

专业课程实验报告

课程名称： 交互式媒体

实验教师： 王峻

计算机与信息科学学院 软件学院

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目名称 | | 创作第一个网页 | | | |
| 实验时间 | | 2018.11.10 | 实验类型 | | □验证性 ☑设计性 □综合性 |
| 1. 实验目的   图像的基本处理算法的编程实现与交互式演示界面设计   * 1. 编程实现图像处理基础经典算法，并集成到一个交互演示界面中   2. 可以通过界面选择相应的算法进行演示，可以通过文件选择进行算法所处理的图像的输入输出，可以在界面上进行参数选择，以及算法结果图像展示（要求在界面上同时展现输入图像和输出图像，便于比对）   实现：  1、256色转灰度图  2、二值化变换  3、图像缩放  通过参数框设置图像的缩放比例，实现在屏幕上对输入图像的缩放。  4、图像旋转  通过参数框设置图像的旋转角度，实现在屏幕上对输入图像的旋转   1. 实验要求   （1）掌握图像基本处理算法的工作流程，我们在实验中只能涉猎其中很小的一部分，希望大家能够在实验之外自学理解相关算法；  （2）掌握应用高级语言实现基本交互演示界面的技能； | | | | | |
| 1. 实验内容与设计（主要内容，操作步骤、算法描述或程序代码）   整体框架：  使用C# windows窗体应用程序设计交互式界面，可以直接在“工具箱”里找到相应控件并设置代码。界面由两个pictureBox和实现各个功能的buttons（图像处理、保存、打开按钮）以及需要用户输入的参数的textboxes。左边的pictureBox显示打开图片，右边显示处理后的图片。     1. 进入不同控件代码，在Form1.cs中设置不同控件发生的事件。      1. 从头开始编码，形成的Form1首先要初始化：使用InitializeComponent();因为要在打开文件按钮接受文件路径，所以pathname直接定义在整个class Form1:Form中以便后面其他控件的函数使用。 2. 第一个按钮是需要用户打开文件，通过file的Filename函数得到路径并在pictureBox1中显示，pathname被赋值。 3. 保存按钮代码比较简单，直接在pictureBox2中使用image.save      1. 以下开始实现图像处理功能。首先是彩色图转灰度图：使用了整数方法；获得原图的RGB后带入计算可得所需图像的RGB。输出时仍然使用之前用过的pathname获得图像参数然后传入此函数。      1. 二值化：迭代法计算阈值。该算法先假定一个阈值，然后计算在该阈值下的前景和背景的中心值，当前景和背景中心值得平均值和假定的阈值相同时，则迭代中止，并以此值为阈值进行二值化。此方法先获得了直方图，取得最大灰度值和最小灰度值，再分别将图片分为两个大部分进行迭代求各部分自己的阈值，最后将所得的两个阈值将求平均数并二值化          1. 图像缩放：使用了双线性插值的方法得到缩放图片。在界面中有两个textbox分别要用户输入新图片的高与宽，在点击“缩放”这个控件时直接将这两个数据传为参数即可。   缩放代码中新图片需要获得各个像素并赋值。所得图片的像素需要靠旁边的四个点，根据这四个点与本点的距离的关系计算目标图像中(x,y)一点的像素值。通Mat.at<Vec3b>(i,j)得到图片位置i，j像素值，循环得到每个像素的RGB值（体现为三维数组）时并一个个赋值给新图片的Vec3b value，即像素值。针对（x0,y0）value(x0,y0)=value(x2,y2)\*s1  +value(x1,y2)\*s2+value(x1,y1)\*s3+value(x2,y1)\*s4; 其中s1-s4是通过代码中的偏移量得到的。         1. 图像旋转：图片的旋转需要用户输入角度，传参过程同上。图像旋转的原理就是利用新的坐标和旧坐标在cos, sin之间关系。新点（x,y）与旧点(x0,y0)和须按转角度a之间的关系为：   x =x0cosa+y0sina；  y =-x0sina+y0cosa；  首先这里分为两个函数，因为要获得图像中心，旋转是以图像中心为远点的；第一个函数先得到图像旋转的所占区域，在第二个：得到最终图片的函数中，定义的函数宽高就要以旋转后的宽高为准c#的窗体应用程序中本身的  using System.Drawing;  using System.Drawing.Drawing2D;  using System.Drawing.Imaging;  中包含了graphics类型，其中包括了处理图像较多方法，所以在try中主要用它们处理图像。要让graphics围绕某矩形中心点旋转N度，分三步：第一步，将graphics坐标原点移到矩形中心点,假设其中点坐标（x,y）；第二步，graphics旋转相应的角度(沿当前原点)；第三步，移回（-x,-y）。先将画布坐标原点放在中心，然后旋转，最后将图片平移。因为原图像的坐标原点不再中心，所以得到两点偏移量再重合，最后绘制。 | | | | | |
| 1. 图像平移：传参过程同上。基本算法：使用获得像素值的方法得到新图片数据，背景色设为白。   new Rectangle进行初始化，ImageLockMode锁定图像以便于改写像素。新建两个byte数组存入像素数据，通过Stride扫描图片得到新的像素值并传给新图片，最后将所得数据全部拷贝放入系统内存。       1. 测试数据和执行结果 （在给定数据下，执行操作、算法和程序的结果，可使用数据、图表、截图等给出） | | | | | |
| 1. 实验结果分析及总结（对实验的结果是否达到预期进行分析，总结实验的收获和存在的问题等）   **问题：**  1、picturBox里我设置的SizeMode都是Autosize，即 按打开图片本身的大小显示此图；如果设置Stretch，显示图片就会拉伸为设置的框大小，但是在做缩放平移和旋转时效果就不太明显，如果图片过大界面显示会比较混乱。  2、平移函数的缺陷很大。因为设置新图片大小和原图是一样的，这就造成了如果图片平移，空出来的背景就为白色。  3、 system.drawing中的graphics包含了许多图像处理的函数，但是很多处理用opencv处理会更好，虽然opencvsharp是专门给C#语言处理图像使用，但是C#中有一些重要的函数（像素获取和修改）不能直接使用一般的opencv方法，总体来说用C++/C会更适合使用opencv。  4、更好的处理方法是在每个图像处理后弹出新窗口，但是因为要对比所以这里放了两个pictureBox，也就造成了图像大小的限制。  5、 图像名.At<Vec3b>(i,j)[0/1/2]是得到图像像素数据最好的方法，但是不能用来赋值  **总结：**  第一次用C#窗体应用程序写。很多细节上的处理很重要，比如使用opencv的mat类型、用graphics还是用位图bitmap的方法处理图像；关于像素处理上也仍然有很多漏洞，最后赋值时总是报错，获取数据倒不难，但如何赋值和转换数据类型与系统的命名空间关系很大（包含所需方法）；尝试过用C语言和opencv包来写，但是C#语言上有一些方法是OpenCvSharp不包含的。  除此外还有界面设计，这次只是简陋的放了几个控件并且因为算法过于简陋在两个pictureBox上显示图片效果非常不好。图片尺寸比设置的控件更小时还比较合适，但是如果大了就会超出；可是如果设置SizeMode为Stretch时只会拉伸图片，在图像处理时更无法完全显示，也就无法对比；  一开始自己做不清楚用什么语言，本身打算使用python，图像算法上更加方便、准确，但是在交互式界面上还是C#更方便一些。C或C++虽然也可以但是需要使用MFC，开发难度比较大，并且C/C++使用命令行输入交互更方便。 | | | | | |
| 教  师  评  阅 | 实验内容和设计（含操作过程、算法或代码）（A-D）： | | |  | |
| 实验结果（A-D）： | | |  | |
| 实验分析和总结（A-D）： | | |  | |
| 实验成绩（A-D）：  反馈评语： | | | | |