重 庆 交 通 大 学

学生实验报告

课 程 名 称： 数 字 图 像 处 理 .

开 课实验室： 软 件 实 验 中 心 .

学 院： 信息科学与工程学院 年级 21级 专业 物联网1 班

学 生 姓 名： 杨琳 学 号 632107060432 .

指 导 教 师： 蓝 章 礼 .

开 课 时 间：2023 至 2024学年第 二 学期

|  |  |
| --- | --- |
| 成 绩 |  |
| 教师签名 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目名称** | | **第二周实验作业** | | | |
| **姓名** | **杨琳** | **学号** | **632107060432** | **实验日期** | **2024.4.1** |
| **教师评阅：**  1:实验目的明确□A□B□C□D  2:内容与原理□A□B□C□D  3:实验报告规范□A□B□C□D  4:实验主要代码与效果展示□A□B□C□D  5:实验分析总结全面□A□B□C□D | | | | | |
| **实验记录** | | | | | |
| 1. **实验目的**   完成图像的灰度化、二值化、亮度调整实验。  基本要求：输入彩色图像，通过自己设计的算法代码编写，实现输入图像的灰度化、固定阈值的二值化，并能进行亮度调整。  拓展要求：在基本要求的基础上，实现通道提取，可变化阈值的二值化，自适应二值化，能进行对比度、饱和度的调整。答案要有算法描述，核心代码，完成图片的效果。   1. **实验主要内容及原理**   1）图像灰度化：  每一张图都是由多个像素点组成的，一个像素点颜色的显示为RGB三通道模型，图像的灰度化，就是调整RGB的数值，主要展现其亮度的差别，RGB转换为灰度的公式：  2）图像二值化：  固定阈值的图像二值化，先将图片转换成灰度图片，及只考虑其亮度，然后再和设定的阈值进行比较，若大于则将该像素点赋值为全白，若小于则赋值为全黑。  3）亮度调整：  RGB其实是3个通道，每个通道值谨代表亮度，所以，如果要调整亮度，直接在三通道整体加上给定的亮度值。考虑边界条件0和255，作为亮度的约束。  4)通道提取：  获取某个像素点在某通道下的强度大小，然后将RGB三个通道都改为同一个通道下的强度。就可以在一个通道下获得该通道的不同灰度值。  5)可变化阈值的二值化：  可以使用直方图确定一个合适的阈值。    6）HSI调节:  H为色调，S为饱和度，I为强度。使用HSI转化RGB的公式：     1. **实验环境**   Window10  Visual Studio2019  **四、实验主要代码与效果展示**  1）图像灰度化  使用两个for循环遍历图片（可以是行优先，也可以是列优先），然后将每一个像素点进行灰度转换，即如下公式：  所有像素点转换完成后，更新图片框2的图片。   |  | | --- | | private void btnHD\_Click(object sender, EventArgs e)  {  if(pictureBox1.Image == null)  {  MessageBox.Show("错误，没有导入图片");  return;  }  label2.Text = "灰度化图片";  Bitmap bt = new Bitmap(pictureBox1.Image);  Bitmap bt1 = new Bitmap(pictureBox1.Image);  Color color = new Color();  for(int i = 0; i < bt.Height; i++)  {  for(int j = 0; j < bt.Width; j++)  {  color = bt.GetPixel(j, i);  int n = (int)((color.G \* 59 + color.R \* 30 + color.B \* 11) / 100);  bt1.SetPixel(j, i, Color.FromArgb(n, n, n));  }  }  pictureBox2.Refresh();  pictureBox2.Image = bt1;  } |     2）固定阈值图像二值化  先将像素点的值转换为灰度值，然后再和设定的阈值比较大小，如果大于就为白，小于就是黑。   |  | | --- | | private void btnBinary\_Click(object sender, EventArgs e)  {  if (pictureBox1.Image == null)  {  MessageBox.Show("错误，没有导入图片！");  return;  }  Color color = new Color();  double r, g, b, y;  Bitmap bt1 = new Bitmap(pictureBox1.Image);  int binaryThreshole = int.Parse(txtBinary.Text);  for(int i = 0; i < bt1.Width; i++)  {  for(int j = 0; j < bt1.Height; j++)  {  color = bt1.GetPixel(i, j);  r = color.R;  g = color.G;  b = color.B;  y = 0.59 \* r + 0.3 \* g + 0.11 \* b;  if(y > binaryThreshole)  {  Color ColorProcessed = Color.FromArgb(255, 255, 255);  bt1.SetPixel(i, j, ColorProcessed);  }  if(y <= binaryThreshole)  {  Color ColorProcessed = Color.FromArgb(0, 0, 0);  bt1.SetPixel(i, j, ColorProcessed);  }  }  }  pictureBox2.Refresh();  pictureBox2.Image = bt1;  } |     3）亮度调整  将遍历像素点，每个通道的值加上给定的值即可。   |  | | --- | | private void btnBrightnessadjust\_Click(object sender, EventArgs e)  {  if(pictureBox1.Image == null)  {  MessageBox.Show("错误，没有导入图片！");  return;  }  int value = int.Parse(txtLD.Text);  Bitmap bt = new Bitmap(pictureBox1.Image);  Bitmap bt1 = new Bitmap(pictureBox1.Image);  int r, g, b;  for(int i = 0; i <bt1.Width; i++)  {  for(int j = 0; j < bt1.Height; j++)  {  Color color = bt.GetPixel(i, j);  r = color.R;  g = color.G;  b = color.B;  r += value;  g += value;  b += value;  if (r > 255) r = 255;  if (r < 0) r = 0;  if (g > 255) g = 255;  if (g < 0) g = 0;  if (b > 255) b = 255;  if (b < 0) b = 0;  Color c1 = Color.FromArgb(r, g, b);  bt1.SetPixel(i, j, c1);  }  pictureBox2.Refresh();  pictureBox2.Image = bt1;  }  } |     4）通道提取  提取出该像素点在某一通道的亮度。然后将三通道都改为同一亮度。   |  | | --- | | private void btnRed\_Click(object sender, EventArgs e)  {  //只需要将RGB全部改成R就可以了  if (pictureBox1.Image == null)  {  MessageBox.Show("错误，没有导入图片！");  return;  }  Color color;  int r;  Bitmap bt1 = new Bitmap(pictureBox1.Image);  for (int i = 0; i < bt1.Width; i++)  {  for(int j = 0; j < bt1.Height; j++)  {  color = bt1.GetPixel(i, j);  r = color.R;  Color processed = Color.FromArgb(r, r, r);  bt1.SetPixel(i, j, processed);  }  }  pictureBox2.Refresh();  pictureBox2.Image = bt1;  } |     **5）**可变化阈值的二值化  先将图像像素转换为灰度，然后灰度255个等级统计出现的频率，查找最高频率点，其灰度等级作为二值化的阈值。   |  | | --- | | private void btnIBinary\_Click(object sender, EventArgs e)  {  if (pictureBox1.Image == null)  {  MessageBox.Show("错误，没有导入图片！");  return;  }  //先灰度化  int[] pixel = new int[256];  Bitmap bt1 = new Bitmap(pictureBox1.Image);  Color color = new Color();  for(int i = 0; i < bt1.Width; i++)  {  for(int j = 0; j < bt1.Height; j++)  {  color = bt1.GetPixel(i, j);  int grid;  grid = (int)((color.G \* 59 + color.R \* 30 + color.B \* 11) / 100);  pixel[grid]++;  }  }  //找最大值作为阈值  int max = pixel[0], site = 0;  for(int i = 0; i < 256; i++)  {  if(pixel[i] > max)  {  max = pixel[i];  site = i;  }  }  for(int i = 0; i < bt1.Width; i++)  {  for(int j = 0; j < bt1.Height; j++)  {  color = bt1.GetPixel(i, j);  int grid;  grid = (int)((color.G \* 59 + color.R \* 30 + color.B \* 11) / 100);  if (grid >= site) bt1.SetPixel(i, j, Color.FromArgb(255, 255, 255));  else bt1.SetPixel(i, j, Color.FromArgb(0, 0, 0));  }  }  pictureBox2.Refresh();  pictureBox2.Image = bt1;  }  } |     6）HSI调整  获取输入值，然后使用转换公式进行转换即可   |  | | --- | | private void btnHSI\_Click(object sender, EventArgs e)  {  if (pictureBox1.Image == null)  {  MessageBox.Show("错误，没有导入图片！");  return;  }  Color color = new Color();  Bitmap bt1 = new Bitmap(pictureBox1.Image);  int I = int.Parse(txtI.Text);  int S = int.Parse(txtS.Text);  int H = int.Parse(txtH.Text);  double Hudu = Math.PI / 180;  int r, g, b, R, G, B;  for (int i = 0; i < bt1.Width; i++)  {  for (int j = 0; j < bt1.Height; j++)  {  color = bt1.GetPixel(i, j);  R = r = color.R;  G = g = color.G;  B = b = color.B;  if (H >= 0 && H <= 120)  {  B = b + I \* (I - S);  R = (int)(r + I \* (1 + (S \* Math.Cos(H \* Hudu) / (Math.Cos((60 - H) \* Hudu)))));  G = g + 3 \* I - (B + R);    }  if (H > 120 && H <= 240)  {  R = r + I \* (I - S);  G = (int)(g + I \* (1 + (S \* Math.Cos((H - 120) \* Hudu) / (Math.Cos((180 - H) \* Hudu)))));  B = b + 3 \* I - (G + R);  }  if (H > 240 && H <= 360)  {  G = g + I \* (I - S);  B = (int)(b + I \* (1 + (S \* Math.Cos((H - 120) \* Hudu) / (Math.Cos((300 - H) \* Hudu)))));  R = b + 3 \* I - (G + B);  }  if (R > 255) R = 255;  if (R < 0) R = 0;  if (G > 255) G = 255;  if (G < 0) G = 0;  if (B > 255) B = 255;  if (B < 0) B = 0;  bt1.SetPixel(i, j, Color.FromArgb(R, G, B));  }  }  pictureBox2.Refresh();  pictureBox2.Image = bt1;  } |     **五、实验结果及分析(包括心得体会，本部分为重点，不能抄袭复制)**  通过本次实验，了解了灰度化、二值化、亮度调整，HSI的原理，并且完成了代码的撰写。本质上的算法大致都是通过两个for循环进行遍历，其时间复杂度为O(n^2)，修改他们的像素值。但在实验过程中，尝试使用一些较大的图片(.png)，就会发现处理的过程要等较长的时间。可想而知市面上成熟的图像处理软件不会是简单使用for循环，可能会有更加优质的算法进行处理，提高其运行速度。对于HSI的转换，要注意弧度的转换，以及三个值的取值范围。 | | | | | |