

# 浙江大学实验报告

专业：\_自动化（电气）\_

姓名：\_潘盛琪\_

学号：\_3170105737\_

日期：\_

地点：\_生工食品学院机房\_

课程名称：\_计算机图像处理与机器视觉\_ 指导老师：\_饶秀勤\_ 成绩：\_

实验名称：\_图像的灰度变换\_ 实验类型：\_设计型\_

## 一、实验目的和要求

由于受到输入方法中的设备、参数、环境等多种因素的影响，存在：

图像灰度偏暗或偏亮

整体灰度范围不足

某些图像有用区域的灰度层次差，而不必要处的灰度却显得过于丰富

为了改善图像的灰度对比度或满足图像上灰度的某些特殊要求，往往采用点运算方式进行灰度变换处理。

## 二、计算机配置与软件处理平台

基于 matlab

## 三、算法描述

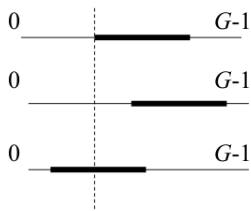
**1、灰度范围移动处理：**指将输入图像的像素的灰度加上或减去某一常数，得到输入图像的灰度的方法。

$$g(i,j)=f(i,j)+d$$

其中：

当 $d>0$ 时，灰度范围向高端移动，图像变亮；

当 $d<0$ 时，灰度范围向低端移动，图像变暗；



程序实现

## 2、灰度线性变换：

(1)、变换类型：整体灰度线性变换、局部灰度线性变换

■ 整体灰度线性变换：

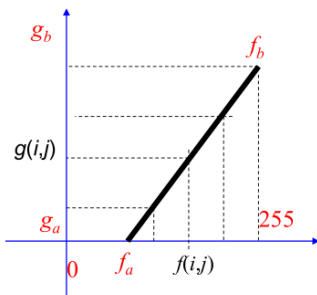
**应用：**当输入图像的灰度范围 $[f_a, f_b]$ 比允许的灰度范围 $[0, 255]$ 少得多时，以至于图像的对比度差，看不清楚，则往往将灰

度范围按上式拉伸到 $[g_a, g_b]$  ( $g_a=0, g_b=255$ ),

使图像中的灰度层次分明。

变换式：

$$g(i,j)=\frac{g_b-g_a}{f_b-f_a}[f(i,j)-f_a]+g_a$$



程序实现

## 2、灰度线性变换：(续)

### ■ 局部灰度线性变换：

限幅灰度拉伸、锯齿形灰度拉伸、阈值灰度拉伸

#### a、限幅灰度拉伸：

**应用：**图像中局部感兴趣的细节其灰度对比度差而难以分辨

变换式：

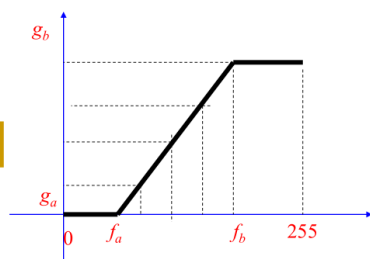
$$g(i,j) = \begin{cases} g_a & \text{当 } f(i,j) < f_a \\ \frac{g_b - g_a}{f_b - f_a} [f(i,j) - f_a] + g_a & \text{当 } f_a \leq f(i,j) \leq f_b \\ g_b & \text{当 } f(i,j) > f_b \end{cases}$$

## 2、灰度线性变换：(续)

#### a、限幅灰度拉伸：(续)

**结果：**可将局部灰度拉伸到最大限度，而无用信息被抑制为黑色或白色的单一灰度。

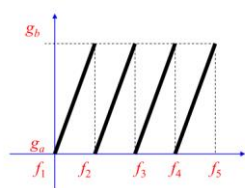
程序实现



## 2、灰度线性变换：(续)

#### b、锯齿形灰度拉伸：

**应用：**将输入图像中不同灰度区间  $[f_1, f_2]$ 、 $[f_2, f_3]$ 、 $[f_3, f_4]$  进行同样的灰度拉伸，使各个灰度区间都扩展到允许的整个灰度范围  $[g_a, g_b]$ 。



**结果：**使输入图像中原来是缓慢变化的灰度，经变换后在这些区间的  $f_1$ 、 $f_2$ 、 $f_3$  等分割点的两端灰度发生了突变。

如：当输入  $f_2$ ，输出为  $g_a$ ，但当输入略大于  $f_2$  时，则输出为  $g_b$ 。

请在实验课上实现

## 2、灰度线性变换：(续)

### ■ 阈值灰度分割法：

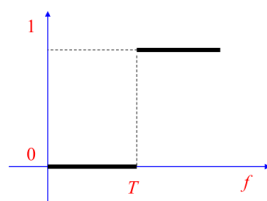
**方法：**以某灰度阈值为界，将图像信息分割成两部分。

变换式：

$$g(i,j) = \begin{cases} 0 & f(i,j) \leq T \\ 1 & f(i,j) > T \end{cases}$$

(二值图像)

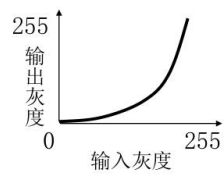
程序实现



### 3、灰度非线性变换：

$$G(x,y)=f(x,y)^2 \quad 0 \leq f(x,y) < 255$$

```
For j = 0 To h - 1
  For i = 0 To w - 1
    g(i,j)=f(i,j)*f(i,j) /255;
  Next i
Next j
```

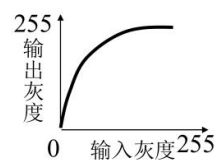


程序实现

### 3、灰度非线性变换：

$$G(x,y)=\text{sqr}(f(x,y)) \quad 0 \leq f(x,y) < 255$$

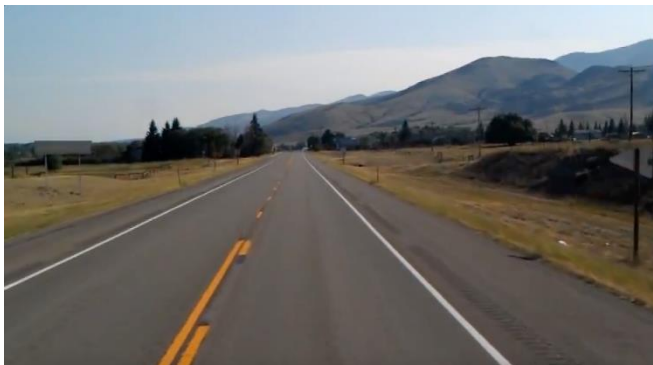
```
For j = 0 To h - 1
  For i = 0 To w - 1
    g(i,j) = (sqr(f(i,j))/16)* 255
  Next i
Next j
```



请在实验课上实现

## 四、结果与讨论

### 4.1 输入图像



### 4.2 输出图像

#### 1. 灰度范围移动处理

$d = -100$



原图像



$d = 100$



## 2. 整体灰度线性变换

原图像



灰度线性变换后的图像



## 3. 限幅灰度拉伸

原图像



灰度限幅拉伸后的图像



#### 4. 锯齿形灰度拉伸

原图像

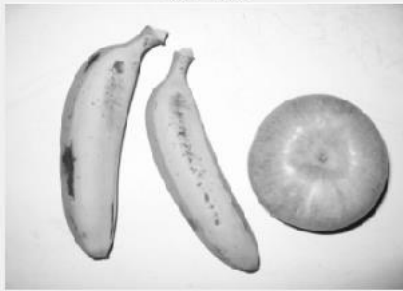


锯齿灰度拉伸后的图像



#### 5. 阈值灰度分割法

原图像



阈值灰度分割后的图像



Th= 205

## 6. 平方变换

原图像



平方变换后的图像



## 7. 开方变换

原图像



开方变换后的图像



### 4.3 讨论

可以看到灰度范围移动处理可以使图像整体变亮或变暗。整体灰度变换和平方变换对提升图像对比度有很好的效果。而灰度限幅拉伸后，图像中的人变得更突出了，可以突出图片中的有效信息。灰度阈值分割可以很容易实现图片二值化。

## 五、结论

本次实验编程实现了 7 种不同的灰度变换方法，不同的方法的运用场景各不相同。平方变换，整体灰度线性变换均可以提高对比度。而开方变换可以降低对比度。灰度阈值分割可以很容易地实现图像二值化，但对于大部分图片效果不是很好。

## 六、源程序

### 1. 灰度范围移动处理

```
img = imread('test_image.jpg');
```

```
subplot(3, 1, 1);  
imshow(img - 100);  
title('d = -100');
```

```
subplot(3, 1, 2);  
imshow(img);  
title('原图像');
```

```
subplot(3, 1, 3);  
imshow(img + 100);  
title('d = 100');
```

## 2. 整体灰度线性变换

```
img = imread('haze.jpg');
fa = min(min(img));
fb = max(max(img));
gb = 255;
ga = 0;

for i = 1:3
    img1(:, :, i) = ((gb - ga)/(fb(:, :, i) - fa(:, :, i)))*(img(:, :, i)-fa(:, :, i)) + ga;
end
subplot(3, 1, 1);
imshow(img1);
title('d = -100');

subplot(3, 1, 2);
imshow(img);
title('原图像');

for i = 1:3
    img2(:, :, i) = ((gb - ga)/(fb(:, :, i) - fa(:, :, i)))*(img(:, :, i)-fa(:, :, i)) + ga;
end
subplot(3, 1, 3);
imshow(img2);
title('d = 100');
```

## 3. 限幅灰度拉伸

```
img = imread('haze.jpg');
img = rgb2gray(img);
fa = floor(min(min(img)) + 10);
fb = floor(max(max(img)) - 10);
gb = 256;
ga = 1;
[row, col] = size(img);%获取输入图像的尺寸
% imgout = zeros(row, col);%初始化变换后的图像

%显示原图像
subplot(2, 1, 1);
imshow(img);
title('原图像');

imgout = img;
imgout(imgout > fb) = 1;
imgout(imgout < fa) = 256;
```



```

for i = 1 : row
    for j = 1 : col
        if imgout(i, j) > fa && imgout(i, j) < fb
            imgout(i, j) = ((gb - ga)/(fb - fa))*(img(i, j) - fa) + ga;
        end
    end
end

```

%显示变换后的图像

```

subplot(2, 1, 2);
imshow(uint8(imgout));
title('灰度限幅拉伸后的图像');

```

#### 4. 锯齿形灰度拉伸

```

img = imread('haze.jpg');
img = rgb2gray(img);
fa0 = floor(min(min(img)));
fb0 = floor(max(max(img)));
gb = 256;
ga = 1;
step = floor((fb0 - fa0) / 4);
[row, col] = size(img);%获取输入图像的尺寸
% imgout = zeros(row, col);%初始化变换后的图像

```

%显示原图像

```

subplot(2, 1, 1);
imshow(img);
title('原图像');

```

```

imgout = img;
for k = 1 : 4
    fa = fa0 + (k - 1) * step;
    fb = fa + step;
    if i == 4
        fb = fb0;
    end
    for i = 1 : row
        for j = 1 : col
            if imgout(i, j) > fa && imgout(i, j) < fb
                imgout(i, j) = ((gb - ga)/(fb - fa))*(img(i, j) - fa) + ga;
            end
        end
    end
end

```

```
end
%显示变换后的图像
subplot(2, 1, 2);
imshow(uint8(imgout));
title('锯齿灰度拉伸后的图像');
```

## 5. 阈值灰度分割法

```
img = imread('fruit.png');
img = rgb2gray(img);
```

```
%显示原图像
subplot(2, 1, 1);
imshow(img);
title('原图像');
```

```
Th = 205;%设定阈值
imgout = img;
imgout(imgout < Th) = 1;
imgout(imgout > Th) = 256;
```

```
%显示变换后的图像
subplot(2, 1, 2);
imshow(imgout);
title('阈值灰度分割后的图像');
```

## 6. 平方变换

```
img = imread('haze.jpg');
img = double(img);
imgout = img.^ 2 / 256;
```

```
%显示原图像
subplot(2, 1, 1);
imshow(uint8(img));
title('原图像');
```

```
%显示变换后的图像
subplot(2, 1, 2);
imshow(uint8(imgout));
title('平方变换后的图像');
```

## 7. 开方变换

```
img = imread('haze.jpg');  
img = double(img);  
imgout = sqrt(img) / 16 .* 255;
```

%显示原图像

```
subplot(2, 1, 1);  
imshow(uint8(img));  
title('原图像');
```

%显示变换后的图像

```
subplot(2, 1, 2);  
imshow(uint8(imgout));  
title('开方变换后的图像');
```

.....