



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）		
<div>1. 设计并实现一种灰度线性变换。</div> <div>要求：了解灰度线性变换基本原理，实现灰度线性变换。</div> <div>说明：灰度线性变换基本原理</div> <div>任务：<div><div>（1）在左视图中打开一幅位图。</div><div>（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。</div></div></div> <div>2. 设计并实现一种灰度非线性变换。</div> <div>要求：了解灰度非线性变换基本原理，实现灰度非线性变换。</div> <div>说明：灰度非线性变换基本原理</div> <div>任务：<div><div>（1）在左视图中打开一幅位图。</div><div>（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。</div></div></div>		
实验过程与实验结果		
<div>1.设计并实现一种灰度线性变换</div> <div>原理：<div><div>$g(x,y) = T(f(x,y))$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：$g(x,y) = af(x,y) + b$</div><div>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。</div><div><div><div>•$a>1$，增加图像的对比度</div><div>•$a<1$，减小图像的对比度</div><div>•$a=1$ 且 $b\neq 0$，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。</div><div>•$a<0$ 且 $b=0$，图像的亮区域变暗，暗区域变亮</div><div>•$a=1$ 且 $b=0$，恒定变换，不变</div><div>•$a=-1$ 且 $b=255$，图像灰度反转。</div></div></div></div></div>		

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:

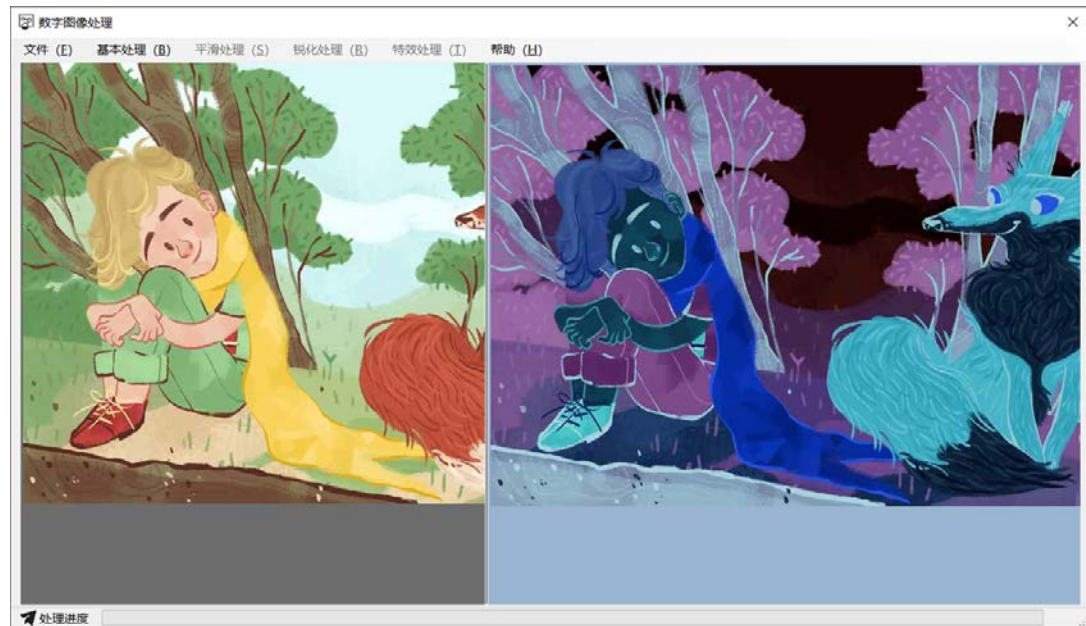


图 1 线性变换（反色变换）

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 ϵ 置换后再计算对数值。

输入数组 $i[\text{Row}][\text{Col}]$ =输入图像

输出数组 $o[\text{Row}][\text{Col}]$ =输出图像

输入参数=变换前区间两 endpoints

输入参数=变换后区间两 endpoints

对数变换的一般表达式为: $t = c * \log(1 + s)$

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 $bMap[\text{color.R}]$;
- 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：

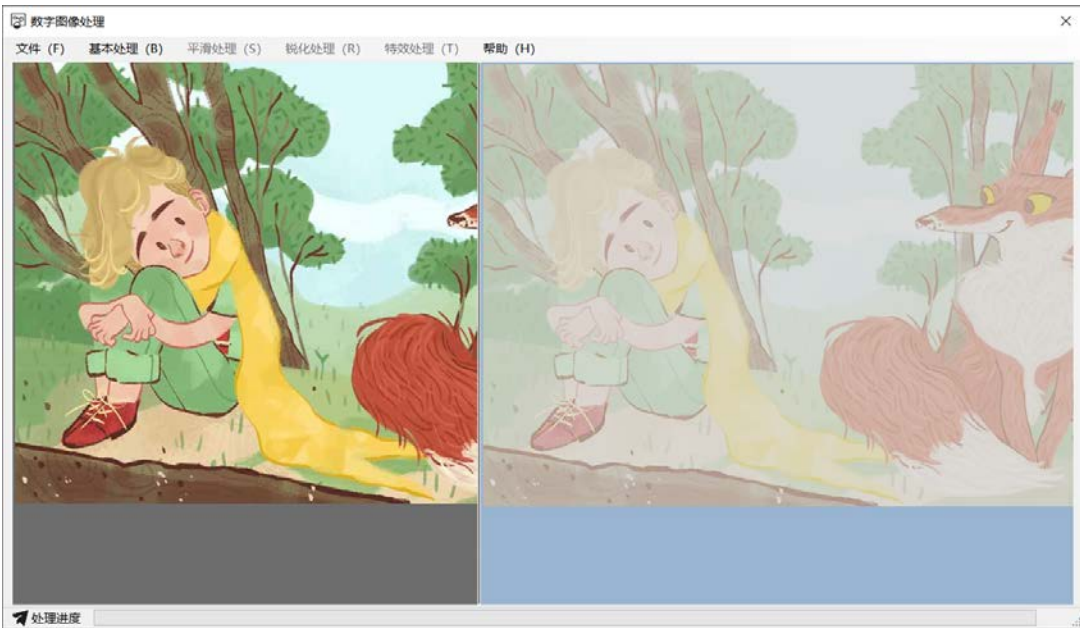


图 2 非线性变换（对数变换）

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>

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>

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

<https://github.com/aooy/blog/issues/4>