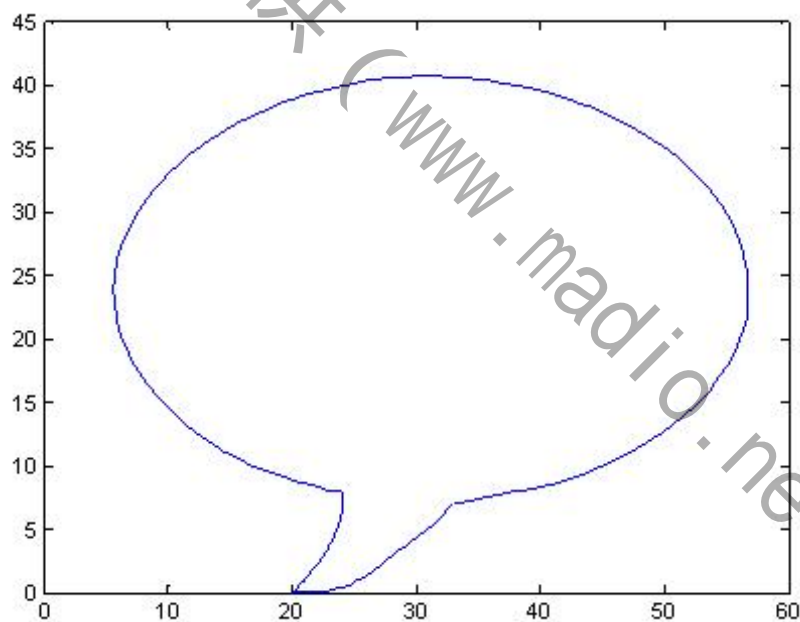
## (3) 模型的求解：

对于 AC 曲线进行拟合的结果精确却太复杂，所以我们继续对 AC 段曲线进行切割，切割成 AB, BC, CD, DE, EF, AF 六段，切割情况如下：



图十一 图像轮廓再切割图

对这五段曲线的轮廓特征点在进行贝塞尔曲线拟合的基础下，分别用 matlab 进行拟合（见程序 2）并求解方程结果如下：



图十二 图像轮廓拟合曲线

由于采取的是贝塞尔函数拟合，而贝塞尔函数特征即为很多高次项之和，故结果虽然复杂，但是能够精确拟合出曲线轮廓。

AE 段：

$$\begin{aligned}
 X(t) = & 24 - 27t + 2.22 \times 10^6 t^{19} - 1.69 \times 10^7 t^{20} + 5.65 \times 10^7 t^{21} - 1.09 \times 10^8 t^{22} \\
 & + 1.32 \times 10^8 t^{23} - 1.03 \times 10^8 t^{24} + 5.03 \times 10^7 t^{25} - 1.41 \times 10^7 t^{26} + 1.74 \times 10^6 t^{27} t^{27}
 \end{aligned}$$

$$\begin{aligned}
Y(t) = & 8 + 2.93 \times 10^3 t^3 - 5.27 \times 10^4 t^4 + 4.84 \times 10^5 t^5 - 2.96 \times 10^6 t^6 + 1.42 \times 10^7 t^7 \\
& - 6.22 \times 10^7 t^8 + 2.67 \times 10^8 t^9 - 1.09 \times 10^9 t^{10} + 3.92 \times 10^9 t^{11} - 1.20 \times 10^{10} t^{12} + \\
& 3.11 \times 10^{10} t^{13} - 6.78 \times 10^{10} t^{14} + 1.25 \times 10^{11} t^{15} - 1.94 \times 10^{11} t^{16} + 2.56 \times 10^{11} t^{17} \\
& - 2.84 \times 10^{11} t^{18} + 2.63 \times 10^{11} t^{19} - 2.02 \times 10^{11} t^{20} + 1.26 \times 10^{11} t^{21} - 6.29 \times 10^{10} t^{22} \\
& + 2.45 \times 10^{10} t^{23} - 7.14 \times 10^9 t^{24} + 1.47 \times 10^9 t^{25} - 1.90 \times 10^8 t^{26} + 1.16 \times 10^7 t^{27}
\end{aligned}$$

EF 段:

$$\begin{aligned}
X(t) = & 5 + 27t - 3.51 \times 10^2 t^2 + 5.85 \times 10^3 t^3 - 5.27 \times 10^4 t^4 + 3.23 \times 10^5 t^5 - 1.48 \times 10^6 t^6 + \\
& 5.33 \times 10^6 t^7 - 1.55 \times 10^7 t^8 + 3.75 \times 10^7 t^9 - 7.59 \times 10^7 t^{10} + 1.30 \times 10^8 t^{11} - 1.91 \times 10^8 t^{12} + \\
& 2.41 \times 10^8 t^{13} - 2.61 \times 10^8 t^{14} + 2.43 \times 10^8 t^{15} - 1.96 \times 10^8 t^{16} + 1.35 \times 10^8 t^{17} - 7.97 \times 10^7 t^{18} \\
& + 4.00 \times 10^7 t^{19} - 1.69 \times 10^7 t^{20} - 2.61 \times 10^8 t^{14} + 2.43 \times 10^8 t^{15} - 1.96 \times 10^8 t^{16} + 1.35 \times 10^8 t^{17} \\
& - 7.97 \times 10^7 t^{18} + 4.00 \times 10^7 t^{19} - 1.69 \times 10^7 t^{20} - 2.61 \times 10^8 t^{14} + 2.43 \times 10^8 t^{15} - 1.96 \times 10^8 t^{16} \\
& + 1.35 \times 10^8 t^{17} - 7.97 \times 10^7 t^{18} + 4.00 \times 10^7 t^{19} - 1.69 \times 10^7 t^{20} + 5.92 \times 10^6 t^{21} - 1.70 \times 10^6 t^{22} \\
& + 3.86 \times 10^5 t^{23} - 6.73 \times 10^4 t^{24} + 8.42 \times 10^3 t^{25} - 6.75 \times 10^2 t^{26} + 2.6 \times 10 t^{27}
\end{aligned}$$

$$\begin{aligned}
Y(t) = & 28 + 5.40 \times 10^2 t - 1.05 \times 10^3 t^2 + 1.76 \times 10^4 t^3 - 1.76 \times 10^5 t^4 + 1.21 \times 10^6 t^5 \\
& - 6.22 \times 10^6 t^6 + 2.49 \times 10^7 t^7 - 7.99 \times 10^7 t^8 + 2.06 \times 10^8 t^9 - 3.88 \times 10^8 t^{10} + \\
& 2.61 \times 10^8 t^{11} + 1.70 \times 10^9 t^{12} - 9.45 \times 10^9 t^{13} + 2.97 \times 10^{10} t^{14} - 6.91 \times 10^{10} t^{15} \\
& + 1.27 \times 10^{11} t^{16} - 1.91 \times 10^{11} t^{17} + 2.36 \times 10^{11} t^{18} - 2.40 \times 10^{11} t^{19} + 2.00 \times 10^{11} t^{20} \\
& - 1.36 \times 10^{11} t^{21} + 7.32 \times 10^{10} t^{22} - 3.06 \times 10^{10} t^{23} + 9.47 \times 10^9 t^{24} - 2.04 \times 10^9 t^{25} \\
& + 2.70 \times 10^8 t^{26} - 1.65 \times 10^7 t^{27}
\end{aligned}$$

DF 段:

$$\begin{aligned}
X(t) = & 31 + 30 \times t - 1.31 \times 10^6 t^{27} - 6.13 \times 10^5 t^{29} + 1.27 \times 10^6 t^{28} + 6.85 \times 10^5 t^{26} \\
& - 1.42 \times 10^5 t^{25} + 1.18 \times 10^5 t^{30}
\end{aligned}$$

$$\begin{aligned}
Y(t) = & 41 + 8.01 \times 10^8 t^9 - 5.93 \times 10^5 t^6 - 3.78 \times 10^9 t^{10} - 1.22 \times 10^8 t^8 + 1.22 \times 10^7 t^7 \\
& - 1.46 \times 10^{12} t^{19} - 5.29 \times 10^9 t^{27} - 4.68 \times 10^8 t^{29} + 2.06 \times 10^9 t^{28} + 1.14 \times 10^{10} t^{26} \\
& - 3.72 \times 10^{10} t^{25} + 1.41 \times 10^{11} t^{24} + 8.80 \times 10^{11} t^{22} - 1.38 \times 10^{12} t^{21} + 1.64 \times 10^{12} t^{20} \\
& + 9.42 \times 10^{11} t^{18} - 3.68 \times 10^{11} t^{17} - 1.52 \times 10^{10} t^{16} + 1.57 \times 10^{11} t^{15} - 1.45 \times 10^{11} t^{14} \\
& + 8.69 \times 10^{10} t^{13} - 3.90 \times 10^{10} t^{12} + 1.37 \times 10^{10} t^{11} - 4.14 \times 10^{11} t^{23} + 4.56 \times 10^7 t^{30}
\end{aligned}$$

CD 段:

$$\begin{aligned}
X(t) = & 33 + 32 \times t - 2.80 \times 10^7 t^{23} + 289 \times 10^9 t^{28} - 1.85 \times 10^9 t^{29} - 9.28 \times 10^8 t^{25} + 7.64 \times 10_8 \times t^{30} \\
& - 3.01 \times 10^9 t^{27} + 1.96 \times 10^7 t^{32} - 1.84 \times 10^8 t^{31} + 2.42 \times 10^8 t^{24} + 2.08 \times 10^9 t^{26}
\end{aligned}$$

$$\begin{aligned}
Y(t) = & 7 - 9.10 \times 10^6 t^6 + 1.21 \times 10^6 t^5 - 1.08 \times 10^5 t^4 + 4.96 \times 10^3 t^3 - 1.75 \times 10^{13} t^{21} + 5.05 \times 10^7 t^7 \\
& - 4.94 \times 10^{10} t^{12} + 6.99 \times 10^{12} t^{20} + 4.55 \times 10^{12} t^{17} - 2.21 \times 10^8 t^8 - 4.94 \times 10^{10} t^{12} + 6.99 \times 10^{12} t^{20} \\
& + 4.55 \times 10^{12} t^{17} - 2.21 \times 10^8 t^8 - 6.06 \times 10^{11} t^{14} + 8.13 \times 10^8 t^9 - 3.04 \times 10^{12} t^{16} - 4.50 \times 10^{12} t^{18} \\
& + 1.53 \times 10^{12} t^{15} - 3.01 \times 10^{13} t^{23} + 1.86 \times 10^{12} t^{28} - 2.90 \times 10^9 t^{10} + 1.16 \times 10^{10} t^{11} - 5.03 \times 10^{11} t^{29} \\
& + 2.65 \times 10^{13} t^{22} - 1.94 \times 10^{13} t^{25} + 9.62 \times 10^{10} t^{30} - 5.19 \times 10^{12} t^{27} + 1.91 \times 10^{11} t^{13} + 6.80 \times 10^8 t^{32} \\
& - 1.16 \times 10^{10} t^{31} + 2.69 \times 10^{13} t^{24} + 9.33 \times 10^{11} t^{19} + 1.12 \times 10^{13} t^{26}
\end{aligned}$$

BC 段:

$$X(t) = 24 + 200 \times t^7 - 50 \times t^8 - 308 \times t^6 + 224 \times t^5 - 70 \times t^4$$

$$Y(t) = 8 - 8 \times t$$

AB 段:

$$\begin{aligned}
X(t) = & 20 - 3.63 \times 10^4 t^{10} + 1.20 \times 10^4 t^7 + 26t + 4.07 \times 10^4 t^9 - 2.83 \times 10^4 t^8 - 3.43 \times 10^3 t^6 + \\
& 1.28 \times 10^3 t^5 - 7.15 \times 10^2 t^4 + 2.86 \times 10^2 t^3 - 78t^2 + 7.93 \times 10^2 t^{13} - 6.019 \times 10^3 t^{12} + 1.9734 \times 10^4 t^{11}
\end{aligned}$$

$$\begin{aligned}
Y(t) = & 7.72 \times 10^3 t^{10} - 3.26 \times 10^4 t^7 - 2.50 \times 10^4 t^9 + 3.73 \times 10^4 t^8 + 1.71 \times 10^4 t^6 - 5.14 \times 10^3 t^5 + \\
& 7.15 \times 10^2 t^4 + 4.62 \times 10^2 t^{13} - 1.84 \times 10^3 t^{12} + 1.24 \times 10^3 t^{11}
\end{aligned}$$

#### (4) 模型的检验:

##### ①模型检验思想:

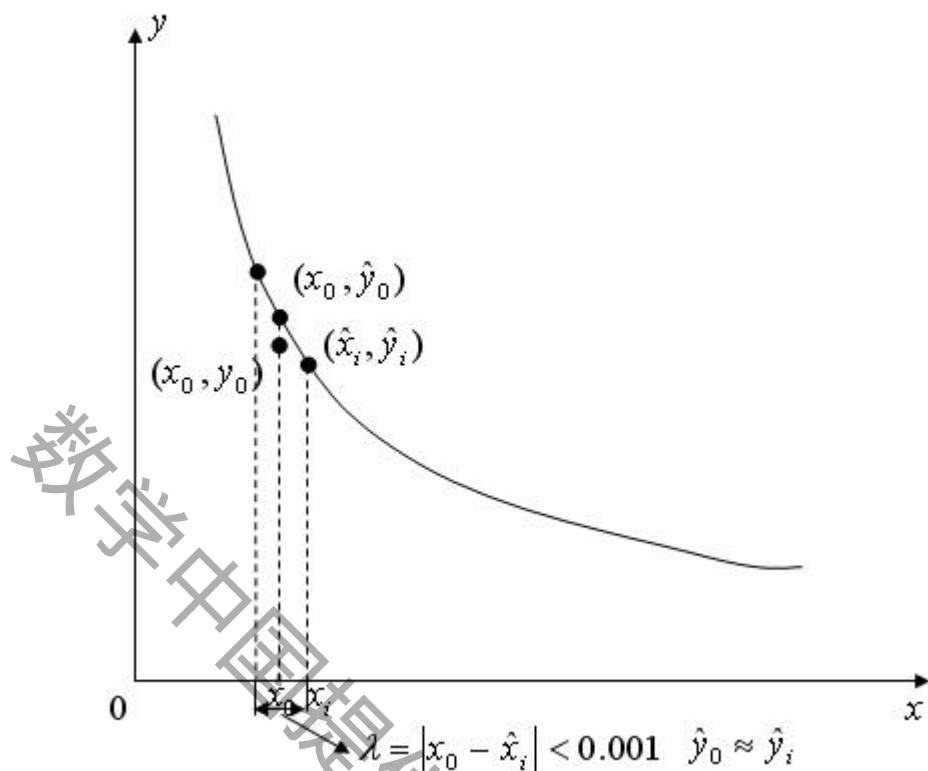
根据贝塞尔曲线的推理过程,其本身无法用一般方程进行表示,则用参数方程:  $X(t), Y(t), (0 \leq t \leq 1)$  表示贝塞尔曲线。但参数方程本身无法利用公式

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

进行方程拟合优度的检验,为此我们提出极限替代法思想;

极限替代法:

如下图十, 设  $t$  的取值从 0 开始到 1 结束, 中间步长为 0.00001, 相应的  $X(t), Y(t)$  有相应的值  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$ ,  $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$ , 原始数据点  $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ , 任取原始数据一点  $x_0$ , 则  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$  中必存在一点  $\hat{x}_i$ , 使得  $\lambda = |\hat{x}_i - x_0| < 0.001$ , 此时因为  $\hat{x}_i$  与  $x_0$  之间的距离非常小 ( $\lambda < 0.001$ ), 我们可以认为  $\hat{x}_i = x_0$ , 此时由于参数  $t$  的问题,  $\hat{y}_i$  的值可以认为是当  $x = x_0$  时  $\hat{y}_0$  的值, 以此类推, 分别可以求出  $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$ , 再代入计算公式  $R^2$

图十三  $\hat{y}_0$  值的模糊取值示意图

## ②模型检验结果：

AE 段拟合曲线  $R^2=0.9431$ , 拟合效果很好；EF 段拟合曲线  $R^2=0.9946$  拟合效果非常好；DF 段拟合曲线  $R^2=0.9921$ , 拟合效果非常好；CD 段拟合曲线  $R^2=0.9796$ , 拟合效果非常好；BC 段拟合曲线  $R^2=0.7572$  拟合效果较好；AB 段拟合曲线  $R^2=0.9898$ , 拟合效果非常好。

综上所述所有的曲线区域都通过了拟合优度的检验。这一点从图和图对比也可以看出。

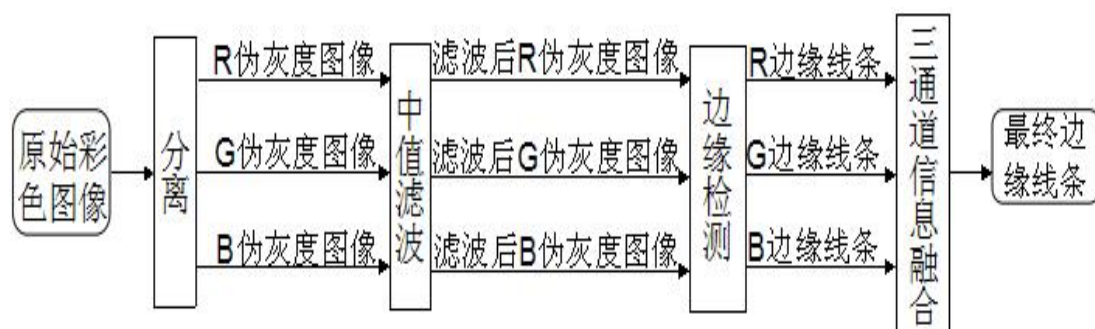
## 三、问题三的分析与求解

## 1. 对问题的分析

题目中所给的图形只是简单的黑白图形，但是现实生活中很多都为彩色图片。这里我们思考对简单的彩色图形边界线条提取问题。

## ①彩色图片边缘线条提取流程：





图十四 彩色图像边缘线条提取流程图

彩色图像的转换为 R, G, B 三个灰度图像使用 matlab 编程实现，由于彩色图像中的噪点对拟合结果产生影响较大，所以仅仅是采用 ps 软件进行噪点剔除是远远不够的，这里采取中值滤波剔除图像中的噪点，排除干扰，从而对于三个灰度图像进行边缘检测处理。

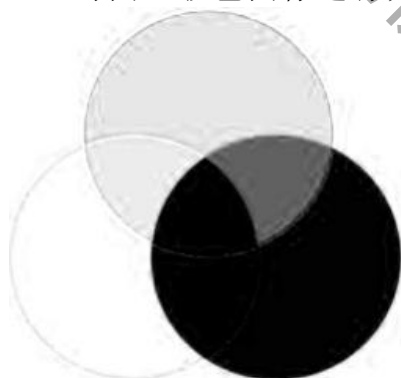
## 2.问题的求解

### 模型III 简单彩色图形边界线条提取模型

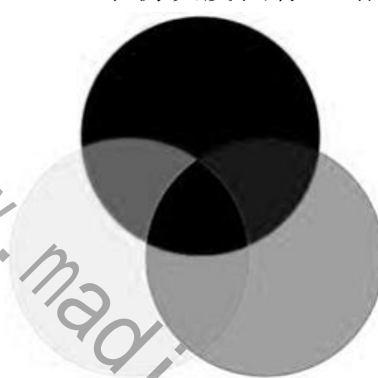
#### (1) 模型的准备

##### ①原始彩色图像分离：

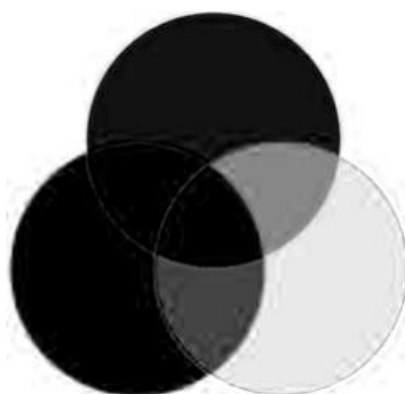
对图一彩色图像进行分离，得到 R, G, B 三个伪灰度图像，结果如下



图十五-1 R 伪灰度图像

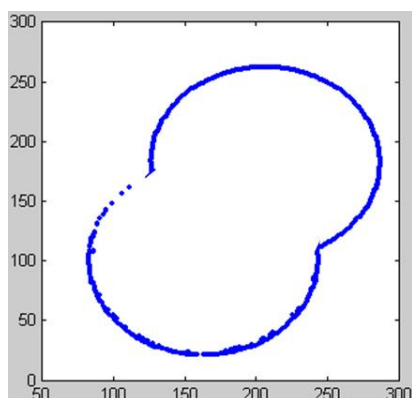


图十五-2 G 伪灰度图像

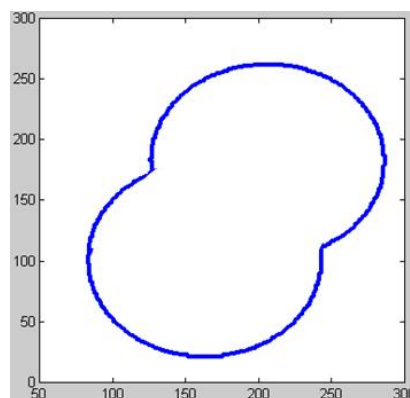


图十五-3 B 伪灰度图像

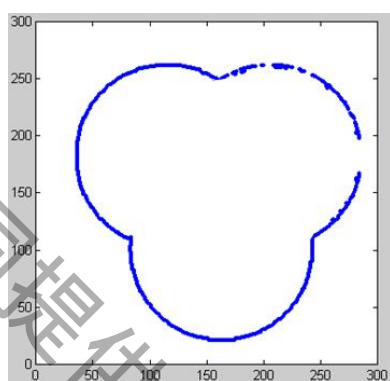
②对分离的图像分别进行中值滤波处理之后，得到的 R, G, B 三个图像的边缘线条，结果如下



图十六-1 R 边缘线条图



图十六-2 G 边缘线条图



图十六-3 B 边缘线条图

## (2) 模型的建立和求解

假设 R 边缘线条图上点坐标为  $(X_R, Y_R)$ , G 边缘线条图上点坐标为  $(X_G, Y_G)$ , B 边缘线条图上点坐标为  $(X_B, Y_B)$ , 三幅图融合成的曲线的坐标为  $(X_0, Y_0)$ 。考虑到三条曲线轮廓特征点的分布情况, 采取如下方法进行坐标点的剔除和融合:

$$\begin{cases} (X_R - X_G)^2 + (Y_R - Y_G)^2 = 0 & \dots\dots\dots \text{I} \\ (X_R - X_G)^2 + (Y_R - Y_G)^2 \neq 0 & \dots\dots\dots \text{II} \end{cases}$$

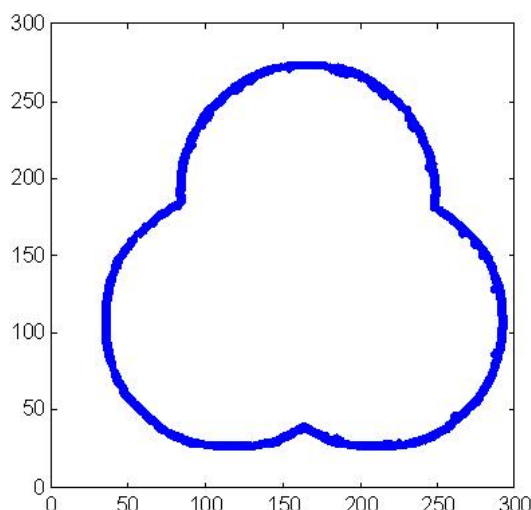
①边缘轮廓特征点并点法: 如果上面 I 式成立, 则说明该两点在 R 图和 G 图上为坐标位置相同的点, 保留其中一点剔除另外一点即可; 如果 II 式成立, 则说明该两点在 R 图和 G 图上分布位置不同, 都要保留。

②非边缘点剔除法: 对于不是轮廓的特征点我们对其提出, 以免干扰曲线拟合结果。

同理, 对 R 图和 B 图, G 图和 B 图做同样的处理, 最后提取的点进行贝塞尔方程的拟合处理, 即为

$$\begin{aligned} B(t) &= \sum_{i=0}^n \binom{n}{i} (1-t)^{n-i} P_i \\ &= \binom{n}{0} P_0 (1-t)^n t^0 + \binom{n}{1} P_1 (1-t)^{n-1} t^1 + \dots + \binom{n}{n} P_n (1-t)^0 t^n, t \in [0,1] \end{aligned} \quad (1)$$

对处理后的坐标值用 matlab 画出来, 结果如下:



图十七 彩色图像边界线条图

## § 6 模型的误差分析

**1、考虑噪点的影响：**噪点主要是指图像中不该出现的外来像素，通常由电子干扰产生。由于噪点在图像之中的广泛存在性，再经栅格化之后原先的这些噪点会形成一些围绕在图像周围的无序点，这样在经过逆栅格化过程中找到的图形轮廓便会产生一定的散乱点，虽然我们已经使用 ps 软件处理噪点，但是还是会给拟合曲线产生一定误差。

**2、考虑图像切割和拼接的影响：**为了精确拟合曲线方程，我们采取对图像的切割，然后再将切割的曲线拼接，但是在切割过程中，任意两段原先可连接曲线之间处的端点并不可能都一致。且按照贝塞尔曲线拟合过程中，每段拟合曲线较之前散点图走向会出现一定程度内扭曲或偏离，导致分段拟合之后的曲线之间可能缺乏一定的连贯性，则拟合后的曲线会与原图像产生一定误差。

**3、考虑数据插值的影响：**栅格化图像轮廓特征点提取之后，存在对于一个  $x_i$  坐标可能同时对应多个  $y_i$  坐标(如图中的  $x=5$  的点有多个)的情况，同时也会出现一个  $y_i$  对应过个  $x_i$  的情况，这时便需要对插值点的选取做一定微调，采取

$(x_i + \eta(\eta \neq 0, \eta \rightarrow 0), y_i + \eta(\eta \neq 0, \eta \rightarrow 0))$  对于  $(x_i, y_i)$  做微调后带入，再进行 spline 插值拟合。虽然  $\eta$  取值很小，但是也会对图形的拟合造成一定的误差影响。

## § 7 模型的评价

### 一、模型的优点：

1. 本文巧妙运用流程图，将建模思路完整清晰的展现出来；
2. 利用 Malab, Excelde 等软件对数据进行处理并作出各种图表，简便，直观，快捷，准确；
3. 本文建立的模型与实际联系紧密，充分考虑简单图形的不同曲线轮

廓图的不同情况，从而使模型更贴近实际。

4. 插值采取多种方法更加准确拟合曲线，并且对于直线和曲线轮廓线采取不同的处理办法。

5. 创新性地引入贝塞尔曲线进行曲线拟合，使得曲线边缘线的矢量化处理更加精确圆滑。

6. 不仅到黑白图片的边缘轮廓线的处理，还考虑到彩色图片的情况，思维周到且全面。

## 二、模型的缺点：

1. 对于一些数据，我们对其进行了一些必要的处理，例如对于曲线的插值的处理，异常值的处理，这些都会带来一些误差；

2. 模型中为使计算简便，使所得结果更理想化，忽略了一些次要影响因素。例如处理图片本身质量问题对于图片边缘轮廓线提取的影响。

## § 8 模型的改进与推广

### 一、改进

对于任意一种简单平面图形，如果经 matlab 初步计算出散点数量较大，可以人为观察出来曲线占据了轮廓图形的主要部分且曲线扭曲程度较大，采用模型二的贝茨拟合方法能够较为准确地模拟出原曲线，但对于一些明显的简单曲线，

可以直接选取

$$y = ax^n + bx^{n-1} + cx^{n-2} + \cdots M(n \leq 10), y = ae^x (a \neq 0), y = ax + b \text{ 等基本函数}$$

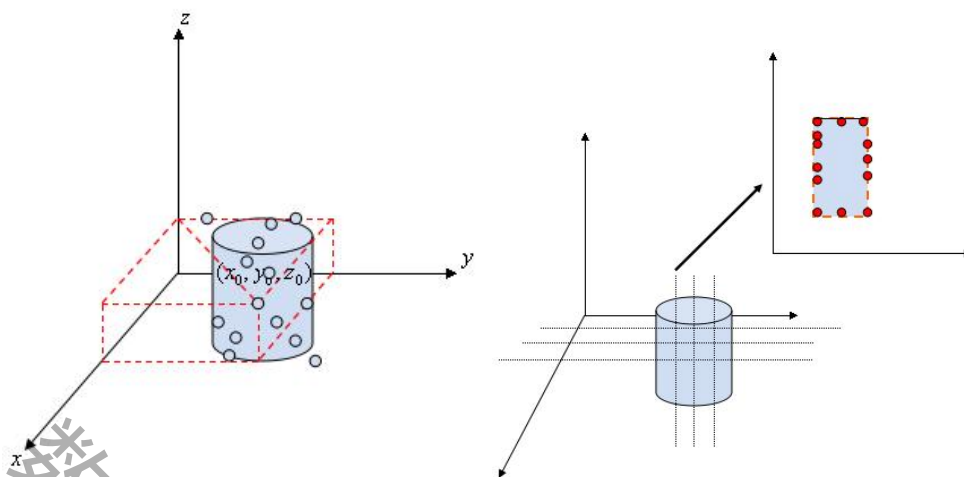
来模拟，这样较贝茨函数模拟出的原函数差距并不会太大的情况下，并且可以减少运算程序，提高效率。

### 二、模型的推广

已建的模型是在二维坐标系上的曲线拟合，适用于拟合简单平面图形。同样在三维立体空间中，如果已知某立体图形在坐标系中的一些列三维坐标

$(x_i, y_i, z_i), i \text{ 取 } 0, 1, 2, 3 \cdots$ ，分别对  $x$ 、 $y$ 、 $z$  轴作  $M'$  次垂直平面切割。以  $x$  轴为

例，用与  $x$  轴垂直的  $M'$  个平面  $\alpha_i$  切割  $x$  轴，可以获得  $M'$  个  $yz$  平面上的二维散点图，这时再使用模型二对每一个曲线进行拟合，将拟合过后的所有曲线连接起来便可以得到环绕  $x$  轴的一个连续曲面，同理可以求出围绕  $y$ 、 $z$  轴的两个环绕曲线。再对三个曲面进行取交线，做一些简单杂线去除便可以求出空间坐标中物体的大致轮廓图。



图十八 三维图形推广原理图

### 参考文献

- [1] 用 ps 技术处理图片噪点：  
<http://jingyan.baidu.com/article/b907e627f170fb46e6891c5c.html>
- [2] 严兰兰, 宋来忠, 李军成. 有理 Bezier 曲线的拼接[J]. 三峡大学学报(自然科学版). 2005. 27(5): 469-471.
- [3] 姜启源等. 数学模型[M]. 北京: 高等教育出版社, 2003. 8 第三版.
- [4] 杨桂元, 黄己立. 数学建模[M]. 合肥: 中国科技大学出版社, 2008
- [5] 吴礼斌. 经济数学实验与建模[M]. 天津大学出版社. 2009. 8.

## 附录

表[1] 图像轮廓特征点坐标

分 组 编 号	(1)		(2)		(3)		(4)	
	X	Y	X	Y	X	Y	X	Y
1	5	28	31	41	5	28	26	1
2	6	30	32	41	5	27	26	2
3	6	29	33	41	5	26	27	3
4	7	31	34	41	5	25	28	4
5	8	32	35	41	5	24	29	4
6	9	33	36	41	5	23	30	5
7	10	34	37	40	5	22	31	6
8	11	35	38	40	5	21	32	7
9	12	36	39	40	5	20	33	7
10	13	36	40	40	6	19	34	7
11	14	37	41	40	6	18	35	8
12	15	37	42	39	7	17	36	8
13	16	38	43	39	8	16	37	8
14	17	38	44	39	9	15	38	8
15	18	38	45	38	10	14	39	8
16	19	39	46	38	11	13	40	8
17	20	39	47	38	12	12	41	9
18	21	40	48	37	13	12	42	9
19	22	40	49	37	14	11	43	9
20	23	40	50	36	15	11	44	10
21	24	40	51	35	16	10	45	10
22	25	40	52	34	17	10	46	1
23	26	41	53	34	18	9	47	1
24	27	41	54	33	19	9	48	11
25	28	41	55	32	20	9	49	12
26	29	41	55	31	21	9	50	12
27	30	41	56	30	22	8	51	13
28	31	41	57	29	23	8	52	14
29			57	28	24	8	53	14
30					24	7	54	15
31					24	6	55	16
32					24	5	55	17
33					23	4	56	18
34					23	3	57	19
35					22	2	57	20

36					21	1	57	21
37					22	0	58	22
38					23	0	58	23
39					24	0	58	24
40					25	1	58	25
41					26	1	58	26
42							57	27
43							58	28

程序[1]:

```
>> a=imread('1.bmp');
```

```
>> imshow(a)
```

```
>> b=rgb2gray(a);
```

```
>> c=edge(b);
```

m 文件:

```
i=1;
```

```
x=[];y=[];
```

```
for x1=1:124
```

```
    for x2=1:83
```

```
        if c(x2,x1)==1
```

```
            x(i)=x1;
```

```
            y(i)=x2;
```

```
            i=i+1;
```

```
        end
```

```
    end
```

```
end
```

```
y1=[];
```

```
for i=1:136
```

```
    y1(i)=45-y(i);
```

```
end
```

```
xx2=[];yy2=[];j=1;k=1;
```

```
for i=1:136
```

```
    if y1(i)>28
```

```
        xx2(j)=x(i);
```

```
        yy2(j)=y1(i);
```

```
        j=j+1;
```

```
    else
```

```
        xx1(k)=x(i);
```

```
        yy1(k)=y1(i);
```

```
        k=k+1;
```

```

end
end
程序[2]:
贝塞尔曲线编程:
function [X,Y]=bezier(x,y)
%用法:
%bezier(x,y)
%      生成 n-1 次贝塞尔曲线, 其中 x 和 y 是 n 个点的坐标
%h=bezier(x,y)
%      生成 n-1 次贝塞尔曲线并返回曲线句柄
%[X,Y]=bezier(x,y)
%      返回 n-1 次贝塞尔曲线的坐标
%例子:
%bezier([5,6,10,12],[0 5 -5 -2])
n=length(x);
t=linspace(0,1);
xx=0;yy=0;
for k=0:n-1
    tmp=nchoosek(n-1,k)*t.^k.*(1-t).^(n-1-k);
    xx=xx+tmp*x(k+1);
    yy=yy+tmp*y(k+1);
end
if nargin==2
    X=xx;Y=yy;
end
h=plot(xx,yy);
if nargin==1
    X=h;
end

xxx1=[9,xx2(:,1:40)];yyy1=[15,yy2(:,1:40)];
xxx2=[xx2(:,40:80),58];yyy2=[yy2(:,40:80),22];
xxx3=[xx1(:,1:41)];yyy3=[yy1(:,1:41)];
xxx4=[xx1(:,41:82)];yyy4=[yy1(:,41:82)];

```

贝塞尔曲线参数方程求解:

```

syms t
n=length(X3);
xx=0;yy=0;
for k=0:n-1
    tmp=nchoosek(n-1,k)*t.^k.*(1-t).^(n-1-k);
    xx=xx+tmp*X3(k+1);
    yy=yy+tmp*Y3(k+1);
end

```



```
simplify(xx)
simplify(yy)
```

曲线拟合度检验：

```
i=1;d=[];e=[];
for t=0:0.00001:1
z=31+30*t-1319500*t.^27-613440*t.^29+1271940*t.^28+685125*t.^26-142506*t.^25+118378*t.^30;
z1=41+801200400*t.^9-593775*t.^6-3785671890*t.^10-122911425*t.^8+12214800*t.^7-1462700584800*t.^19-5296046700*t.^27-468742500*t.^29+2068733850*t.^28+11429501895*t.^26-37201191300*t.^25+141374858625*t.^24+880086773475*t.^22-1384188148200*t.^21+1640608044075*t.^20+942516672825*t.^18-368980097850*t.^17-15269380875*t.^16+157444282800*t.^15-145713520350*t.^14+86945651100*t.^13-39008444475*t.^12+13711452300*t.^11-414309729600*t.^23+45677580*t.^30;
    if (abs(X3(i)-z)<0.001)&(i<=length(X3))
        d(i)=z;
        e(i)=z1;
        i=i+1;
    end
End
```

程序[3]：

```
R2=1-sum((Y3-e).^2)/sum((Y3-mean(Y3)).^2)
```

彩色图像三通道分离：

```
>> I=imread('2.bmp');
>> R=I(:, :, 1);
>> G=I(:, :, 2);
>> B=I(:, :, 3);
>> C1=edge(R);
>> C2=edge(G);
>> C3=edge(B);
```

```
c=[];
for i=1:289
    for j=1:302
        C(i,j)=C1(i,j)+C2(i,j)+C3(i,j);
    end
end
```

```
>> imshow(C);
>> J=imread('4.bmp');
>> D=rgb2gray(J);
>> c=edge(D);
```

```
i=1;
```

```
x=[];y=[];  
for x1=1:306  
    for x2=1:291  
        if c(x2,x1)==1  
            x(i)=x1;  
            y(i)=x2;  
            i=i+1;  
        end  
    end  
end  
plot(x,y,'.')  
axis square
```