**实验报告**

装 订 线

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

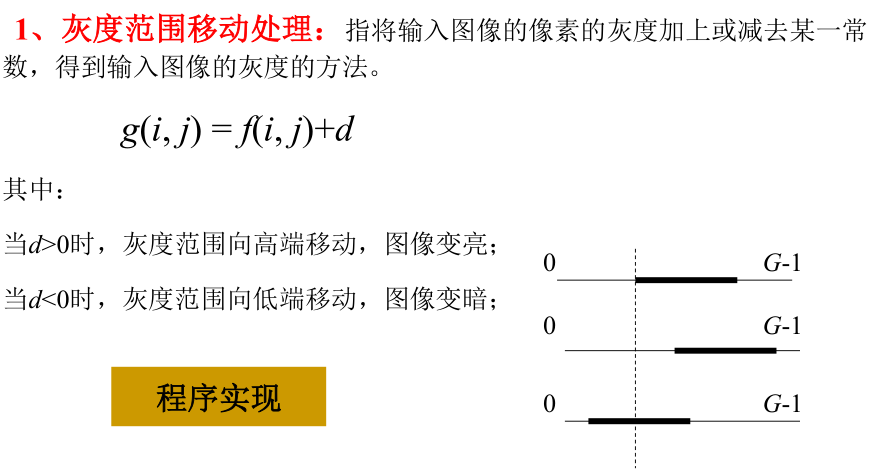
为了改善图像的灰度对比度或满足图像上灰度的某些特殊要求，往往采用点

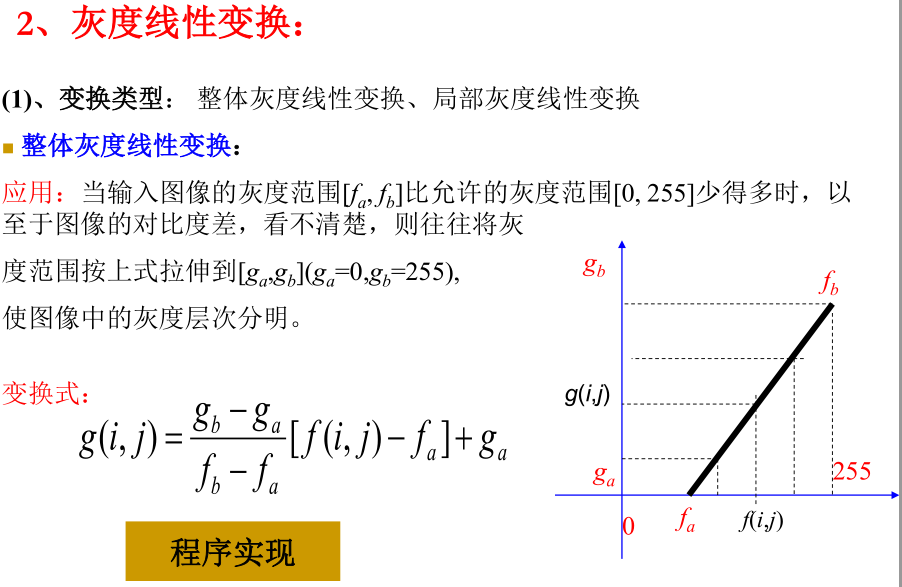
运算方式进行灰度变换处理。

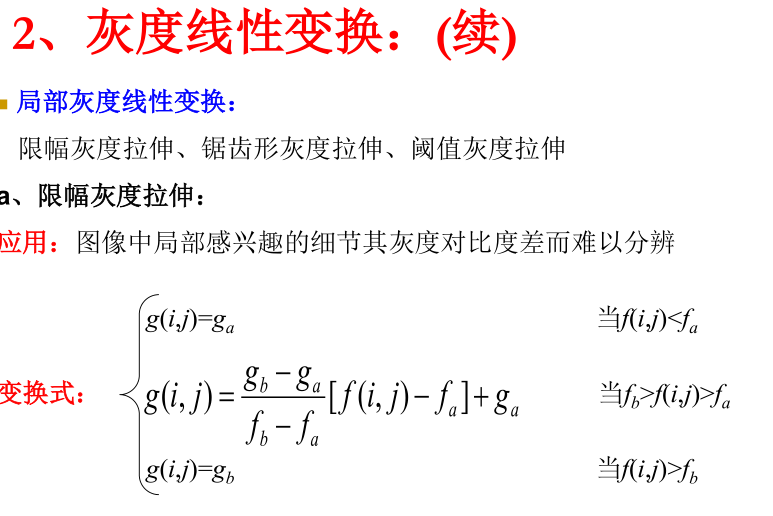
**二、计算机配置与软件处理平台**

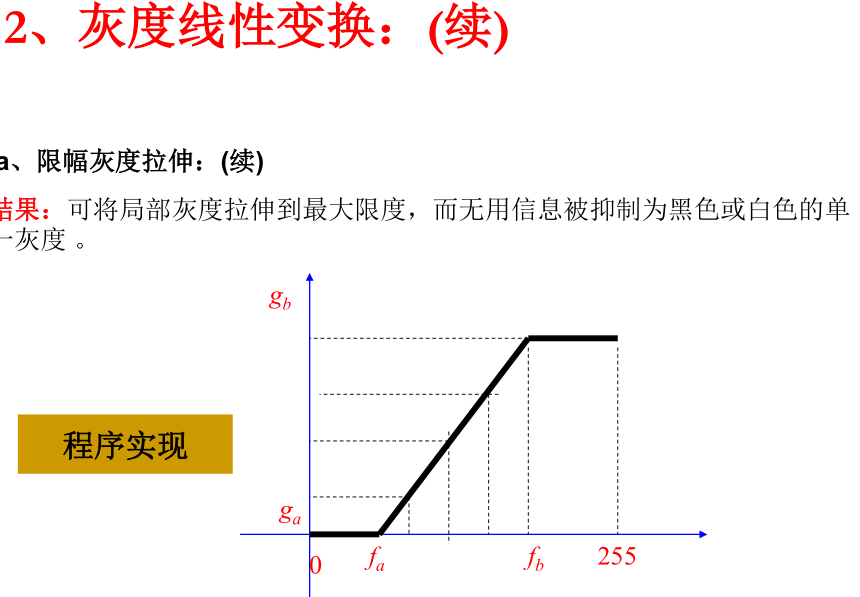
基于matlab

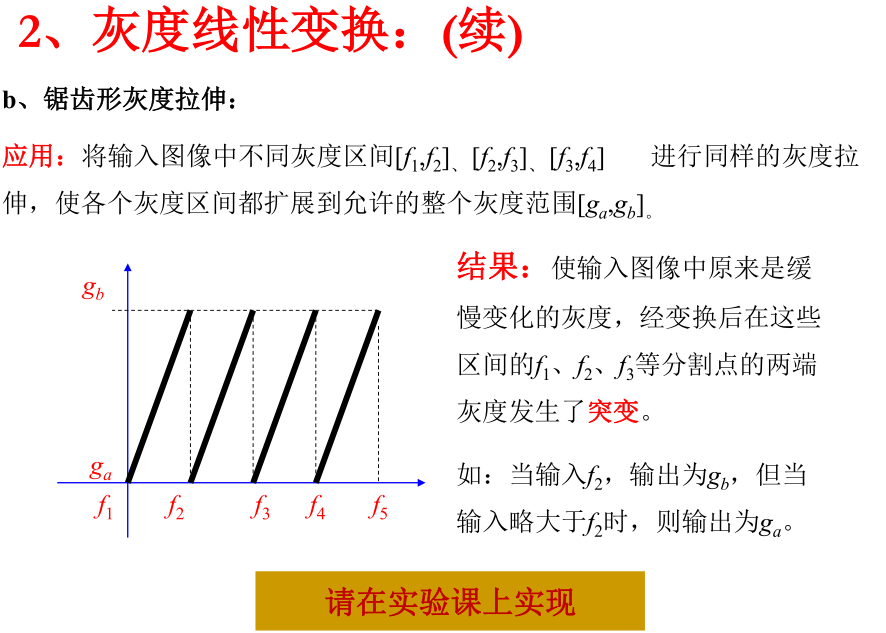
**三、算法描述**

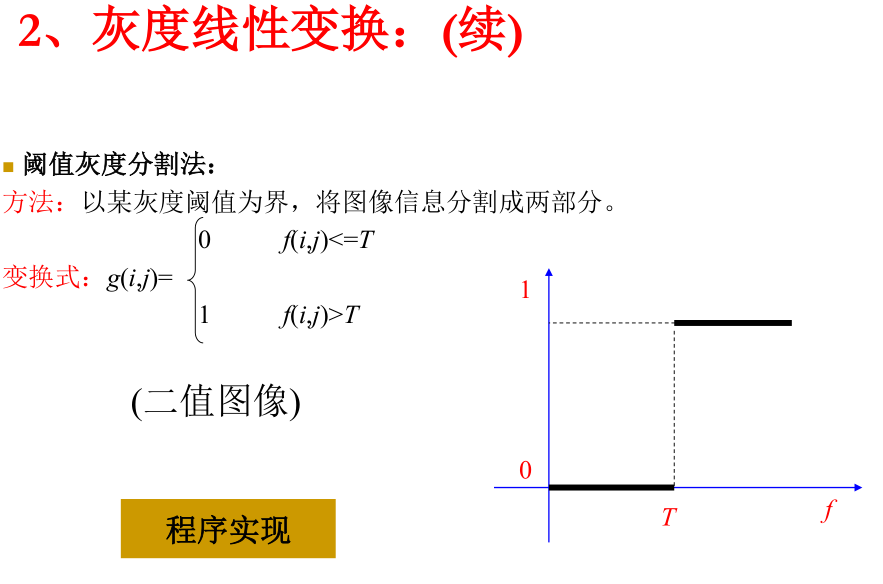


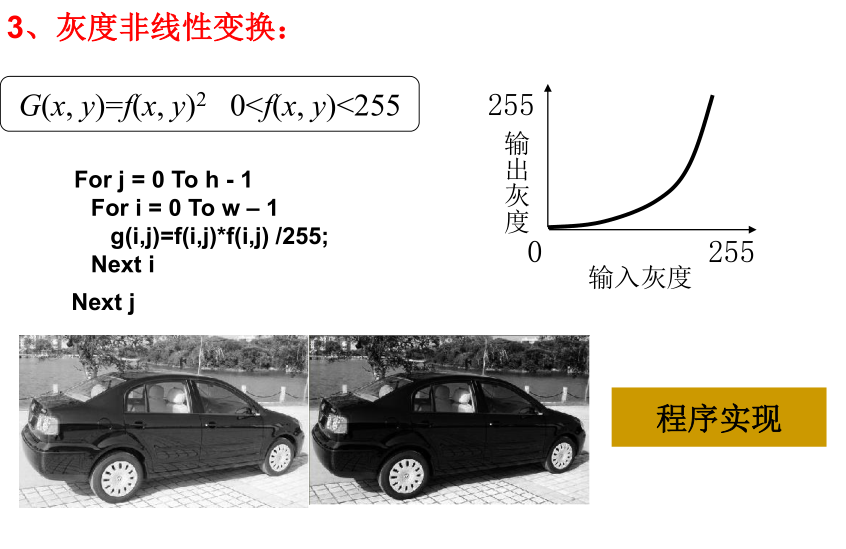


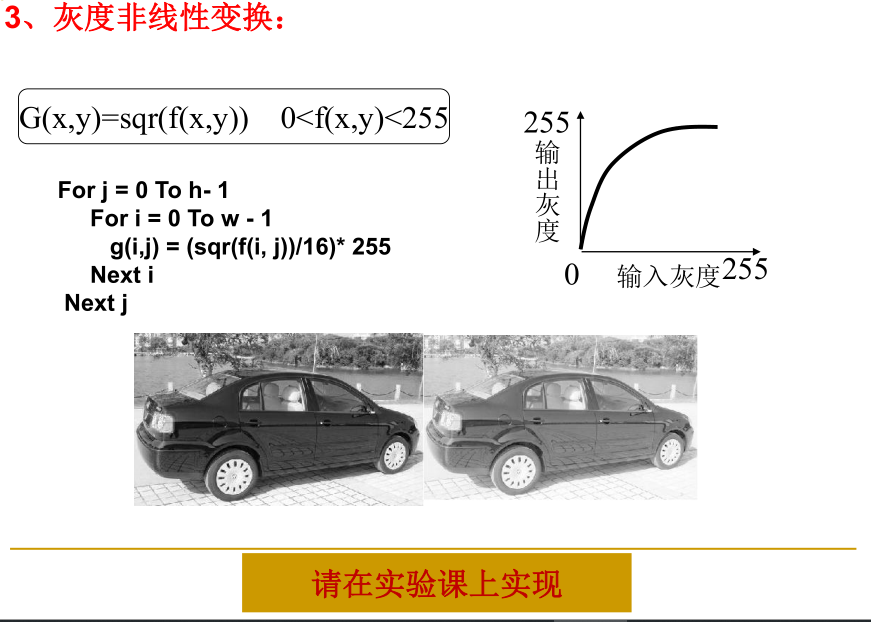












**四、结果与讨论**

**4.1 输入图像**



**4.2 输出图像**

1. 灰度范围移动处理



1. 整体灰度线性变换



1. 限幅灰度拉伸



1. 锯齿形灰度拉伸



1. 阈值灰度分割法

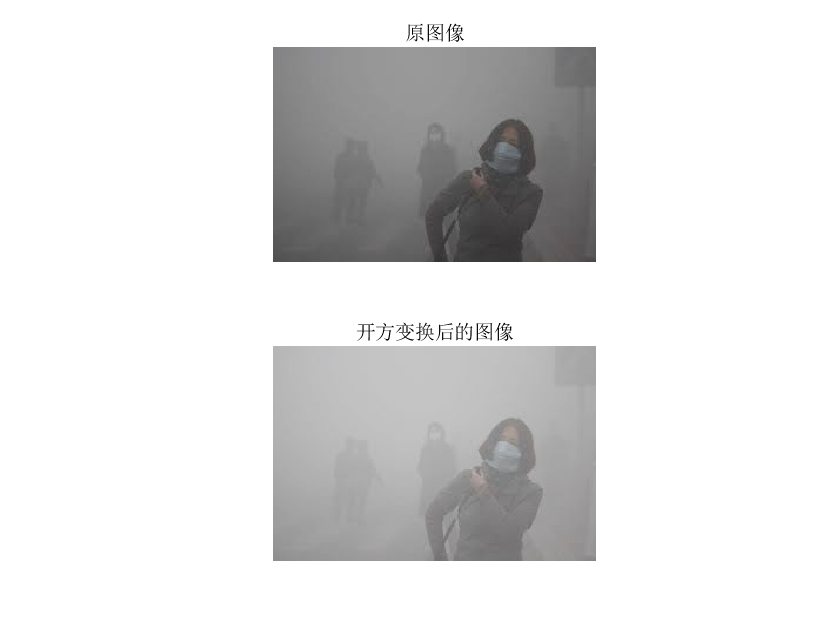


Th= 205

1. 平方变换



1. 开方变换



**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');

1. 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');

1. 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');

1. 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

%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');

1. 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');

1. 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');

1. 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');