****

**天津理工大学**

**计算机科学与工程学院**

**实验报告**

**2017至 2018 学年 第 二 学期**

**实验三 图像灰度变换处理**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **课程名称** | **数字图像处理** | | | | |
| **学号** | **20152180** | **学生姓名** | **王帆** | **年级** | **2015** |
| **专业** | **计算机**  **科学与技术** | **教学班号** | **2** | **实验地点** | **主7-212** |
| **实验时间** | **2018年4月 9日 第 7节 至 第8 节** | | | | |
| **主讲教师** | **杨淑莹** | | | | |

**实验成绩**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **软件运行** | **效果** | **算法分析** | **流程设计** | **报告成绩** | **总成绩** |
|  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **实验（三）** | **实验名称** | **图像灰度变换处理** |
| **软件环境** | Windows  Visual Studio 2017 | |
| **硬件环境** | PC | |
| **实验目的** | | |
| 掌握图像的灰度变换原理，编程实现图像的灰度变换功能。 | | |
| **实验内容（应包括实验题目、实验要求、实验任务等）** | | |
| 1. **设计并实现一种灰度线性变换。**   要求：了解灰度线性变换基本原理，实现灰度线性变换。  说明：灰度线性变换基本原理  任务：  （1）在左视图中打开一幅位图。  （2）制作一个【灰度线性变换】菜单，将消息映射到右视图中，在右视图  中实现灰度线性变换。   1. **设计并实现一种灰度非线性变换。**   要求：了解灰度非线性变换基本原理，实现灰度非线性变换。  说明：灰度非线性变换基本原理  任务：  （1）在左视图中打开一幅位图。  （2）制作一个【**灰度非线性变换**】菜单，将消息映射到右视图中，在右视图  中实现灰度对数变换。 | | |
| **实验过程与实验结果**  **1.设计并实现一种灰度线性变换**  **原理：**  是一个线性或分段线性的单值函数，输出灰度级与输入灰度级呈线性关系的点运算，则由它确定的灰度变换称为灰度线性变换，简称灰度的线性变换。灰度的线性变换公式为：  式中，参数a为线性函数的斜率；b为线性函数在y轴的截距；f(x,y)表示输入图的灰度；g(x,y)表示输出图像的灰度。   * a>1，增加图像的对比度 * a<1，减小图像的对比度 * a=1且b≠0，图像整体的灰度值上移或者下移，也就是图像整体变亮或者变暗，不会改变图像的对比度。 * a<0且b=0，图像的亮区域变暗，暗区域变亮 * a=1且b=0，恒定变换，不变 * a=−1且b=255，图像灰度反转。 | | |
| **实现步骤：**  1.获取原图像的Bitmap对象objBitmap的其大小参量，并以此构造新Bitmap对象bitmap；  2.迭代实现对objBitmap每一个像素点（Pixel）到bitmap的反色变换操作：  对每个像素点的R,G,B色彩分量分别减去255并取相反数，获得反色后的新R,G,B色彩分量，并构造当前像素点；  3.使用bitmap构造全局变量curBitmap，销毁bitmap对象，使用curBitmap初始化右侧显示框。  **代码：**  //选项：基本处理-灰度变换-反色变换  private void ToolStripMenuItem\_gray\_line\_Click(object sender, EventArgs e)  {  try  {  height = objBitmap.Height;  width = objBitmap.Width;  Bitmap bitmap = new Bitmap(objBitmap);  Color color;  int r, g, b;  for (int i = 0; i < width; i++)  {  for (int j = 0; j < height; j++)  {  color = objBitmap.GetPixel(i, j);  r = 255 - color.R;  g = 255 - color.G;  b = 255 - color.B;  bitmap.SetPixel(i, j, Color.FromArgb(r, g, b));  }  }  curBitmap = new Bitmap(bitmap);  bitmap.Dispose();  this.pictureBox\_new.Image = curBitmap;  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "错误提示", MessageBoxButtons.OK, MessageBoxIcon.Stop);  throw;  }  }  **效果图：**    **图1线性变换（反色变换）**  **2.设计并实现一种灰度非线性变换**  **原理：**  当灰度图像f的各像素点f(x,y)的值域在区间[a,b]（a<b）上时，可将它按自然对数变换到区间[c,d]（c<d）上，从而求得输出图像g。因此算法的功能是把输入图像区间[a,b]对数变换到输出图像的灰度区间[c,d]，灰度值为0时，用一个很小的数eps置换后再计算对数值。 输入数组i[Row][Col]=输入图像  输出数组o[Row][Col]=输出图像  输入参数=变换前区间两端点  输入参数=变换后区间两端点  对数变换的一般表达式为：  其中，c为尺度比例常数，s为原图灰度值，t 为变换后的目标灰度值。  **实现步骤：**  1.获取原图像的Bitmap对象objBitmap的其大小参量，并以此构造新Bitmap对象bitmap；  2.迭代实现对灰度映射表bMap的建立；  3.迭代实现对objBitmap每一个像素点（Pixel）到bitmap的灰度映射操作：  对每个像素点的RGB色彩设置为灰度映射表内对应值，如bMap[color.R]；  4.使用bitmap构造全局变量curBitmap，销毁bitmap对象，使用curBitmap初始化右侧显示框。  **代码：**  //选项：基本处理-灰度变换-对数变换  private void ToolStripMenuItem\_gray\_log\_Click(object sender, EventArgs e)  {  try  {  height = objBitmap.Height;  width = objBitmap.Width;  Color color;  int[] bMap = new int[256];  Bitmap bitmap = new Bitmap(width, height);  for (int i = 0; i < 256; i++)  {  bMap[i] = (int)(Math.Log((double)i + 1.0) / (double)(25 \* 0.001) + 0);  if (bMap[i] < 0)  {  bMap[i] = 0;  }  else if (bMap[i] > 255)  {  bMap[i] = 255;  }  }  for (int x = 0; x < width; x++)  {  for (int y = 0; y < height; y++)  {  color = objBitmap.GetPixel(x, y);  bitmap.SetPixel(x, y, Color.FromArgb(bMap[color.R], bMap[color.G], bMap[color.B]));  }  }  curBitmap = new Bitmap(bitmap);  bitmap.Dispose();  this.pictureBox\_new.Image = curBitmap;  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "错误提示", MessageBoxButtons.OK, MessageBoxIcon.Stop);  throw;  }  }  **效果图：**    **图2 非线性变换（对数变换）** | | |

**附录**

**参考资料：**

**1. C# 图像处理（三）—— 反色处理 - linFen - 博客园**

[**https://www.cnblogs.com/luluping/archive/2012/07/11/2585552.html**](https://www.cnblogs.com/luluping/archive/2012/07/11/2585552.html)

**2. 图像处理基础(7)：图像的灰度变换 - Brook\_icv - 博客园**

[**https://www.cnblogs.com/wangguchangqing/p/6983680.html**](https://www.cnblogs.com/wangguchangqing/p/6983680.html)

**3. 6种图片灰度转换算法 · Issue #4 · aooy/blog · GitHub**

[**https://github.com/aooy/blog/issues/4**](https://github.com/aooy/blog/issues/4)