实验报告

——基于 AlphaShape 的轮廓识别

1. 软件概述

1.1 项目背景

随着计算机技术的发展，越来越多的领域有着大量待处理的图像数据。而在进行图像处理前，必须经过一些预处理把原始图像数据中所蕴含的有效信息进行提炼。而在各种预处理技术中，轮廓识别是一种最基本的方法，有着广阔的应用背景。传统的边界识别的方法基本上是利用边界点像素颜色特征进行运算的，并没有利用到位置信息。这种方法不仅计算量特别大，而且还会存在误差。所以我们希望能够绕开颜色，从边界点的位置特征入手，利用 AlphaShape 算法对图像中物体轮廓进行识别。

1.2 AlphaShape 算法介绍

AlphaShape 算法是一种利用某些特征点来刻画点集直观轮廓的一种算法。一个 AlphaShape 是指由一些特定点所决定的具体的几何体。而 AlphaShape 算法就是来求解这一几何体的。

AlphaShape 是凸包的一般化。给定一个点集 S，以及一个参数 alpha，那么 S 的 AlphaShape 是一个多边形（高维中就是多面体），它不一定是凸多边形（或多面体），甚至不一定是连通的。通过设置参数 alpha 我们可以得到同一点集的一系列 AlphaShape，而这些 AlphaShape 都是在设定了 alpha 值条件下的这些点集的轮廓。

具体说来，AlphaShape 是点集的曲边凸包。随着 alpha 取值得不同，边的弯曲程度不同（这里的曲边是指在生成凸包时假想的曲边，利用这些曲边我们能够得到构成这个凸包的边界点）。当 alpha 足够大时，边的曲率为 0，这时 AlphaShape 就是凸包，随着 alpha 值慢慢减小，各边慢慢向里弯曲，从而得到一个更加形象的点集的边界。

实际操作中的具体做法是，对于给定点集中的任意两个点，如果有他们和 alpha 值所确定的左右两个圆中，任意一个圆内不包含任意其他点，那么这两点之间便有边相连，否则则不相连。

1.3 算法实现构思

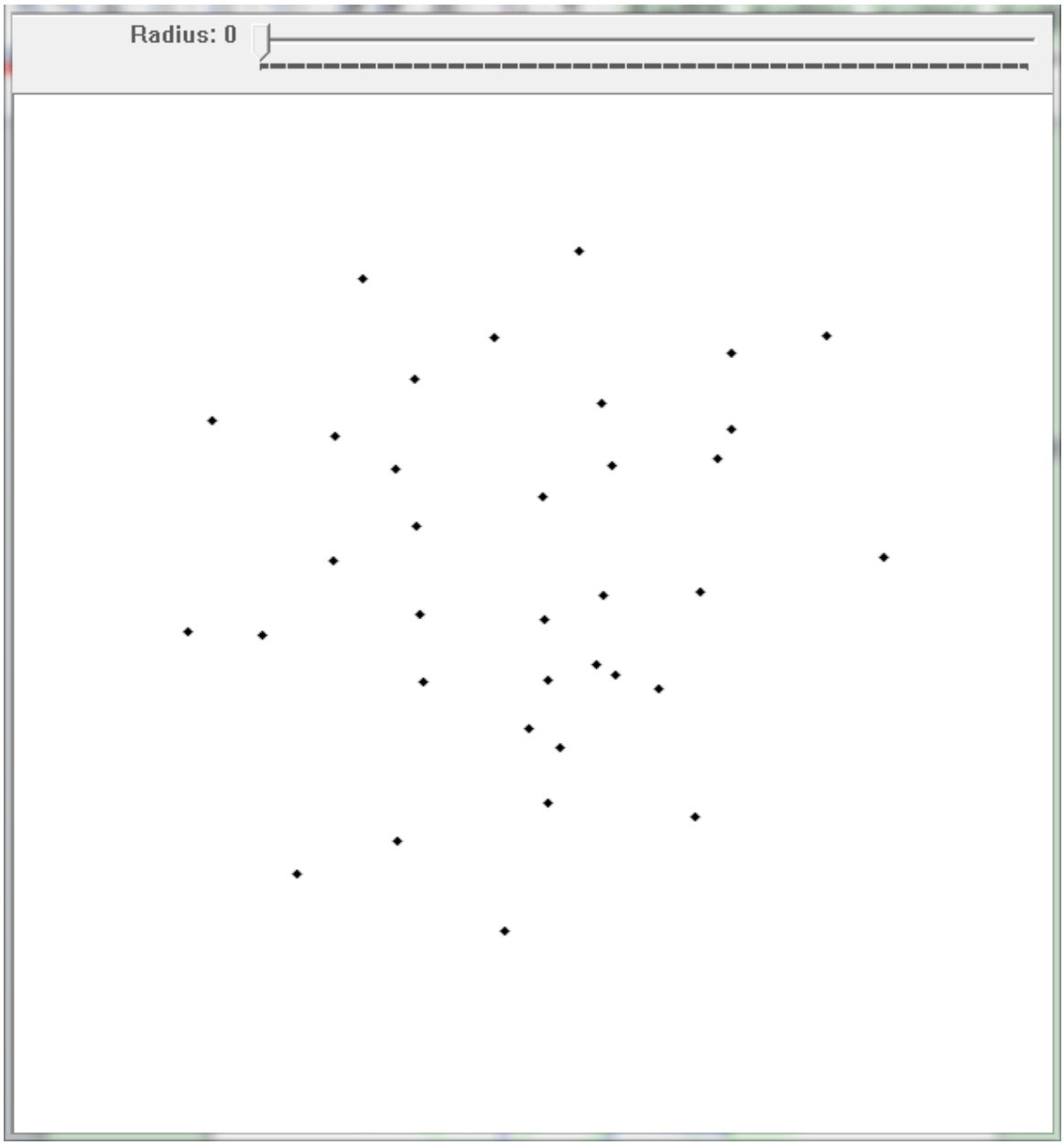
虽然 AlphaShape 是由更一般意义上的凸包引出的，但我们并不能沿用凸包算法对 AlphaShape 进行构造。这是因为在凸包算法中，我们只需要考虑边界上的点，对于与边界点不相连的点可以忽略，但是在 AlphaShape 构造中，并不能简单的把凸包的内点省略，因为内部点的也可能满足连边的条件。所以，为了求出 AlphaShape，我们几乎需要考虑每一对点。那时间复杂度不就是 O(n^3)了吗？

后来通过查找文献，我们发现 AlphaShape 还有一个很好的性质——它是 DT 剖分的子图。这样我们只需要用 O(nlogn)时间找出 DT 剖分，然后对 DT 剖分的每一条边的两个顶点进行判断，看看与它们相连的顶点是不是都在某一个 AlphaShape 圆外，而这一过程的时间复杂度是 O(n)的。这样，原本 O(n^3)的问题就可以在 O(nlogn)时间复杂度内做出来了。

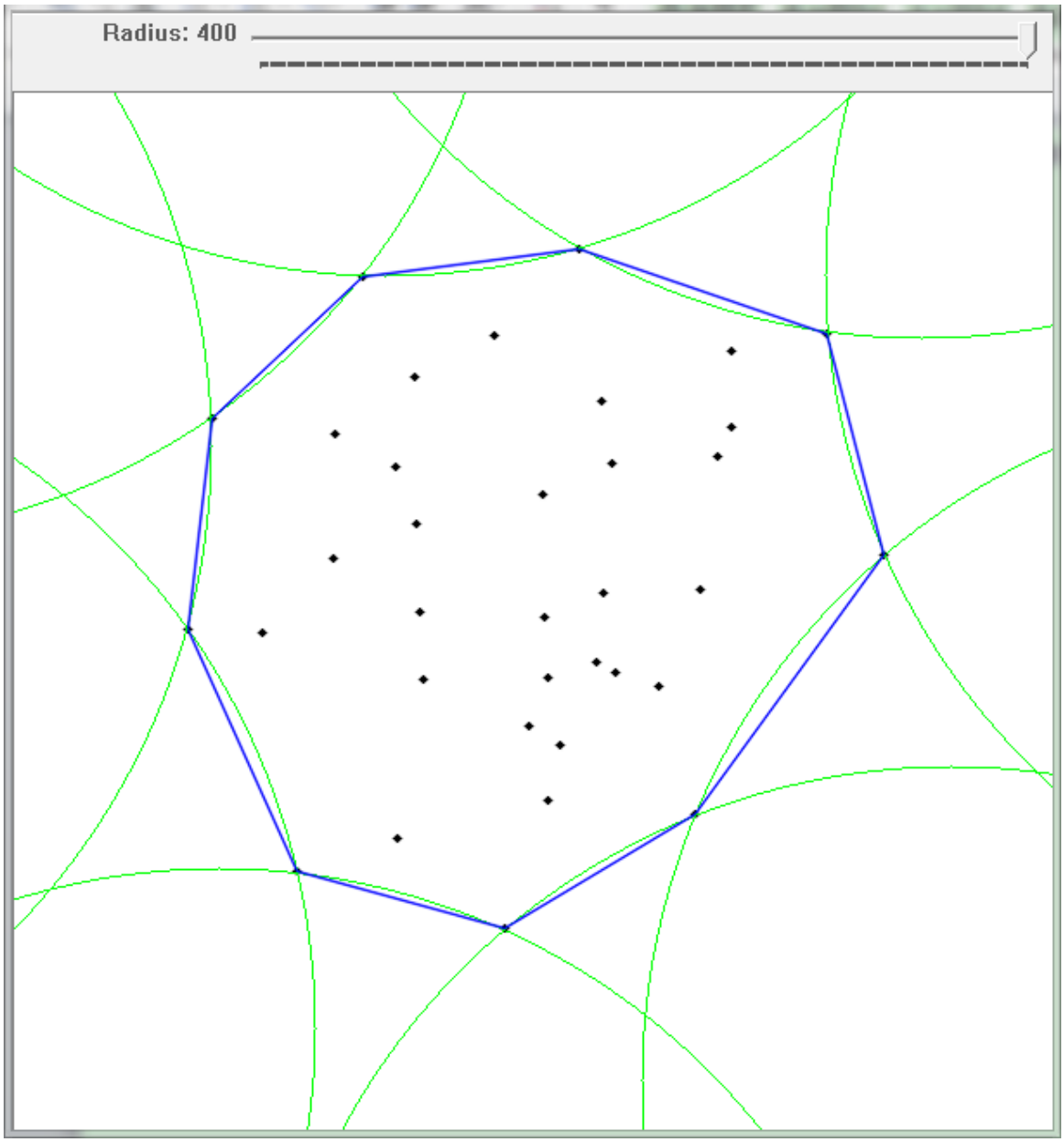
我们的程序正是利用这一方法进行设计和实现的。

为了便于大家理解 AlphaShape 原理，我们的项目中还包含了一个 AlphaShape 原理演示程序

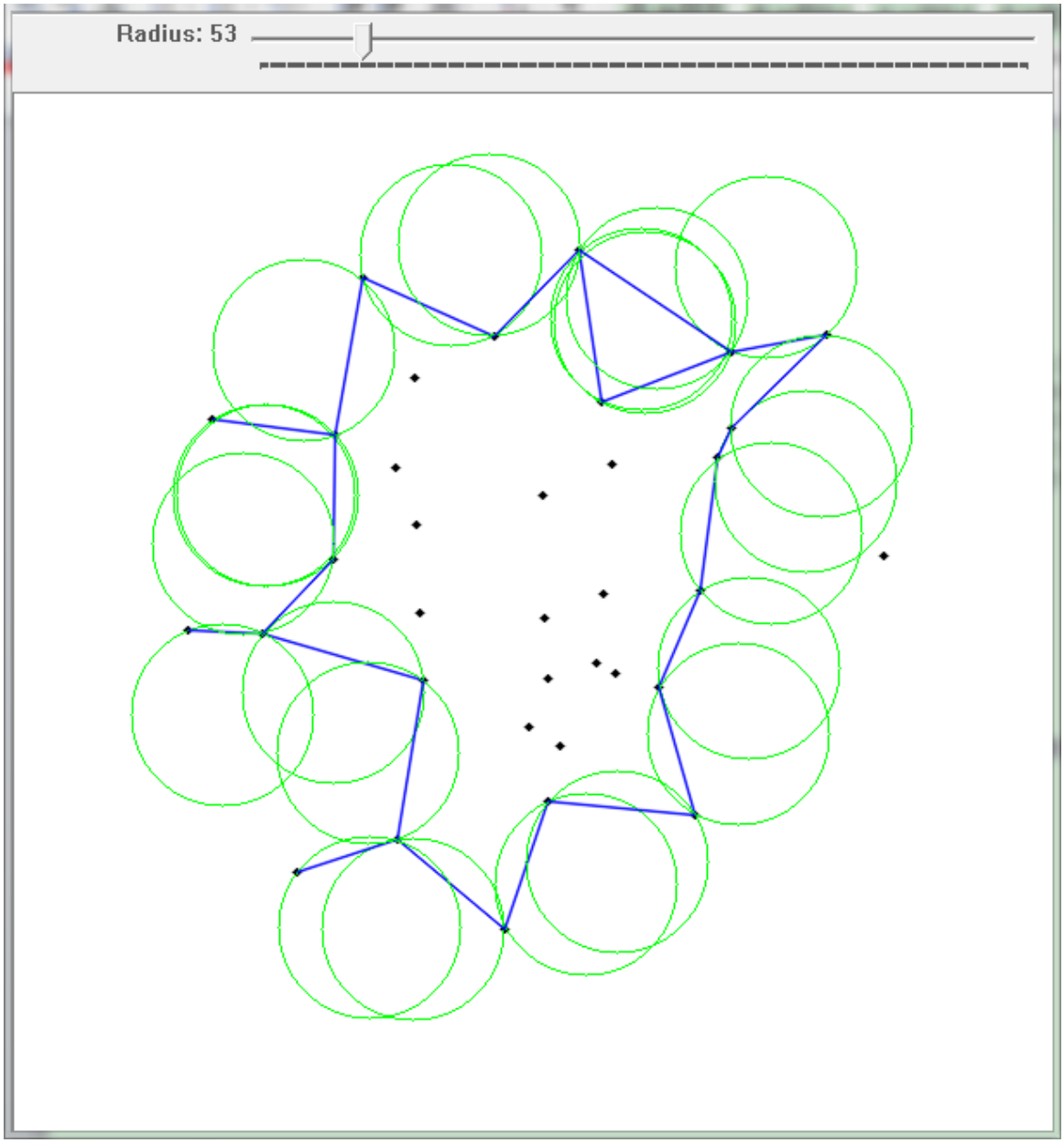
（Demo 文件夹下的 AlphaShape 原理.exe），下面是该程序运行过程中的一些关键画面截图。



当参数 Radius 取 0 时为单个的点



当参数 Radius 取 400（相当于足够大）时为凸包



当参数 Radius 取中间某一合适值时提取轮廓

1.4 算法应用

由于 AlphaShape 能够通过改变 alpha 参数值来改变最终 AlphaShape 的形状，而 AlphaShape 是点集的一种非常直观的轮廓的刻画，因此我们发现可以利用 AlphaShape 的这个性质来刻画图像中物体的轮廓。不同的 alpha 参数值所产生 AlphaShape 能够在不同细致程度上体现出点集的直观轮廓，所以当我们减小 alpha 值时，便可以得到点集轮廓的更细致的描绘，这使得我们可以通过优化 alpha 参数值在有干扰的情况下仍能准确得出图像中物体的轮廓，这是因为少量的干扰像素不足以改变图像中物体的基本轮廓。

2. 使用说明

2.1 运行环境

普通计算机上 Windows 系统环境下均可运行。

2.2 测试数据

Data 文件夹中是我们组开发时的测试数据集，请大家优先选用。如果大家需要用自己的数据集进行测试，建议使用轮廓比较明显的后缀名为 jpg 或 bmp 的图片（推荐使用卡通图片）。

2.3 操作说明

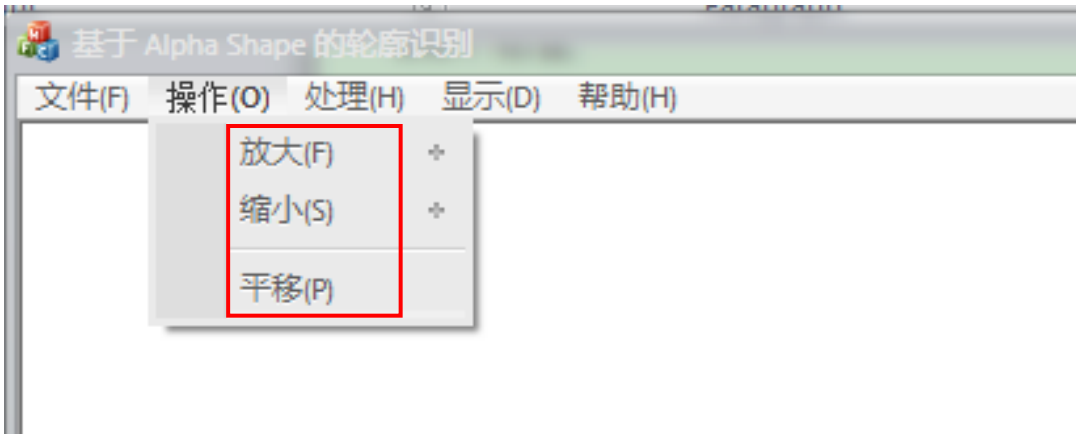
简单起见，请大家参照演示录像进行学习。

2.3.1 功能

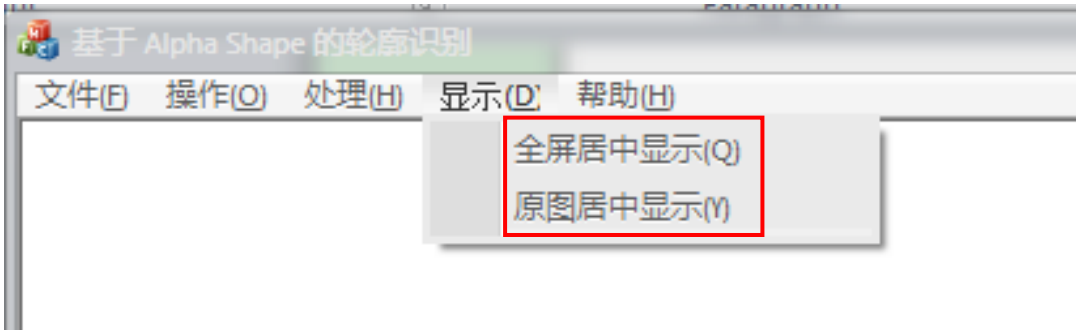
1、图片或数据文件的导入和导出（主意文件名称以及文件存放路径不含英文）

此处需要特殊说明的是，导出功能只导出特征点，后缀是.dat。之所以这样设计的好处是，对任何复杂的图形轮廓，我们只需要将少量的特征点存在硬盘上，在需要获得轮廓时用该软件把这些特征点复原成轮廓即可。

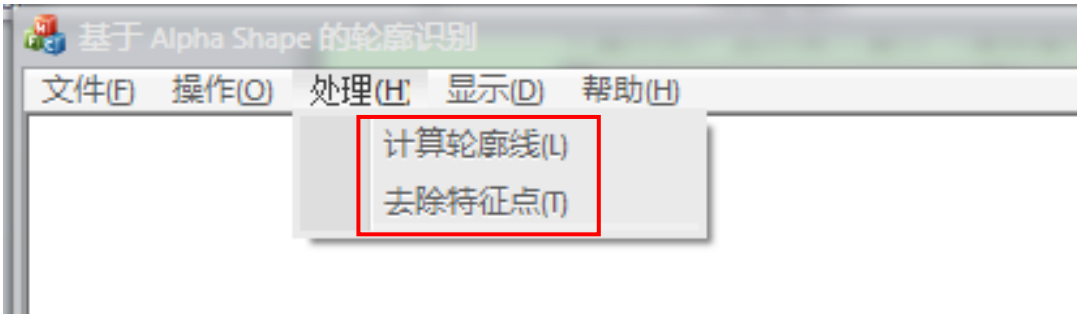
2、操作类：放缩（同比例放缩或 X 轴 Y 轴放缩）、平移



3、显示类：全屏居中显示（适合窗口大小）和原图居中显示（保持原图大小）

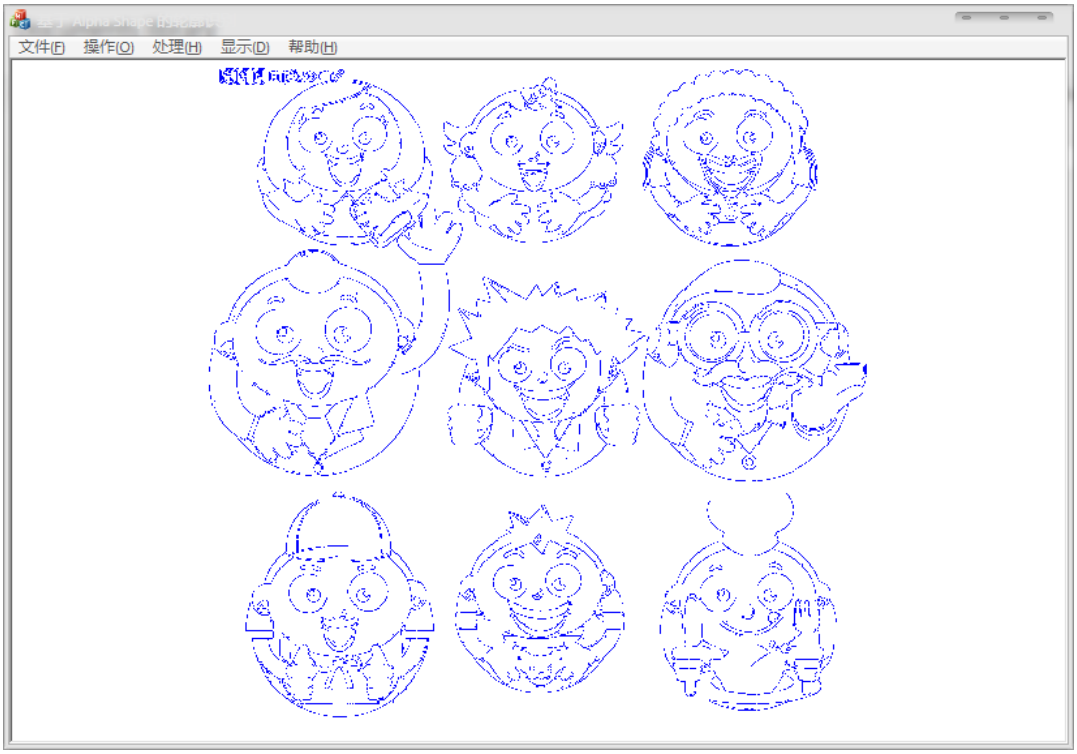


4、处理类：计算轮廓线和去除特征点



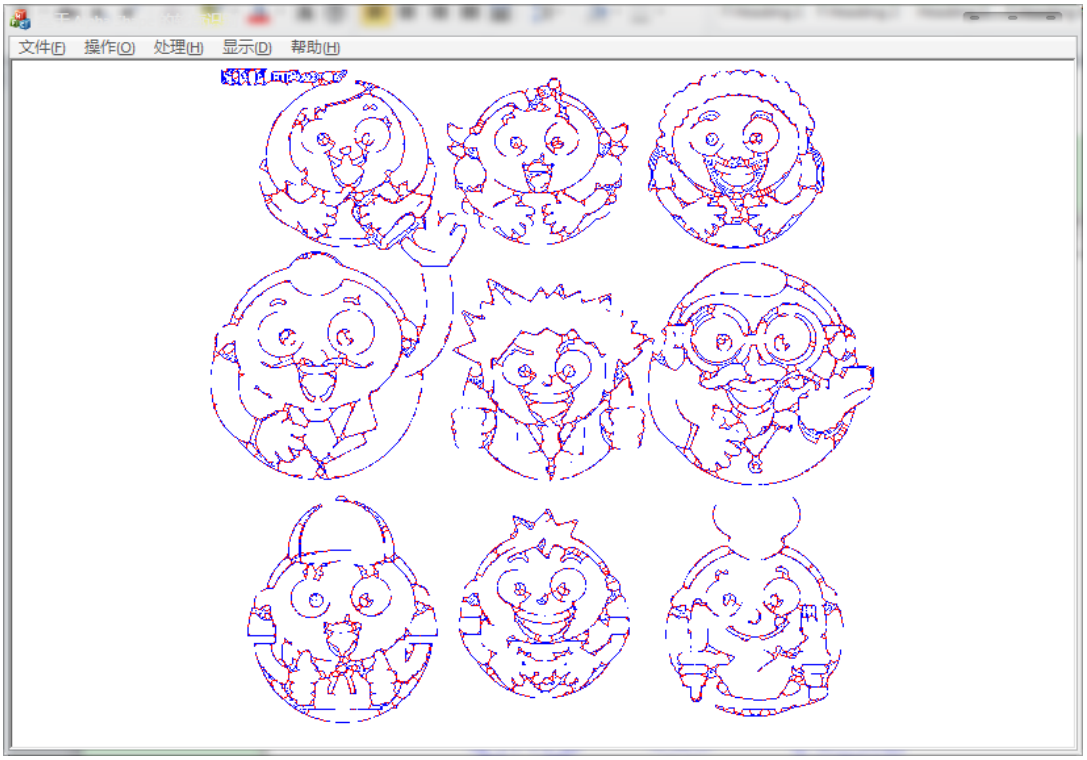
下面具体说明下“处理”菜单下的“计算轮廓线”和“去除特征点”功能的效果。

首先，导入图片，并选择“显示”菜单下的“全屏居中显示”功能。

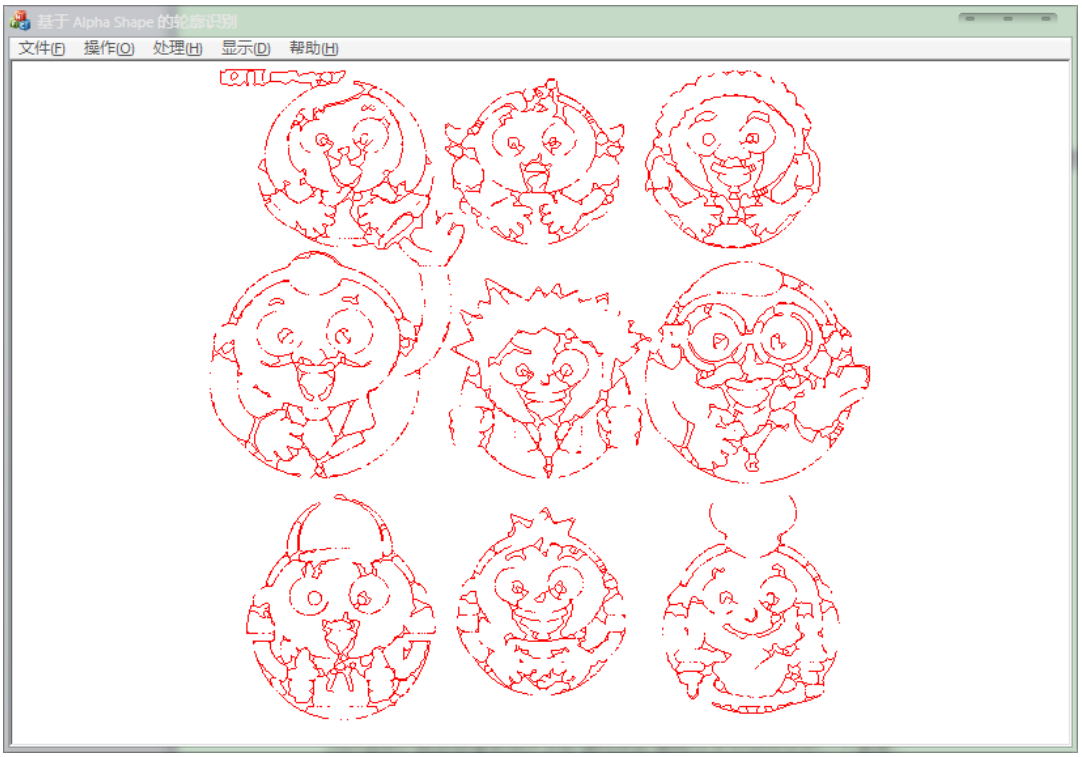


这时候我们能看到的是所导入图片的轮廓特征点。

然后，选择“处理”菜单下的“计算轮廓线”功能。



这时候我们就能够看到图片的轮廓线和轮廓特征点共同存在的一个画面。
最后，为了更清楚的看到轮廓线，我们可以选择“处理”菜单下的“去除特征点”功能。



这样我们就得到了最终的只有图片轮廓线的画面。

2.3.2 相关鼠标操作对应的快捷键（不区分大小写）

i/I 导入
o/O 导出
p/P 平移
y/Y 原图大小显示
q/Q 全屏显示图片
l/L 计算/去除轮廓线

t/T 显示/去除特征点
滚轮 放大/缩小

3. 小组成员以及工作分配

雷挺：主要负责文档撰写、测试用例的生成以及系统结合测试。
圣长军：主要负责应用程序框架搭建以及用户界面操作的设计与实现。
杨安宁（组长）：主要负责图片特征点的提取算法实现以及 AlphaShape 生成算法的实现。