# 光学图像处理实验报告

来源：网络 作者：星月相依 更新时间：2024-07-24

*第一篇：光学图像处理实验报告光学图像处理实验报告直方图均衡化的研究一、摘要直方图均衡化就是把一已知灰度概率分布的图像经过一种变换,使之演变成一幅具有均匀灰度概率分布的新图像。它是以累积分布函数变换法为基础的直方图修正法。分析和总结灰度直方...*

**第一篇：光学图像处理实验报告**

光学图像处理实验报告

直方图均衡化的研究

一、摘要

直方图均衡化就是把一已知灰度概率分布的图像经过一种变换,使之演变成一幅具有均匀灰度概率分布的新图像。它是以累积分布函数变换法为基础的直方图修正法。分析和总结灰度直方图的均衡化算法并通过VC++实验验证该方法能有效达到图像增强的目的。对于较为暗淡的图像，采用直方图均衡化能够增强其整体对比度，获的较为理想的观察效果。

二、关键字

灰度统计

直方图

均衡化

三、实验原理

1、直方图的理论基础：

（1）直方图概念：灰度直方图表示图像中每种灰度出现的频率。（2）直方图的作用： 反映一幅图像的灰度分布特性

n（3）直方图的计算：

p(rk)k0rk1k0,1,2,,l1 n式中：nk为图像中出现rk级灰度的像素数，n是图像像素总数，而nk/n即为频数。

2、设计目的： 产生一幅灰度级分布具有均匀概率密度的图像，扩展像素取值的动态范围，达到了图象增强的目的。

3、直方图均衡化的效果 ：

1）变换后直方图趋向平坦，灰级减少，灰度合并。

2）原始象含有象素数多的几个灰级间隔被拉大了，压缩的只是象素数少的几个灰度级，实际视觉能够接收的信息量大大地增强了，增加了图象的反差。同时，也增加了图象的可视粒度。

4、离散情况下的直方图均衡化的算法：

A、列出原始图像的灰度级 fj,j0,1,,L1

B、统计各灰度级的像素数目 nj,j0,1,,L1

C、计算原始图像直方图各灰度级的频数 Pf(fj)nj/n,j0,1,,L1

kD、计算累积分布函数 C(f)j0Pf(fj),j0,1,,k,L1

F、应用以下公式计算映射后的输出图像的灰度级，P为输出图像灰度级的个数，其中INT为取整符号：

giINT[(gmaxgmin)C(f)gmin0.5] G、用的映射关系修改原始图像的灰度级，从而获得直方图近似为均匀分布的输出图像。



四、实验内容及源程序

1、灰度分布密度的统计 程序代码如下：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*函数名称：ZhiFangTu(float \*tongji)\*函数类型：void \*变量说明：tongji，灰度分布密度统计 \*功能：对图像进行灰度直方图统计

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ void CAAAView::OnZhifangtu(float \*tongji){ // TODO: Add your command handler code here

int huidu[256];//灰度计数

CAAADoc\* pDoc = GetDocument();

LPSTR lpDIB;LPSTR

lpDIBBits;

lpDIB =(LPSTR)::GlobalLock((HGLOBAL)pDoc->GetHDIB());

lpDIBBits = ::FindDIBBits(lpDIB);//原图数据区指针

int iH,iW;memset(huidu,0,sizeof(huidu));//变量初始化

iH = ::DIBHeight(lpDIB);//宽

iW = ::DIBWidth(lpDIB);//长

LPBYTE temp1=new BYTE[iH\*iW];//新图像缓冲区

memcpy(temp1,lpDIBBits,iH\*iW);//复制原图像到缓冲区

for(int i=0;iGetHDIB());

lpDIBBits = ::FindDIBBits(lpDIB);//原图数据区指针

float Hdmd[256];//灰度密度

float temp[256];//中间变量

int a[256];long i,j;memset(temp,0,sizeof(temp));//初始化

OnZhifangtu(Hdmd);//获取图像的灰度密度分布

for(i=0;iUpdateAllViews(NULL);::GlobalUnlock((HGLOBAL)pDoc->GetHDIB());EndWaitCursor();}

五、实验结果的分析与比较

a原图如下所示：

b均衡后的结果如下：

结论：图像直方图趋于平坦化，且灰度间隔被拉大，从而有利于图像的分析和识别。对于较为暗淡的图像，采用直方图均衡化能够增强其整体对比度，获的较为理想的观察效果。

六、参考文献

VC++图像处理程序设计（第二版）（杨淑莹等 编著）

**第二篇：数字图像处理实验报告**

数字图像处理

实验报告

目录

1.数字图像处理简介

2.实验目的3.实验内容

4.实验结果及代码展示

5.算法综述

6.Matlab优势

7.总结

8.存在问题

一、数字图像处理简介

图像处理，是对图像进行分析、加工、和处理，使其满足视觉、心理以及其他要求的技术。图像处理是信号处理在图像域上的一个应用。目前大多数的图像是以数字形式存储，因而图像处理很多情况下指数字图像处理。此外，基于光学理论的处理方法依然占有重要的地位。

图像处理是信号处理的子类，另外与计算机科学、人工智能等领域也有密切的关系。

传统的一维信号处理的方法和概念很多仍然可以直接应用在图像处理上，比如降噪、量化等。然而，图像属于二维信号，和一维信号相比，它有自己特殊的一面，处理的方式和角度也有所不同。

二、实验目的

巩固所学知识，提高所学能力

三、实验内容

利用matlab的GUI程序设计一个简单的图像处理程序，并含有如下基本功能： 1.读入一幅RGB图像，变换为灰度图像和二值图像，并在同一个窗口内分成三个子窗口来分别显示RGB图像和灰度图像，注上文字标题 2.对给定图像进行旋转

3．对给定的图像添加噪声（椒盐噪声、高斯噪声）

四、实验结果及代码展示

1.软件设计界面

2.各模块功能展示以及程序代码

（1）读入一幅RGB图像，变换为灰度图像和二值图像，并在同一个窗口内分成三个子窗口来分别显示RGB图像和灰度图像，注上文字标题

效果展示：

代码：

a = imread(\'C:Documents and SettingsAdministrator桌面数字图像舞美.JPG\');

i = rgb2gray(a);I = im2bw(a,0.5);

subplot(3,1,1);imshow(a);title(\'源图像\')subplot(3,1,2);imshow(i);title(\'灰度图像\')subplot(3,1,3);imshow(I);title(\'二值图像\')

（2）图像旋转 原图

效果展示：

代码：

clc;clear all;close all;

Img=imread(\'D:My DocumentsMy Pictures5.JPG\');Img=double(Img);[h w]=size(Img);alpha=pi/4;

wnew=w\*cos(alpha)+h\*sin(alpha);hnew=w\*sin(alpha)+h\*cos(alpha);wnew=ceil(wnew);

hnew=ceil(hnew);u0=w\*sin(alpha);

T=[cos(alpha),sin(alpha);-sin(alpha),cos(alpha)];Imgnew2=zeros(hnew,wnew);Imgnew1=zeros(hnew,wnew);for u=1:hnew

for v=1:wnew

tem=T\*([u;v]-[u0;0]);x=tem(1);y=tem(2);if x>=1&&x=1&&y>I=imread(\'第一个.tif\');2．利用whos命令提取该读入图像”第一个.tif”的基本信息； >>whos I 3．利用imshow()函数来显示这幅图像； >>imshow(I);

第一个.tif 4．利用imfinfo函数来获取图像文件的压缩，颜色等等其他的详细信息；

>>imfinfo(\'第一个.tif\');5．利用imwrite()函数来压缩这幅图象，将其保存为一幅压缩了像素的jpg文件,设为flower.jpg；语法：imwrite(原图像，新图像，‘quality’,q), q取0-100。

>>imwrite(I,\'第一个.jpg\',\'quality\',50)6．同样利用imwrite()函数将最初读入的tif图象另存为一幅bmp图像，设为flower.bmp。>>imwrite(I,\'第一个.bmp\');7．用imread()读入图像：Lenna256.jpg 和camemaman.jpg； >>b=imread(\'lena256.bmp\');>>c=imread(\'cameraman.tif\');8．用imfinfo()获取图像Lenna256.jpg和camemaman.jpg 的大小； >>imfinfo(\'lena256.bmp\');>>imfinfo(\'cameraman.tif\');9．用figure,imshow()分别将Lenna256.jpg和camemaman.jpg显示出来，观察两幅图像的质量。>>figure >>imshow(b);>>figure >>imshow(c);

（Lena256.jpg图像截图）

（cameraman.jpg图像截图）

10．用im2bw将一幅灰度图像转化为二值图像，并且用imshow显示出来观察图像的特征。>> d=im2bw(b);>>figure >>imshow(b);>>figure >>imshow(d);

（二值化截图）

三、实验总结

通过实验MatLab软件的基本使用有了基本的了解，学会了使用MatLab软件来读取一个特定格式的图像，并通过相关的命令语句对图像进行格式转换、图像压缩、二值化等的处理，掌握了利用MATLAB来获取图像的大小、颜色、高度、宽度等等相关信息，掌握在MATLAB中如何通过imshow()语句来读取图像等等。

第二 图像基本运算

一、实验目的

1．了解图像的算术运算在数字图像处理中的初步应用。2．体会图像算术运算处理的过程和处理前后图像的变化。

二、实验原理

图像的代数运算是图像的标准算术操作的实现方法，是两幅输入图像之间进行的点对点的加、减、乘、除运算后得到输出图像的过程。如果输入图像为A(x,y)和B(x,y)，输出图像为C(x,y)，则图像的代数运算有如下四种形式：

C(x,y)= A(x,y)+ B(x,y)C(x,y)= A(x,y)-B(x,y)C(x,y)= A(x,y)\* B(x,y)C(x,y)= A(x,y)/ B(x,y)

三、实验步骤 1．图像的加法运算

在MATLAB中，如果要进行两幅图像的加法，或者给一幅图像加上一个常数，可以调用imadd函数来实现。imadd函数将某一幅输入图像的每一个像素值与另一幅图像相应的像素值相加，返回相应的像素值之和作为输出图像。imadd函数的调用格式如下：

Z = imadd（X，Y）首先读入两幅图像

>>a=imread(\'第二个原图1.jpg\');>>b=imread(\'第二个原图2.jpg\')通过一个加法操作：>> c=imadd(a,b);

给图像的每一个像素加上一个常数可以使图像的亮度增加。如截图

第一张为原图，第二张为亮度加50，第三张为亮度减50 2．图像的减法运算

在MATLAB中，使用imsubtract函数可以将一幅图像从另一幅图像中减去，或者从一幅图像中减去一个常数。imsubtract函数将一幅输入图像的像素值从另一幅输入图像相应的像素值中减去，再将这个结果作为输出图像相应的像素值。imsubtract函数的调用格式如下：

Z = imsubtract(X,Y)； 读入一幅画后通过减法 >>a3=imsubtract(a,50);

3.图像的乘法运算

在MATLAB中，使用immultiply函数实现两幅图像的乘法。immultiply函数将两幅图像相应的像素值进行元素对元素的乘法操作（MATLAB点乘），并将乘法的运算结果作为输出图形相应的像素值。immulitply函数的调用格式如下：

Z = immulitply(X,Y)读入一幅图后通过乘法操作 >> a=imread(\'cameraman.tif\');>> b=immultiply(a,1.5);

4．图像的除法运算

在MATLAB中使用impide函数进行两幅图像的除法。impide函数对两幅输入图像的所有相应像素执行元素对元素的除法操作（点除），并将得到的结果作为输出图像的相应像素值。impide函数的调用格式如下：

Z = impide(X,Y)读入一幅图后通过除法操作

四、实验总结

通过对图像的四则运算了结图像的不同变化过程，对软件的进一步使用也有了更加深刻的认识。

实验三 图像增强—空域滤波

一、实验目的

进一步了解MatLab软件/语言，学会使用MatLab对图像作滤波处理，使学生有机会掌握滤波算法，体会滤波效果。

了解几种不同滤波方式的使用和使用的场合，培养处理实际图像的能力，并为课堂教学提供配套的实践机会。

二、实验设备与软件

(1)IBM-PC计算机系统

(2)MatLab软件/语言包括图像处理工具箱(Image Processing Toolbox)

(3)实验所需要的图片

三、实验内容与步骤

a)调入并显示原始图像“原图像.jpg”。>>I=imread(\'原图像.jpg\');b)利用imnoise命令在图像“原图像.jpg”上加入高斯(gaussian)噪声

>>J = imnoise(I,\'gauss\',0.02);

%添加高斯噪声 c)利用预定义函数fspecial命令产生平均(average)滤波器 d）分别采用3x3和5x5的模板，分别用平均滤波器以及中值滤波器，对加入噪声的图像进行处理并观察不同噪声水平下，上述滤波器处理的结果；

>>ave1=fspecial(\'average\',3);

%产生3×3的均值模版 >>ave2=fspecial(\'average\',5);

%产生5×5的均值模版 >>K = filter2(ave1,J)/255;

%均值滤波3×3 >>L = filter2(ave2,J)/255;

%均值滤波5×5 e）选择不同大小的模板，对加入某一固定噪声水平噪声的图像进行处理，观察上述滤波器处理的结果。

>>M = medfilt2(J,[3 3]);

%中值滤波3×3模板 >>N = medfilt2(J,[4 4]);

%中值滤波4×4模板

f）利用imnoise命令在图像Sample2-1.jpg 上加入椒盐噪声(salt & pepper)>>J = imnoise(I,\'salt& pepper\',0.02);

%添加椒盐噪声

四、实验总结

椒盐噪声是由图像传感器，传输信道，解码处理等产生的黑白相间的亮暗点噪声。椒盐噪声往往由图像切割引起，去除脉冲干扰及椒盐噪声最常用的算法是中值滤波。椒盐噪声是指两种噪声，一种是盐噪声，另一种是胡椒噪声。盐=白色，椒=黑色。前者是高灰度噪声，后者属于低灰度噪声。一般两种噪声同时出现，呈现在图像上就是黑白杂点。这点我们通过实验结果可以明显看到。中值滤波对于滤除图像的椒盐噪声非常有效。

实验四图像分割

一、实验目的

使用MatLab 软件进行图像的分割。使学生通过实验体会一些主要的分割算子对图像处理的效果，以及各种因素对分割效果的影响。

二、实验要求

要求学生能够自行评价各主要算子在无噪声条件下和噪声条件下的分割性能。能够掌握分割条件(阈值等)的选择。完成规定图像的处理并要求正确评价处理结果，能够从理论上作出合理的解释。

三、实验内容与步骤

(1)使用Roberts 算子的图像分割实验，使用的原图是cameraman.jpg,截图如下

(2)使用Prewitt 算子的图像分割实验 截图如下

(3)使用Sobel 算子的图像分割实验

(4)使用LoG(拉普拉斯-高斯)算子的图像分割实验

四、实验结果

对Roberts算子、Prewitt 算子、Sobel 算子、LoG(拉普拉斯-高斯)算子的运算对图像的结果有了基本的认识，加深学习效果。

实验五 形态学运算

1、实验目的

学习常见的数学形态学运算基本方法，了解腐蚀、膨胀、开运算、闭运算取得的效果，培养处理实际图像的能力，并为课堂教学提供配套的实践机会。

2、实验要求

利用MatLab工具箱中关于数学形态学运算的函数，计算本指导书中指定二值图像进行处理。

3、实验设备与软件

1.LC-PC计算机系统

2.MatLab软件/语言包括图像处理工具箱(Image Processing Toolbox)3.实验所需要的图片

4、实验内容与步骤

1.调入并显示图像“原图.jpg”； 2.调入并显示图像“原图.jpg”；

3.选取合适的阈值，得到二值化图像“原图.jpg”； >>bw = im2bw(I,level);

%二值化 4.设置结构元素；

5.对得到的二值图像“原图.jpg”进行腐蚀运算； >>BW2 = imerode(bw,SE1);

%腐蚀 6.对得到的二值图像“原图.jpg”进行膨胀运算； >>BW1 = imdilate(bw,SE);

%膨胀 7.对得到的二值图像“原图.jpg”进行开运算；

>>BW3 = bwmorph(bw, \'open\');

%开运算 8.对得到的二值图像“原图.jpg”进行闭运算； >>BW4 = bwmorph(bw, \'close\');

%闭运算 9.将两种处理方法的结果作比较；

五、实验总结

通过本次实验，学习了常见的数学形态学运算基本方法，了解腐蚀、膨胀、开运算、闭运算取得的效果，培养处理实际图像的能力，通过自己动手的实验，对课本上的知识有了更加深刻的理解。

**第四篇：数字图像处理实验报告**

实验一 数字图像的获取

一、实验目的

1、了解图像的实际获取过程。

2、巩固图像空间分辨率和灰度级分辨率、邻域等重要概念。

3、熟练掌握图像读、写、显示、类型转换等 matlab 函数的用法。

二、实验内容

1、读取一幅彩色图像，将该彩色图像转化为灰度图像，再将灰度图像转化为索引图像并显示所有图像。

2、编程实现空间分辨率变化的效果。

三、实验原理

1、图像读、写、显示 I=imread(‘image.jpg’)Imview(I)Imshow(I)Imwrite(I,’wodeimage.jpg’)

2、图像类型转换

I=mat2gray(A,[amin,amax]);按指定的取值区间[amin,amax]将数据矩阵 A 转化为灰度

图像 I，amin 对应灰度 0，amax 对应 1，也可以不指定该区间。

[x,map]=gray2ind(I,n);按指定的灰度级 n 将灰度图像转化为索引图像,n 默认为 64 I=ind2gray(x,map);索引图像转化为灰度图像 I=grb2gray(RGB);真彩色图像转化为灰度图像

[x,map]=rgb2ind(RGB);真彩色图像转化为索引图像 RGB=ind2rgb(x,map);索引图像转化为真彩色图像

BW=im2bw(I,level);将灰度图像转化为二值图像,level 取值在[0,1]之间

BW=im2bw(x,map,level);将索引图像转化为二值图像,level 取值在[0,1]之间 BW=im2bw(RGB,level);将真彩色图像转化为二值图像,level 取值在[0,1]之间

四、实验代码及结果

1、in=imread(\'peppers.png\');i=rgb2gray(in);[x,map]=gray2ind(i,128);subplot(131),imshow(in)subplot(132),imshow(i)subplot(133),imshow(x),colormap(map)

、%空间分辨率变化的效果

clc,close all,clear i=imread(\'cameraman.tif\');

i=imresize(i,[256,256]);i1=i(1:2:end,1:2:end);[m1,n1]=size(i)i2=i1(1:2:end,1:2:end);[m2,n2]=size(i2)i3=i2(1:2:end,1:2:end);[m3,n3]=size(i3)subplot(221),imshow(i),xlabel(\'256 x 256\')subplot(222),imshow(i1),xlabel(\'128 x 128\')subplot(223),imshow(i2),xlabel(\'64 x 64\')subplot(224),imshow(i3),xlabel(\'32 x 32\')256 x 25664 x 64128 x 12832 x 32

实验二

图像的几何变换

一、实验目的

掌握图像的基本几何变换的方法

1、图像的平移

2、图像的旋转

二、实验内容

练习用matalb命令实现图像的平移、旋转操作

1、.编写实现图像平移的函数

2、用imread命令从你的硬盘读取一幅256×256灰度图;

3、调用平移函数，将256×256灰度图平移100行200列，在同一个窗口中显示平移前和平移后的图像。

4、再开辟一个窗口，分别用最近邻插值法、双线性插值法实现图像顺势针旋转50°，显示在同一窗口中，并比较两种效果图（在报告中）

三、实验原理

提示：图片平移就是实现运算

x\'10x0x \' y01y0y10011 

x\'xx0即：y\'yy0

四、实验代码及结果

1、function [I]=hmove(i,x0,y0);%编写实现图像平移的函数hmove，平移量为 [r,c]=size(i);

%x0,y0,平移前图像矩阵为i，I(r+x0,c+y0)=0;

%平移后图像矩阵为I for x=1:r;

for y=1:c;

x1=x+x0;

y1=y+y0;

I(x1,y1)=i(x,y);

end;end;参考程序 subplot(2,2,1)imshow(RGB)subplot(2,2,3)gray1=rgb2gray(RGB);imagesc(gray1),colormap(gray);

subplot(2,2,2)I1=hmove(gray1,100,20);subimage(gray1),axis(\'image\');subplot(2,2,4),imagesc(I1),colormap(gray),axis([1,700],[1,820]);

2、显示图像的傅立叶频谱  a=0:800;b=0:600;

 %[x,y]=meshgrid([-20:0.2:20],[-20:0.2:20]); [x,y]=meshgrid(a,b); i=imread(\'hr.jpg\'); I=rgb2gray(i);

 subplot(1,2,1),subimage([0,800],[0,600],i); subplot(1,2,2); s=fft2(I,601,801); mesh(x,y,log(abs(s)));％图像的傅立叶幅度频谱以三维图形显示  colormap(hsv);

实验三 图像空域变换增强（1）

一、实验目的

1、掌握直方图均衡化算法。

2、巩固灰度变换、直方图修正、图像算术和逻辑运算等基础知识。

3、熟练掌握空域变换增强的matlab 相关函数用法，并能利用算法自己编写matlab 程序实现图像空域变换增强。

二、实验内容

1、用函数imcomplement 对灰度图像cameraman.tif 取反。

2、利用如下分段变换曲线对canmeraman.tif 做线性灰度变换。

3、利用直方图求取算法计算以及显示pout.tif 的直方图，并和imhist 函数生成的直方图作比较。

三、实验原理

在图像处理中，空域是指由像素组成的空间，空域增强方法是指直接作用于像素的增强方法。空域处理可以表示为：

g(x, y)=T[ f(x, y)]

j=imcomplement(i);对图像取反 imhist();显示图像的直方图 histeq();直方图均衡化函数

imnoise(I,type,parameters);给图像加噪声 bitand();图像位与运算 bitor();图像位或运算

四、实验代码与结果

1、i=imread(\'cameraman.tif\');

j=imcomplement(i);subplot(121),imshow(i)subplot(122),imshow(j)

2、clear,close all,clc in1=imread(\'cameraman.tif\');

f0=0;g0=0;f1=100;g1=60;f2=150;g2=220;f3=255;g3=255;figure,plot([f0,f1,f2,f3],[g0,g1,g2,g3])axis tight,xlabel(\'f\'),ylabel(\'g\')title(\'intensity transformation\')r1=(g1-g0)/(f1-f0);b1=g0-r1\*f0;r2=(g2-g1)/(f2-f1);b2=g1-r2\*f1;r3=(g3-g2)/(f3-f2);b3=g2-r3\*f2;[m,n]=size(in1);in2=double(in1);for i=1:m for j=1:n f=in2(i,j);g(i,j)=0;if(f>=0)&(f=f1)&(f=f2)&(fsize(X)Same：返回二维户相关结果的中间部分，Y 的大小与X 相同 Valid：返回二维户相关未使用边缘补0 的部分，size(Y)=0)&(f=f11)&(f=f12)&(f=0)&(f=f21)&(f=f22)&(f=0)&(f=f31)&(f=f32)&(f0 u0=u0+i\*p(i)/w0;else u0=u0+0;end end u1=0;for i=(t+1):L if w1>0 u1=u1+i\*p(i)/w1;else u1=u1+0;end end rou(t+1)=w0\*w1\*(u0-u1)^2;%类间方差的求取 end maxrou=max(rou);%类间方差的最大值 for i=1:L if rou(i)==maxrou r=i-1;end end r %类间方差最大所对应的灰度值 for i=1:m for j=1:n if in(i,j)>r out(i,j)=1;else out(i,j)=0;end end end subplot(121),imshow(in)subplot(122),imshow(out)运行结果得到的阈值为：114

**第五篇：图像处理 实验报告**

摘要：

图像处理，用计算机对图像进行分析，以达到所需结果的技术。又称影像处理。基本内容 图像处理一般指数字图像处理。数字图像是指用数字摄像机、扫描仪等设备经过采样和数字化得到的一个大的二维数组，该数组的元素称为像素，其值为一整数，称为灰度值。图像处理技术的主要内容包括图像压缩，增强和复原，匹配、描述和识别3个部分。图像处理一般指数字图像处理。

数字图像处理的目的是改善图像的质量，它以人为对象，以改善人的视觉效果为目的。目前，图像处理演示系统应用领域广泛医学、军事、科研、商业等领域。因为数字图像处理技术易于实现非线性处理，处理程序和处理参数可变，故是一项通用性强，精度高，处理方法灵活，信息保存、传送可靠的图像处理技术。本图像处理演示系统以数字图像处理理论为基础，对某些常用功能进行界面化设计，便于初级用户的操作。设计要求

可视化界面，采用多幅不同形式图像验证系统的正确性；

合理选择不同形式图像，反应各功能模块的效果及验证系统的正确性 对图像进行灰度级映射，对比分析变换前后的直方图变化；

1.课题目的与要求 目的：

基本功能：彩色图像转灰度图像

图像的几何空间变换：平移，旋转，剪切，缩放 图像的算术处理：加、减、乘

图像的灰度拉伸方法（包含参数设置）； 直方图的统计和绘制；直方图均衡化和规定化； 要求：

1、熟悉图像点运算、代数运算、几何运算的基本定 义和常见方法；

2、掌握在MTLAB中对图像进行点运算、代数运算、几何运算的方法

3、掌握在MATLAB中进行插值的方法

4、运用MATLAB语言进行图像的插值缩放和插值旋转等

5、学会运用图像的灰度拉伸方法

6、学会运用图像的直方图设计和绘制；以及均衡化和规定化

7、进一步熟悉了解MATLAB语言的应用，将数字图像处理更好的应用于实际

2.课题设计内容描述

1>彩色图像转化灰度图像：

大部分图像都是RGB格式。RGB是指红，绿，蓝三色。通常是每一色都是256个级。相当于过去摄影里提到了8级灰阶。

真彩色图像通常是就是指RGB。通常是三个8位，合起来是24位。不过每一个颜色并不一定是8位。比如有些显卡可以显示16位，或者是32位。所以就有16位真彩和32位真彩。

在一些特殊环境下需要将真彩色转换成灰度图像。1单独处理每一个颜色分量。

2.处理图像的“灰度“，有时候又称为“高度”。边缘加强，平滑，去噪，加锐度等。

3.当用黑白打印机打印照片时，通常也需要将彩色转成灰白，处理后再打印 4.摄影里，通过黑白照片体现“型体”与“线条”，“光线”。2>图像的几何空间变化：

图像平移是将图像进行上下左右的等比例变化，不改变图像的特征，只改变位置。

图像比例缩放是指将给定的图像在x轴方向按比例缩放fx倍，在y轴按比例缩放fy倍，从而获得一幅新的图像。如果fx=fy，即在x轴方向和y轴方向缩放的比率相同，称这样的比例缩放为图像的全比例缩放。如果fx≠fy，图像的比例缩放会改变原始图象的像素间的相对位置，产生几何畸变。

旋转。一般图像的旋转是以图像的中心为原点，旋转一定的角度，也就是将图像上的所有像素都旋转一个相同的角度。旋转后图像的的大小一般会改变，即可以把转出显示区域的图像截去，或者扩大图像范围来显示所有的图像。图像的旋转变换也可以用矩阵变换来表示。3>图像的算术处理：

图像代数运算是指对两幅或两幅以上输入图像对应的像素逐个进行和差积商运算以产生增强效果的图像。图像运算是一种比较简单有效的增强处理手段是图像处理中常用方法。三种图像处理代数运算的数学表达式如下： C(x,y)=A(x,y)+B(x,y)C(x,y)=A(x,y)-B(x,y)C(x,y)=A(x,y)\*B(x,y)4>图像的灰度拉伸方法：

灰度拉伸又叫对比度拉伸，它是最基本的一种灰度变换，使用的是最简单的分段线性变换函数，它的主要思想是提高图像处理时灰度级的动态范围。可以有选择的拉伸某段灰度区间以改善输出图像。如图，所示的变换函数的运算结果是将原图在a到b之间的灰度拉伸到c到d之间。如果一幅图像的灰度集中在较暗的区域而导致图像偏暗，可以用灰度拉伸功能来拉伸(斜率>1)物体灰度区间以改善图像；同样如果图像灰度集中在较亮的区域而导致图像偏亮，也可以用灰度拉伸功能来压缩(斜率直方图设计和绘制；以及均衡化和规定化：

灰度直方图是将数字图像的所有像素，按照灰度值的大小，统计其所出现的频度。通常，灰度直方图的横坐标表示灰度值，纵坐标为半个像素个数，也可以采用某一灰度值的像素数占全图像素数的百分比作为纵坐标。

直方图均衡方法的基本原理是：对在图像中像素个数多的灰度值（即对画面起主要作用的灰度值）进行展宽，而对像素个数少的灰度值（即对画面不起主要作用的灰度值）进行归并。从而达到清晰图像的目的。

3.总体方案设计

1> GUI图像处理平台的总体设计

图像处理平台设计的目的是能够将图像处理的各个独立算法集成到一个平台内，方便用户选用多种方法对图像进行处理.平台基于ＭatlabGUI设计，实现图像处理过程的交互和可视化，并为用户二次开发提供平台接口，提高图像处理算法的综合利用效率

2>平台总体功能设计 根据一体化的设计思想，平台主要实现算法集成、交互可视化和提供二次开发接口等功能.其中算法集成分为已有算法集成和新算法集成.具体功能描述如下：

（1）已有算法集成是对Ｍatlab图像处理工具

包中提供的算法进行集成，可以通过使用函数名加参数的方式直接调用.依据功能进行分类，将同类算法集成到同一菜单项内，如将傅里叶变换、小波变换、离散变换等算法归类到图像变换中，进行集成.（2）新算法集成是指对自主开发的算法进行集成，如改进水平集算法[12]、交互式图割算法[13]、细胞自动机分割算法[14]等均为自主开发的图像分割算法，同已有算法集成方式类似，集成到平台中，便于综合运用和算法分析与对比.（3）交互式可视化是指对图像处理过程及结果的可视化显示，并提供用户交互区.（4）二次开发接口是指通过调用集成模板方式，为用户提供一个将自己算法集成到平台中的一个接口

3>总体布局设计

一个高性能的图像处理平台应该为用户提供

方便快捷的操作.平台设计中通过菜单和按钮实现快捷操作，其中菜单项提供平台的整体功能，快捷按钮显示具体的独立功能.图像处理平台的布局设计如图1所示.利用ＭatlabGUI提供的工具包和底层代码，可以实现菜单功能区、快捷按钮功能区、DEMO显示区、可视化效果显示区和用户交互区的布局设计

4.程序实现和测试

4.1各个功能模块的主要实现程序 基本功能：彩色图像转灰度图像

I=imread(\'C:MATLAB7toolboximagesimdemospeppers.png\',\'png\');x=rgb2gray(I);figure(1);subplot(121);imshow(I);title(\'原始图像\');subplot(122);imshow(x);title(\'灰度图像\');实验结果：

图像的几何空间变换：平移，旋转，剪切，缩放

img1=imread(\'C:MATLAB7toolboximagesimdemosguidemo123jujiao3.jpg\',\'jpg\');img1=rgb2gray(img1);figure,imshow(img1);imwrite(img1,\'a1.jpg\');%%%%%%平移

se=translate(strel(1),[20 20]);img2=imdilate(img1,se);figure,imshow(img2);imwrite(img2,\'a2.jpg\');%%%%%%旋转 img3=imrotate(img1,90);figure,imshow(img3);imwrite(img3,\'a3.jpg\');% %%%%%缩放 img4=imresize(img1,2);figure,imshow(img4);imwrite(img4,\'a4.jpg\');

原始图像：

平移图像：

旋转图像：

缩放图像：

图像的算术处理：加、减、乘

加法

A=imread(\'C:MATLAB7toolboximagesimdemosguidemo123jujiao3.jpg\',\'jpg\');B=imread(\'C:MATLAB7toolboximagesimdemosguidemo123jujiao1.jpg\',\'jpg\');subplot(1,3,1);imshow(A);title(\'图像1\');subplot(1,3,2);imshow(B);title(\'图像2\');C=imadd(A,B);subplot(1,3,3);imshow(C);title(\'相加后的图像\')

减法

A=imread(\'C:MATLAB7toolboximagesimdemosguidemo123jujiao3.jpg\',\'jpg\');B=imread(\'C:MATLAB7toolboximagesimdemosguidemo123jujiao1.jpg\',\'jpg\');subplot(2,3,1);imshow(A);title(\'图像1\');subplot(2,3,2);imshow(B);title(\'图像2\');C=imsubtract(A,B);subplot(2,3,3);imshow(C);title(\'相减后的图像\')

乘法

A=imread(\'C:MATLAB7toolboximagesimdemosguidemo123jujiao3.jpg\',\'jpg\');B=imread(\'C:MATLAB7toolboximagesimdemosguidemo123jujiao1.jpg\',\'jpg\');subplot(1,3,1);imshow(A);title(\'图像1\');subplot(1,3,2);imshow(B);title(\'图像2\');C=immultiply(A,B);subplot(1,3,3);imshow(C);title(\'相乘后的图像\')

图像的灰度拉伸方法（包含参数设置）；

img=imread(\'C:MATLAB7toolboximagesimdemosguidemo123jujiao3.jpg\',\'jpg\');figure(1);imshow(img);title(\'原图\');[m,n]=size(img);%测量图像尺寸参数

GreyHist=zeros(1,256);%预创建存放灰度出现概率的向量 for k=0:255 GreyHist(k+1)=length(find(img==k))/(m\*n);%计算每级灰度出现的概率，将其存入GreyHist中相应位置

end figure(2);bar(0:255,GreyHist)%绘制直方图 title(\'原直方图\')xlabel(\'灰度值\')ylabel(\'出现概率\')%灰度拉伸 imggrey=img;prompt={\'请输入系数a\',\'请输入系数b\'};words=\'请输入线性拉伸函数:\';answer = inputdlg(prompt,words,1,{\'0.5\',\'2\'});a=str2double(answer(1));b=str2double(answer(2));for i=1:m for j=1:n img(i,j)=a\*img(i,j)+b;end end figure(3);imshow(img);title(\'灰度拉伸\');GreyHist=zeros(1,256);%预创建存放灰度出现概率的向量 for k=0:255

直方图的统计和绘制；直方图均衡化和规定化；

%一，图像的预处理，读入彩色图像将其灰度化

img=imread(\'C:MATLAB7toolboximagesimdemosguidemo123jujiao3.jpg\',\'jpg\');%读入JPG彩色图像文件

imshow(img)%显示出来 title(\'输入的彩色JPG图像\')imwrite(rgb2gray(img),\'PicSampleGray.jpg\');%将彩色图片灰度化并保存 img=rgb2gray(img);%灰度化后的数据存入数组 %二，绘制直方图

[m,n]=size(img);%测量图像尺寸参数

GP=zeros(1,256);%预创建存放灰度出现概率的向量 for k=0:255 GP(k+1)=length(find(img==k))/(m\*n);%计算每级灰度出现的概率，将其存入GP中相应位置 end figure,bar(0:255,GP,\'g\')%绘制直方图 title(\'原图像直方图\')xlabel(\'灰度值\')ylabel(\'出现概率\')%三，直方图均衡化 S1=zeros(1,256);for i=1:256 for j=1:i S1(i)=GP(j)+S1(i);%计算Sk end end S2=round((S1\*256)+0.5);%将Sk归到相近级的灰度 for i=1:256 GPeq(i)=sum(GP(find(S2==i)));%计算现有每个灰度级出现的概率 end figure,bar(0:255,GPeq,\'b\')%显示均衡化后的直方图 title(\'均衡化后的直方图\')xlabel(\'灰度值\')ylabel(\'出现概率\')%四，图像均衡化 PA=img;for i=0:255 PA(find(img==i))=S2(i+1);%将各个像素归一化后的灰度值赋给这个像素 end figure,imshow(PA)%显示均衡化后的图像 title(\'均衡化后图像\')imwrite(PA,\'PicEqual.jpg\');

4.3.问题说明和总结：对在调试中发现的问题和解决方法做说明。

图像处理，是对图像进行分析、加工、和处理，使其满足视觉、心理以及其他要求的技术。图像处理是信号处理在图像域上的一个应用。目前大多数的图像是以数字形式存储，因而图像处理很多情况下指数字图像处理。

进行程序调试的过程中，最重要的是输入图像，刚开始是为了找到原始图像耗费了很多时间，一般的条件书上有，但要对其进行磨合。程序编写时，应该注意大小写。应该注意最后的输出部分，保证输出条件与输入条件相同。

5．总结与体会 这次使用MATALB进行图像处理的编写，是我对MATALB软件有了更深入的了解，对其的应用能力也有了相应的提高，更深入的了解到MATALB作为绘图软件的方便与快捷。在进行程序调试的过程中，最重要的是输入图像，只有找到图像的原始位置，才能进行下面的程序编码。编码程序时，应该在MATLAB原始文档的位置先行输入，输入时应该注意大小写。程序应该尽可能地简单，只要能达到目的就行，程序越复杂，运行时的错误就越多。以上是我的程序编码经验与感受。

6．参考文献

《数字图像处理实验指导书》 厍向阳 曹颖超 编著 《MATLAB与数学实验》 艾冬梅 李艳晴 编著 《图像处理和分析技术》 章毓晋 编著 《MATLAB实用教程》 郑阿奇 编著

本DOCX文档由 www.zciku.com/中词库网 生成，海量范文文档任你选，，为你的工作锦上添花,祝你一臂之力！