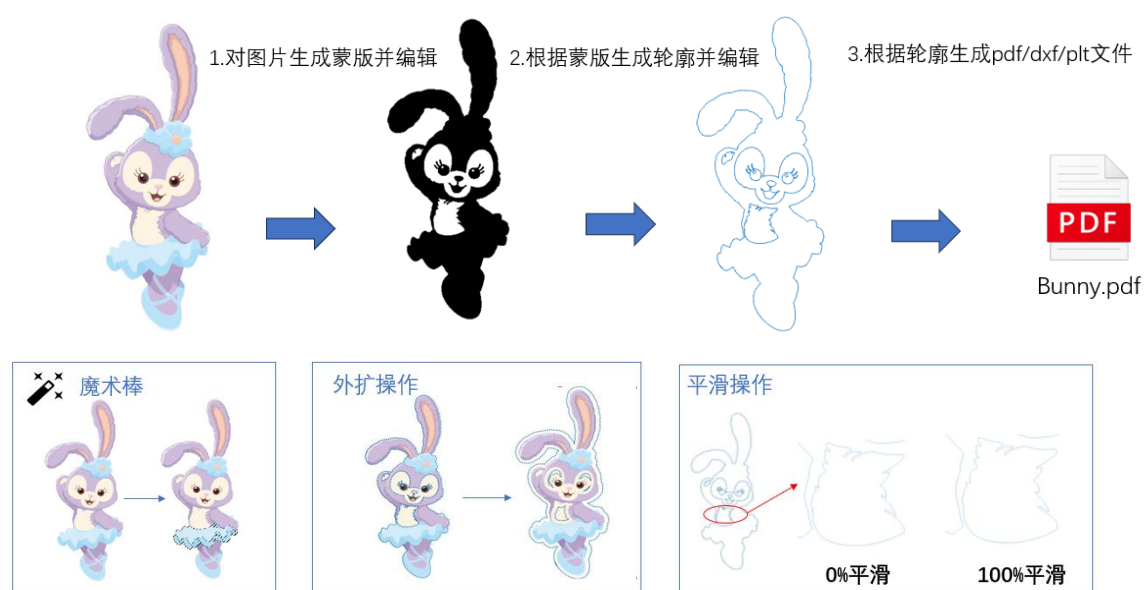


巡边软件设计文档

1.引言

巡边软件（简称软件）用来识别图片中的轮廓，同时支持轮廓编辑，最后生成用于打印切割系统的文件格式(如 PLT、PDF、DXF)。



巡边软件设计文档用于详细说明软件的架构、模块设计、界面、数据流程等。本文档供软件用户、需求提供人员、软件开发人员、软件测试人员使用。

*更多需求细节请参考《巡边软件需求文档-v3》

2.界面设计

初步的界面设计如下图所示，主要分为两个部分（1）容差蒙版与轮廓追踪 界面 （2）轮廓编辑 界面



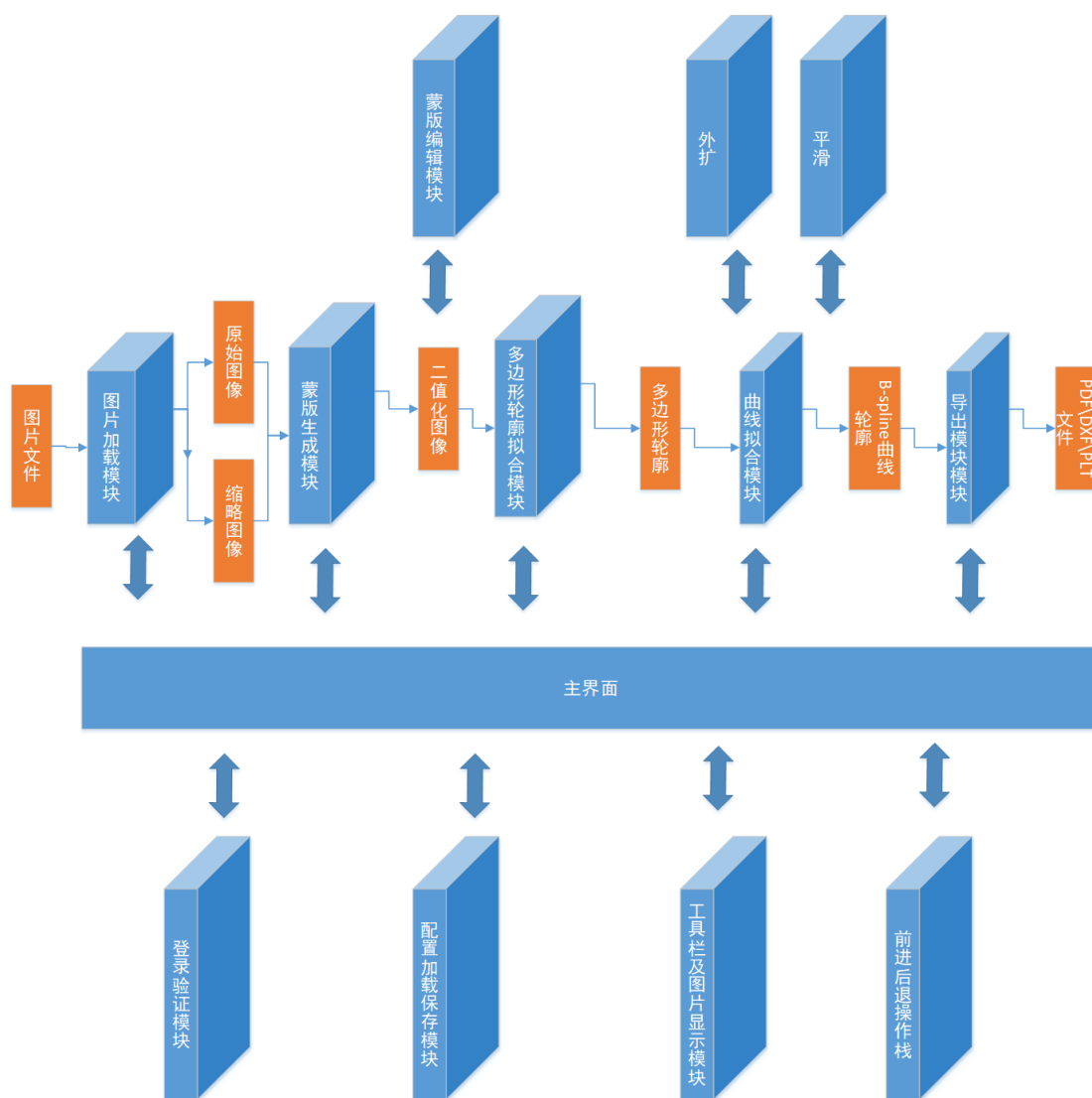
图 2-1 容差蒙版与轮廓追踪 界面



图 2-2 轮廓编辑 界面

3.系统架构

系统架构如下图所示：



流程上：本软件围绕巡边需求构建了一个处理流水线，数据流程为 图片->二值化蒙版图像->多边形轮廓->曲线轮廓->pdf/dxf/plt 文件。

结构上：本软件围绕主界面为骨架，各个模块嵌入主界面中。各个模块较为独立的完成自己的功能。各个模块可以较为独立的开发。其中较为耦合的模块有[蒙版编辑模块]与[工具栏与图片显示模块]

4.算法与数据结构

对于涉及复杂算法或数据结构的部分，提供详细的解释和说明。可以使用伪代码、流程图等来说明算法的执行过程。

4.1 数据结构

4.1.1 图片

使用 QT 的 QImage 统一存储图片

```
C++  
class QImage
```

4.1.2 多边形轮廓

点的定义如下

```
C++  
struct Point{  
    float x,y;  
}
```

一条多边形轮廓的定义如下

```
C++  
typedef std::vector<Point> Contour;
```

整体的多边形轮廓

```
C++  
//树状结构  
  
struct Block{  
    int id;  
    int fatherId;  
    vector<int> childId;  
    vector<int> broId;  
    Contour external;  
    std::vector<Contour> internal;  
}  
vector<Block> Blocks;
```

4.1.3 B-spline 曲线

```
C++
struct BSplineCurve{
    static const int k = 3;
    bool isClose = false;
    bool isUniform = false;
    Contour controlPoints;
    Contour valuePoints;
    std::vector<float> knotVector;
}
```

4.1.4 Bezier 曲线

```
C++
struct BezierCurve{
    static const int k = 3;
    Contour controlPoints;
    Contour valuePoints;
}
```

4.2 算法

4.2.1 多边形拟合算法

1) 区域追踪

输入二值图像进行轮廓追踪，根据给定的起始位置开始遍历图像内的像素：

- 如果像素值为 255，表示背景，则继续处理下一个像素。
- 如果像素值为 0，表示前景（轮廓部分），分三种情况处理：
 - a. 若当前像素的标签为 0，且上方像素为背景（255），则为外部轮廓的起始点，调用 `Trace_Contour` 函数进行外部轮廓追踪，将结果存储在 `blobs` 结构中。
 - b. 若下方像素为背景（255），且下方标签为 0，表示存在内部轮廓，调用 `Trace_Contour` 函数进行内部轮廓追踪，将结果存储在对应的外部轮廓的 `internal` 属性中。
 - c. 若当前像素的标签为 0，但周围存在已标记的像素，则将当前像素标签设为与左边像素相同的标签，以保持连接性。

2) 轮廓追踪：

1. 将起始点 (x, y) 的标签设置为当前标签 `current`，表示这个点属于当前轮廓。
2. 进入一个循环，循环条件为 `!done`，
3. 首先将当前点 (x_0, y_0) 添加到轮廓数据结构中。
4. 在顺时针方向上搜索相邻的像素点，根据轮廓方向选择不同的搜索顺序。如果找到一个黑色像素点（表示轮廓的一部分），则将该点标记为当前标签 `current`，并更新当前点 (x_0, y_0) ，并继续搜索。如果找不到黑色像素点，则将当前标签设为 -1，跳出循环。
5. 如果没有找到黑色像素点，说明当前点是孤立点，将 `done` 设置为 1，结束轮廓追踪。
6. 计算下一个起始位置，即更新搜索方向。

4.2.2 曲线拟合算法

1) LSPIA 算法

思想：通过调整控制点的位置来逐步逼近数据点，最终得到拟合曲线或曲面。

初始状态：给定初始控制点的数量和位置构造初始曲线。初始控制点位置与最终结果无关，只影响拟合速度；初始数量影响拟合结果，点数过少会导致难以拟合，点数过多拟合速度变慢。

迭代过程：在每次迭代中，每个控制点进行调整，其调整向量是数据点与当前迭代拟合曲线上对应点之间的差向量的加权和，并乘以 μ 系数进行步进， μ 的值影响收敛速度和收敛结果， μ 值过小会导致收敛速度慢， μ 值过大可能导致不能收敛，每条轮廓曲线可根据权重矩阵进行计算不同的 μ 。

结束迭代：计算前后两次迭代的误差值的差的绝对值，若小于误差阈值则认为迭代结束。

4.2.3 多边形简化算法

通过对初始轮廓的顶点进行删除或者合并操作，减少轮廓的顶点数，最终形成与初始轮廓形状相近的多边形轮廓。

预定采用的多边形简化算法：

1) smartDecimation 算法

思想：根据简化目标顶点数对原始点集进行复制扩展，然后对扩展后的点集按顺序取均值得到新的顶点位置。

初始处理：将原始轮廓中的点复制 N 份（如 p_0p_1 复制 3 份---> $p_0p_0p_1p_1p_1$ ， N 为想要的简化结果点数）。

新顶点计算：按顺序每次从复制的轮廓中取出 M 个点取均值作为多边形的新顶点 (M 为原始轮廓的点数)。

2) QEM 算法

思想：以每条边合并后新顶点离该边以及其邻边的距离作为简化此边的成本，每次选取简化成本最小的边进行简化并计算新的顶点位置，距离超过阈值或者当前顶点数达到要求则停止简化。

初始处理：计算每条边的简化成本和对应的新顶点并放入最小堆。

迭代过程：每次从最小堆中取出简化成本最小的边进行简化，更新边的邻接关系，更新邻边的简化成本和对应新顶点并放入最小堆。

结束迭代：当最小简化成本大于阈值或者当前顶点数达到要求则结束迭代。

3) minArea 算法

思想：以每条边合并后新多边形的面积与原来多边形面积之差以及新顶点离该边的端点的邻接点连成的线段的距离作为简化此边的成本，每次选取简化成本最小的边进行简化并计算新的顶点位置，距离超过阈值或者当前顶点数达到要求则停止简化。

初始处理：计算每条边的简化成本和对应的新顶点并放入最小堆。

迭代过程：每次从最小堆中取出简化成本最小的边进行简化，更新边的邻接关系，更新邻边的简化成本和对应新顶点并放入最小堆。

结束迭代：当最小简化成本大于阈值或者当前顶点数达到要求则结束迭代。

4) douglas-peucker 算法

思想：将轮廓分割为两个部分，对每个部分取离割线距离最远的点作为多边形顶点，选取的点通过与割线的垂线继续将轮廓分割进行递归处理。距离超过阈值则停止划分。

初始处理：找出多边形中离起始顶点 A 距离最远的点 B ，两点连成多边形的弦将其分为两个部分。

迭代过程：对每一部分，得到轮廓上离该直线段距离最大的点 M ，计算其与 AB 的距离 d 。如果距离大于阈值，则用 M 将曲线分为两段 AM 和 BM ，继续对两段进行迭代。

结束迭代：如果 d 小于给定的阈值，则停止划分。当所有部分都处理完毕时，依次连接各个分割点形成简化后的多边形。

4.2.4 多边形外扩算法

5.模块设计

对系统中的每个模块进行详细说明。包括模块的功能、输入和输出，以及与其他模块的交互方式。可以使用类图、流程图等来展示模块之间的关系和数据流程。

5.1 图片导入模块



图 5-1 图片导入功能图标

进行图片导入

图片文件支持格式：**JPEG\Jpg\TIFF\Tif\Png\Bmp**

支持在启动软件时传入参数导入图片，也支持在软件运行时选择待巡边的图片。

需要支持**拖拽**加载图片

为了保证运算速度，图片在导入时候需要根据原图片大小缩小到合适尺寸。

5.2 二值化蒙版生成模块

容差设置和蒙版编辑将原图像转换成二值化图像，该图像用于轮廓追踪模块的输入。



图 5-2 容差设置的界面 (红框处)

5.2.1 容差设置

以白色为基，像素颜色值与白色之差小于阈值则不选中，反之则选中(容差为 0 选择全部)。

选中部分赋值颜色 A，未选中部分赋值颜色 B 生成蒙版图像。

- 默认初始容差值为 16 (0~200)，或为上次用户所设容差值。



需要拖动容差值时，实时看到容差生成的模板。

5.3 蒙版编辑模块

在视图预览窗口对容差设置生成的蒙版区域进行交互修改。这部分由于涉及到对图像的编辑交互，与“工具栏及图片显示模块”较为耦合。



5-4 蒙版编辑工具界面（红框处）

在视图预览窗口对容差设置生成的蒙版区域进行交互修改。

- 点击蒙版编辑按钮后开始编辑。

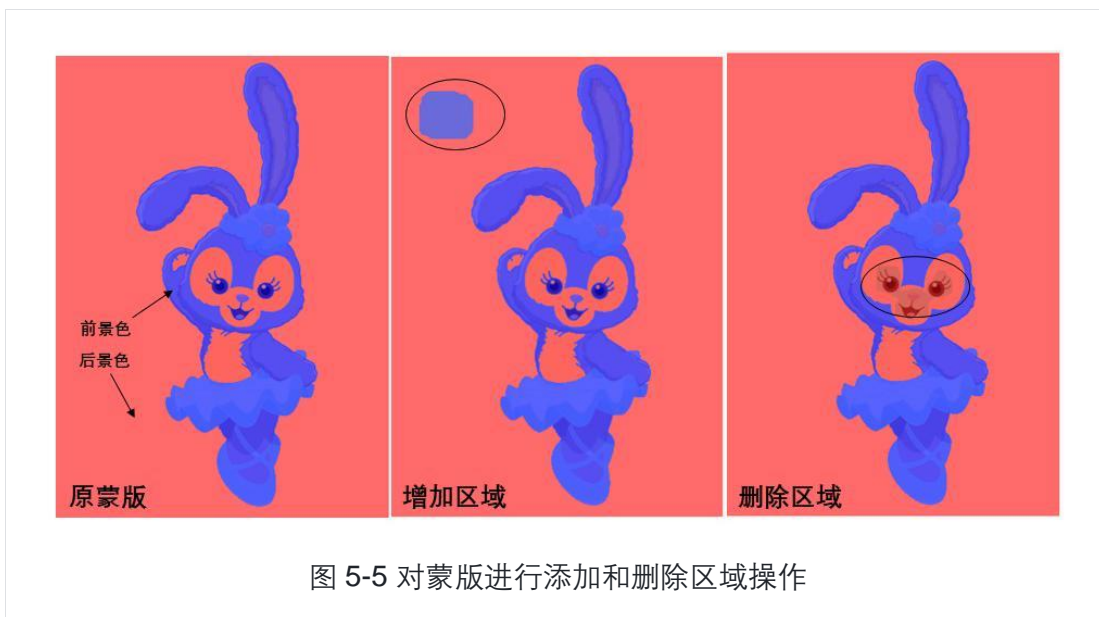


图 5-5 对蒙版进行添加和删除区域操作

选取工具

用选取工具对视图中的蒙版图层的区域进行选取：

- 笔刷：用涂刷的方式选取区域，视图放大时笔刷大小保持不变；
- 矩形：用矩形框选的方式选取区域；
- 椭圆：用椭圆框选的方式选取区域
- 多边形（套索）：通过鼠标画出一个封闭区域进行选取；
- 钢笔：用曲线方式选择区域。也要能够在原图上画线。
- 魔法棒：用于快速选取区域，详情见下一条目。
- 支持加选、减选、交叉，支持连续和反选。

魔法棒

需要实时的进行预览，供用户查看效果。

首先设置魔法棒容差（默认容差 32），通过鼠标点选像素后选取与其颜色差小于容差的连通区域的像素，可以选择加选、减选、交叉，连续和反选。可以删除选择的区域颜色。



图 5-6 魔术棒效果示意图

前进/后退

对用户自定义选取的区域执行撤销，前进操作。

画笔

可以涂上上色，通常闭合没闭合的边缘。

钢笔

抠图或者闭上没闭合的边缘。

多边形 套索

套索抠图。

橡皮擦

对用户自定义选取的区域执行擦除操作。

- 橡皮擦同笔刷进行涂刷的方式进行擦除。

以上抠图工具，都需要支持选定的区域，可以填色，也可以删除背景色。

我们软件面板需要有几个固定的颜色，比如红、黄、蓝、黑，白，透明，允许用户进行选择填色。

重置

清除用户的所有自定义区域选取，返回原来容差设置生成的蒙版。

前景色/后景色

切换模式，是增加选中区域还是删除选中区域：

- 前景色：添加选中区域，之后选择的区域为添加的区域，颜色与容差蒙版选中区域颜色类似（蓝/黑）；
- 后景色：删除选中区域，之后选择的区域为删除的区域，颜色与容差蒙版未选中区域颜色类似（红/白）。

5.4 工具栏及图片显示模块



图 5-7 工具栏及图片显示模块的界面

5.4.1 透明图片

需要支持透明度图片自动识别，需要有提示的标题，透明或不透明。

5.4.2 视图预览

对原图与蒙版的视图进行平移、放缩、不透明度等操作。

- 用户导入图片后自动弹出此视图窗口，初始视图为原图

视图

- 视图分为原图视图、复合视图、蒙版视图；
- 复合视图为原图和蒙版的复合，其中蒙版图层选取区域为蓝色，未选取区域为红色；
- 蒙版视图中选取区域为黑色，未选取区域为白色；
- 将导入图片缩放至刚好能放入窗口，边缘留一定空白；
- 用户可以在三个视图之间进行切换预览。



图 5-8 软件需要支持显示三种蒙版生成视图（原图、复合、蒙版视图）

缩放

以鼠标位置为中心缩放视图，并将鼠标位置的视图内容平移到视口的中心：

- 鼠标滚轮对视图放大缩小；
- 选择左侧工具栏的按钮进行缩放。

平移

当视图尺寸大于视口时，需要移动来查看视图的不同位置：

- 通过水平方向和垂直方向的滑动条来移动画布；
- 通过鼠标拖拽来移动视图。

不透明度

通过不透明度滑块设置复合视图中蒙版图层的不透明度。

- 默认不透明度为 150 (0~255)，或为上次用户所设不透明度。

5.5 多边形拟合模块

用容差蒙版模块产生的二值化图像进行输入，进行轮廓提取，并对轮廓点集进行多边形拟合得到多边形点集，最后对多边形点集用三阶贝塞尔曲线进行拟合形成多段连续贝塞尔曲线轮廓。



5-9 多边形与曲线拟合模块的界面

轮廓追踪部分需要注意在进行轮廓提取的时候，要有进度条，待处理的图片有一个从上到下或者从左到右扫描的效果（如图 3-4 所示）。

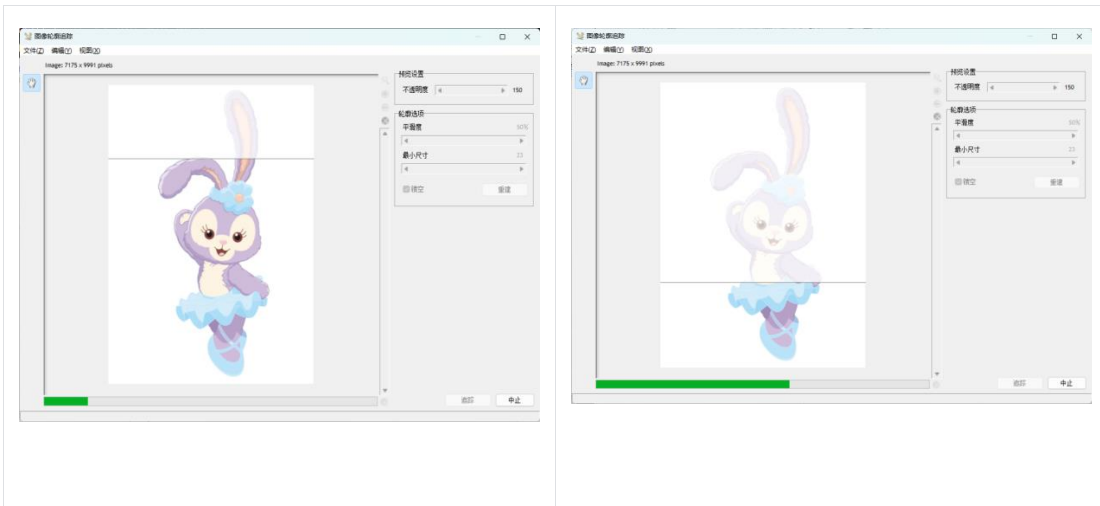
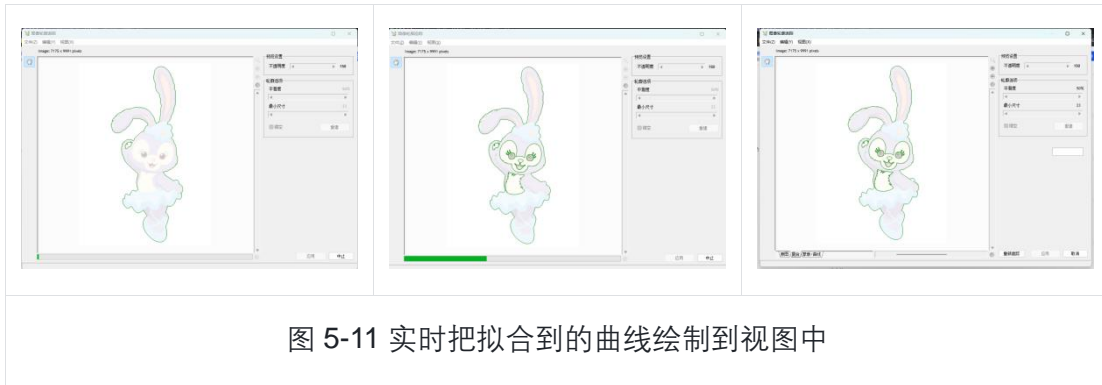


图 5-10 扫描效果与进度条

最后生成轮廓时要实时的把当前生成的曲线绘制到视图中（如图 5-11 所示）。



1. 轮廓提取

对二值图进行轮廓边缘提取，得到对应轮廓对离散点集。

- 内外边缘提取（镂空）。

使用扫描线方法对二值图进行边缘提取，得到内外的轮廓点集。

- 外边缘提取。

边缘提取过程中仅提取外部边缘，得到外轮廓点集。

2. 节点数目控制

用多边形近似算法，以多边形近似轮廓点集，根据追踪方式的不同从轮廓点集中选择不同数量的点集作为多边形轮廓点集。

- 低质追踪：追踪相对较少的点；
- 高质追踪：追踪相对较多的点。

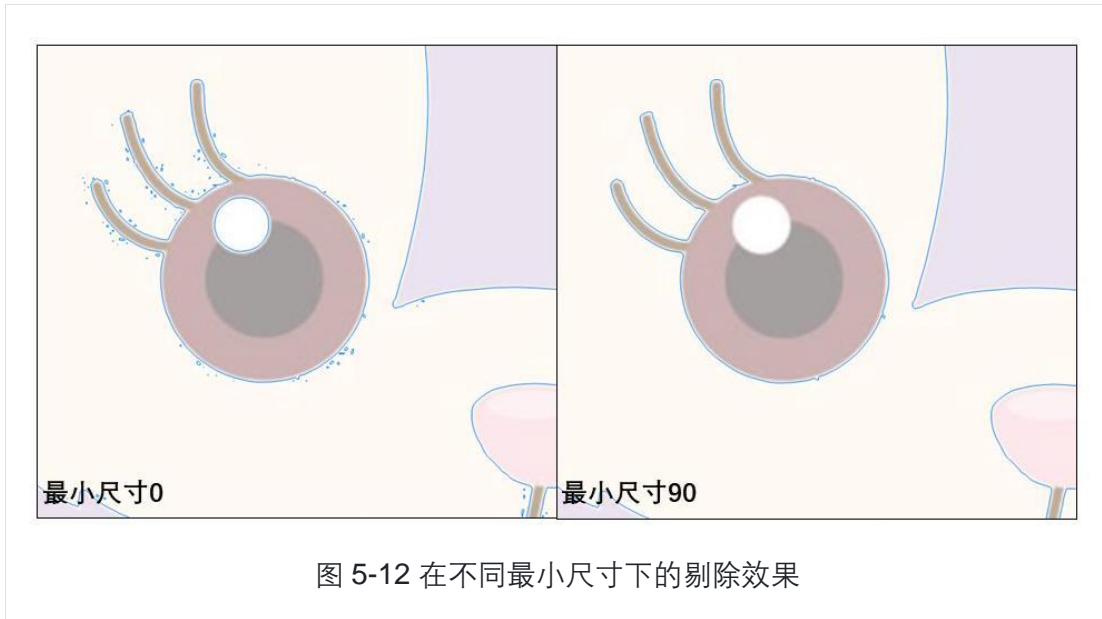
5.6 曲线拟合模块

根据追踪后得到的多边形轮廓点集进行拟合，计算每个点对应的两个控制点，用于连接相邻的两条曲线，然后通过得到的每段四个控制点绘制各段曲线。

最小尺寸

设置轮廓曲线中的最小尺寸，面积小于最小尺寸的轮廓区域剔除，暂定使用矩形包围盒计算面积。

- 通过最小尺寸滑块设置最小尺寸的值，默认最小尺寸 23（0~150）或为用户上次设置的值；
- 单位平方毫米。



5.7 曲线外扩

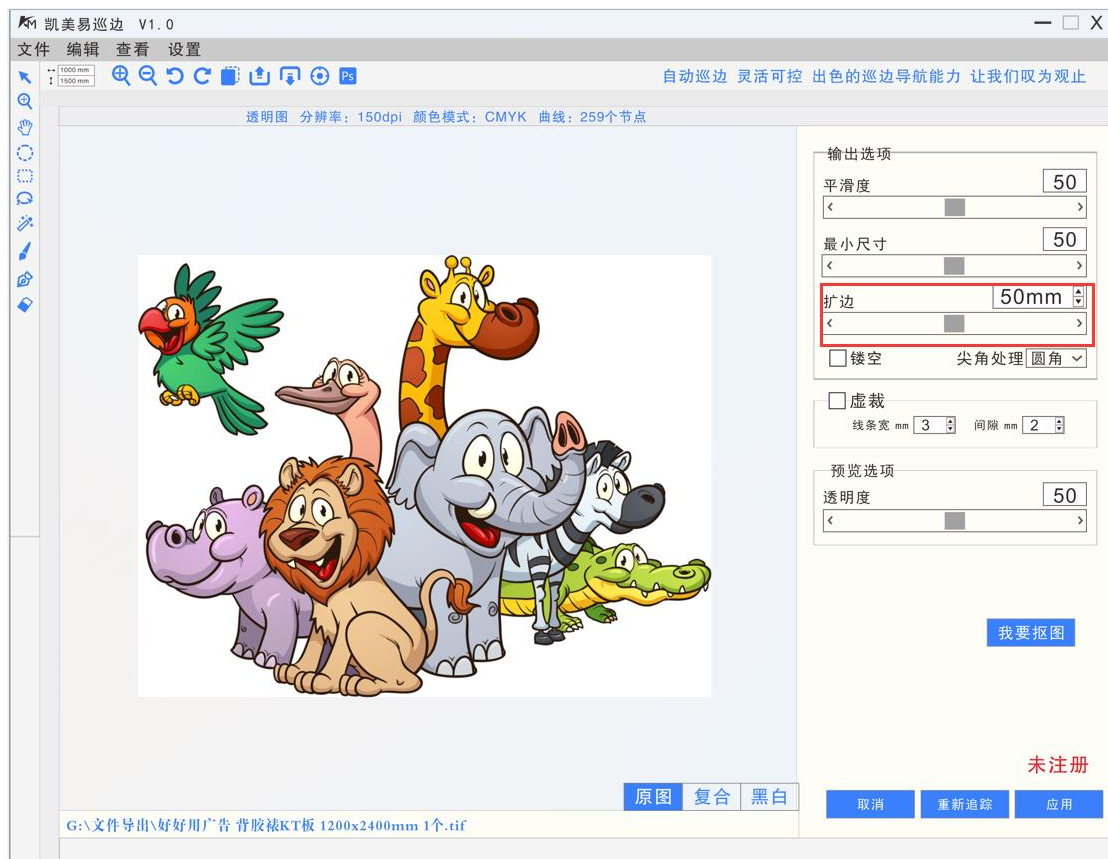


图 5-13 曲线外扩的界面

- 将轮廓上的点根据指定的扩展距离（偏移量，默认值为 0mm）向外（法线方向）扩展；
- 曲线修正：对曲线相交区域进行合并，得到新的曲线轮廓点集。



图 5-14 外扩操作的效果图

5.8 曲线平滑

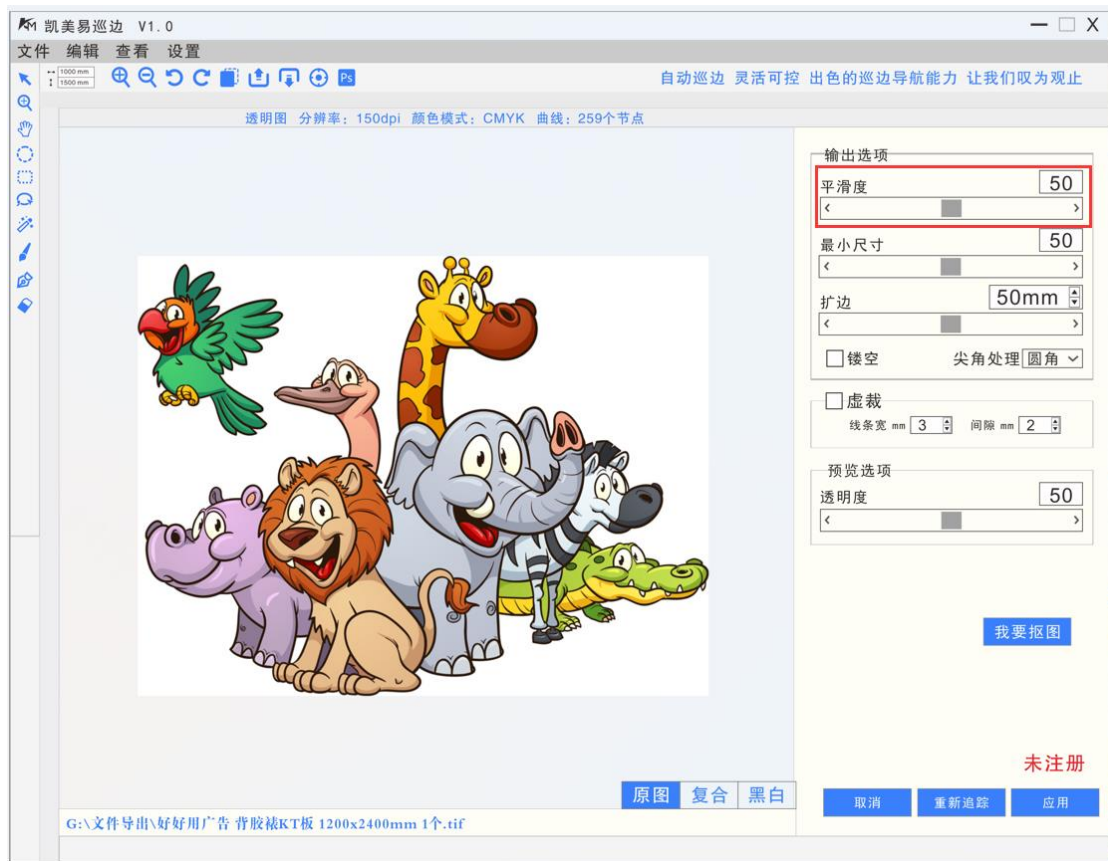
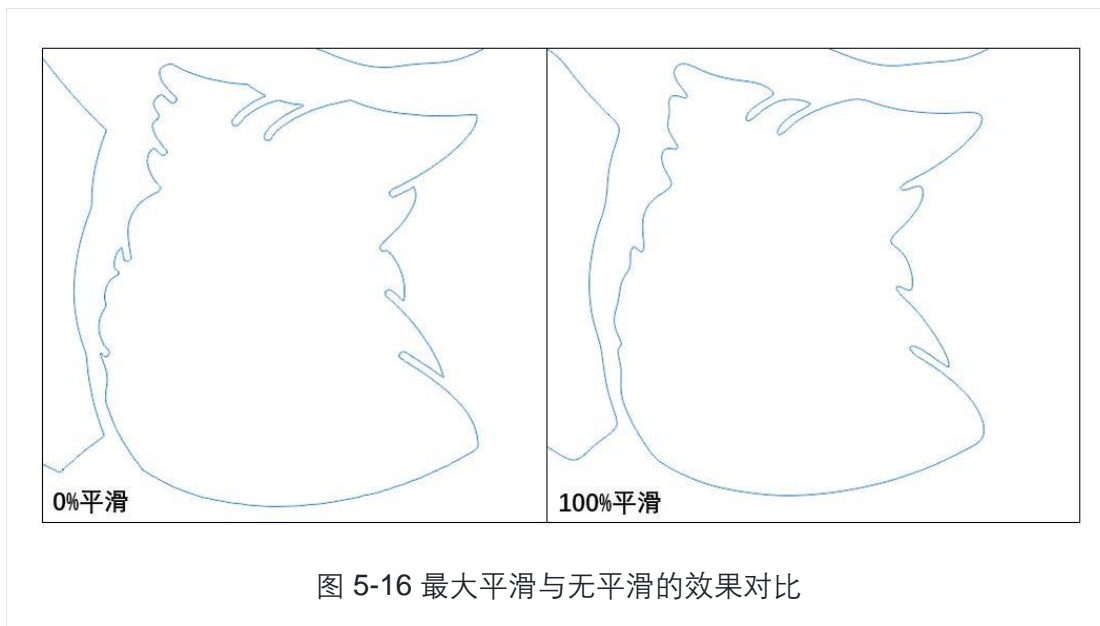


图 5-15 曲线平滑界面

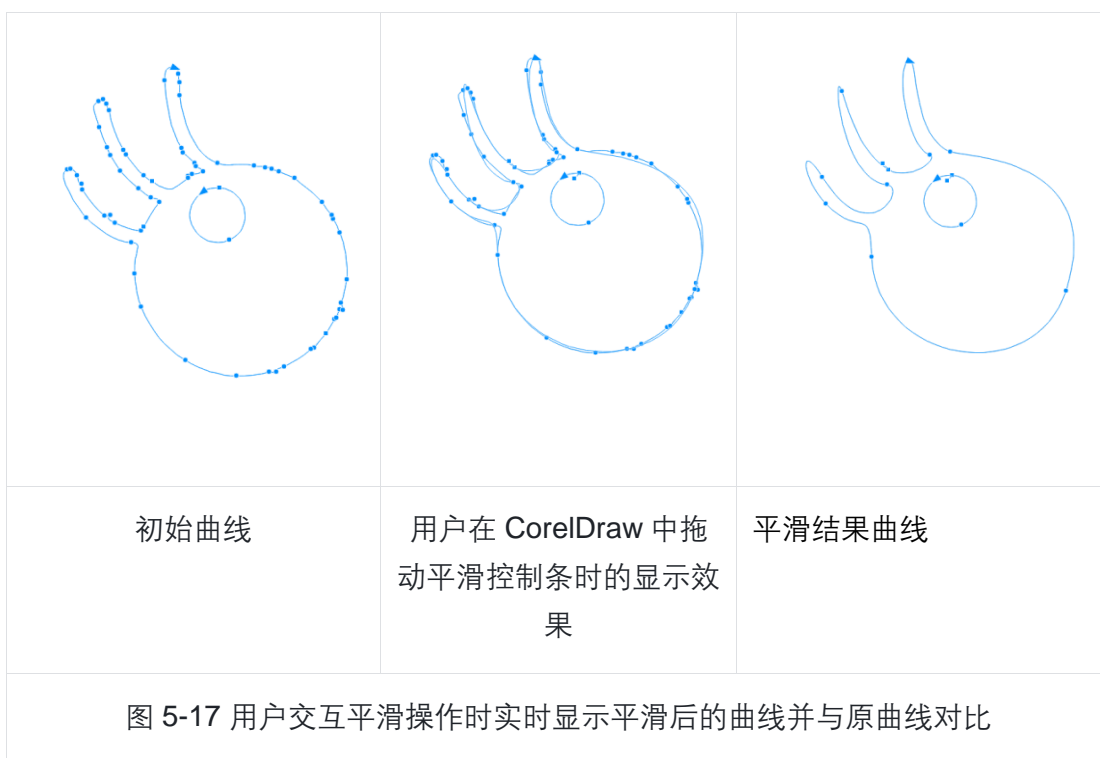
对轮廓进行平滑操作，根据平滑度设置的值进行相应程度的平滑。

- 默认平滑度为 50%，或为用户上次设置的值；

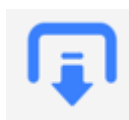
- 通过平滑度滑块设置平滑度的值；
- 有默认的平滑阈值，防止平滑时出现较大的变形。



实时给用户预览平滑结果和原始图片的对比，参考 CorelDraw 的平滑操作效果：



5.9 导出模块



导出格式：PDF/DXF/PLT 三种。

生成的导出文件使用固定的命名方式放置在软件的根目录中，也可以在软件运行时手动指定导出位置；

优先支持 PDF 与 DXF 格式。

导出格式的路径需要支持虚线[?]，可以指定虚线的间隙。

尖角处理，圆角、直角、斜切角

5.10 登录验证模块

软件需有登陆的窗口和一个注册按钮，可以输入账号和密码，还有需要获取电脑的主板信息，登录成功后把账号和密码记录在配置文档内供下次使用，同时把登录返回的信息包括软件的项目名称、标题、声明、联系我们等信息写入配置文档，供软件使用，每次使用软件都需要登录，除了初次登录之外，其他登录采用后台多线程登录机制，如果验证失败，强制的返回登录窗口，每次登录成功都需要把最新返回的信息写入到配置文档，软件需要加入防破解机制，①需要使用 VMP 加密，②主要模块循环体的初始值，都要从云端获取，具体数值需要我们约定好。

5.11 配置加载保存模块

设置

用配置文件保存用户上次设置的参数。

- 开始时读入配置文件中的参数，将各部分的默认参数修改为文件中的参数；
- 结束时将用户设定的巡边参数写入配置文件

5.12 前进后退操作栈



6. 与其他软件交互接口

1. 默认的交互接口

- 启动本软件前，调用巡边软件的程序将待巡边图片存放在本软件根目录下，并且以固定名称命名；

- 本软件得出轮廓后，生成的导出文件也以固定名称存放在本软件的根目录下；
- 调用本软件的程序可以定时轮询查看本软件进程是否终止运行，如果本程序终止说明导出文件已经生成或者出现异常。
- 可以交互包括有输出路径，输出格式，文件名，和定位孔的位置以及大小和类型。
- 交互包括有尺寸大小、像素，以及巡边路径的位置偏移值

2.与排版软件的接口

排版软件也使用默认交互方式与本软件进行交互，但是需要注意的是，当排版软件导出的图片有定位孔时，需要通过某种方式把定位孔去掉再进行巡边。生成的结果导入排版软件中后再添加定位孔。

我们的排版软件和巡边软件需要做好开发的对接接口，巡边结果需要载入到排版软件以及添加定位孔，实现所有的交互对接需求，排版软件可以一键提取雕刻路径。排版软件需要支持单张图和多张图巡边，如多张图，我们所说的是整版巡边，那么我们有两种处理办法，第一种，那么就用自动分辨率自动导出图片后载入到巡边软件，巡边完成，自动载入巡边路径到排版软件。第二种，遍历所有图片，逐一的送入巡边软件，每送入一张，巡完成，路径载入排版软件，再送入下一张，再巡边，依次处理，直至完成全部。

7. 其它

- 需要支持添加定位点
- 日志系统 报错系统

内存问题

- 实体-关系图设计

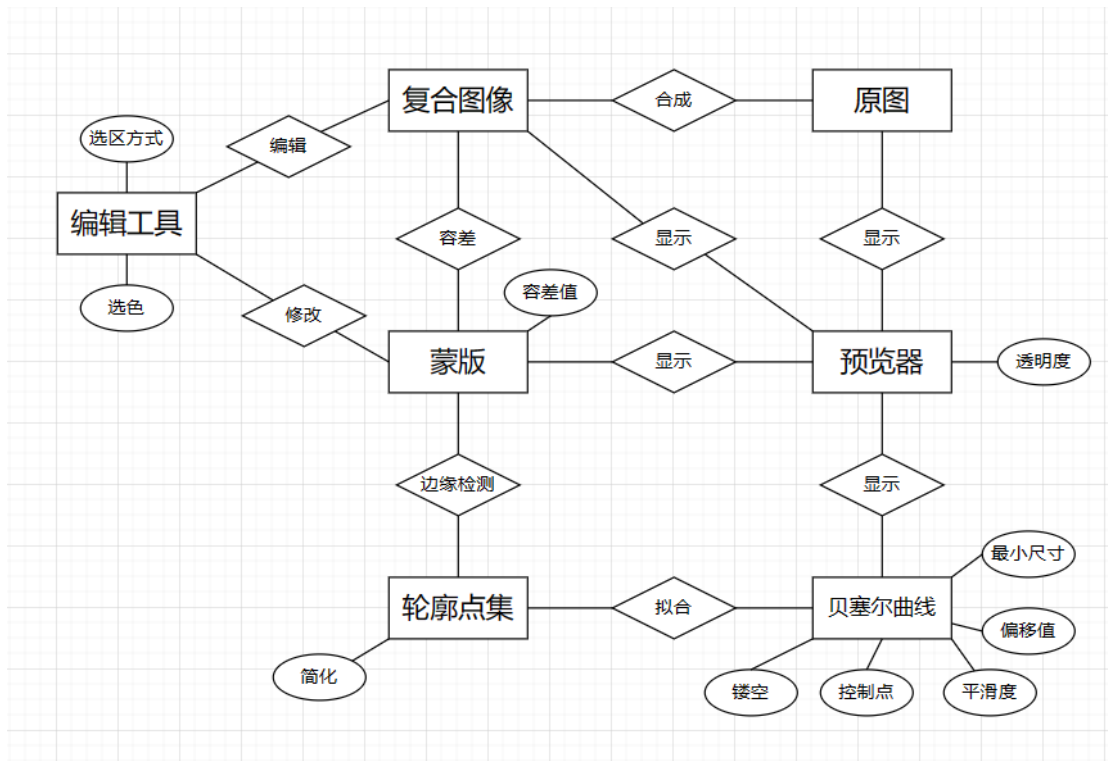


图 4-3 巡边软件的实体-关系图

8.性能优化

提供优化策略，包括代码性能优化、数据库查询优化等。

9.未来扩展

提供关于未来可能的扩展、功能增加等方面的考虑。

10.参考文献

引用所参考的技术资料、文档等。