

贵州大学实验报告

学院：大数据与信息工程学院

专业：通信工程

班级：通信 152 班

姓名	朱桐	学号	1500890087	实验组	
实验时间	2018/05/30	指导教师	刘洪	成绩	
实验项目名称	实验一 图像的基本运算及常用变换				
实验目的	<ol style="list-style-type: none">1. 通过本实验的学习使学生熟悉 MATLAB 图像处理工具箱的编程和使用；2. 掌握数字图像的基本算术运算和逻辑运算的原理和应用；3. 掌握常用的变换：二维离散傅立叶变换、二维离散余弦变换的正反变换，及一些性质的验证。				
实验要求	<ol style="list-style-type: none">1. 认真阅读实验手册，严格遵守实验室安全管理规定，规范使用设备；2. 保存实验过程中的输出图像及结果；3. 认真撰写实验报告并及时上交。				
实验原理	<p>1、图像文件的读取：imread 函数</p> <p>语法：</p> <pre>A = imread('filename',fmt) [X,map] = imread('filename',fmt) [...] = imread('filename') [...] = imread(... ,idx) (CUR, ICO, and TIFF only) [...] = imread(... ,ref) (HDF only) [...] = imread(... ,'BackgroundColor',BG) (PNG only) [A,map,alpha] = imread(...) (PNG only)</pre> <p>2、图像文件的显示：imshow 函数</p> <p>当用户调用 imshow 函数显示一幅图像时，该函数将自动设置图像窗口、坐标轴和图像属性。这些自动设置的属性具体包括图像对象的 CData 属性和 CDataMapping 属性、坐标轴对象的 CLim 属性、图像窗口对象的 Colormap 属性。</p>				

另外，根据用户使用参数的不同，`imshow` 函数在调用时除了完成前面提到的属性设置外还可以完成以下的操作：

设置其它的图形窗口对象的属性和坐标轴对象的属性以优化显示效果，如可以设置隐藏坐标轴及其标示等。包含和隐藏图像边框。调用 `true_size` 函数以显示没有彩色渐变效果的图像。

语法：

```
imshow(I,n)
imshow(I,[low high])
imshow(BW)
imshow(X,map)
imshow(RGB)
imshow(...,display_option)
imshow(x,y,A,...)
imshow filename
h = imshow(...)
```

3、在图像处理中一种最简单的图像恢复算法是利用其邻域像素的值求和再取平均来得到改点的一个恢复值：

$$f(x,y)=1/9[f(x-1,y-1)+f(x-1,y)+f(x-1,y+1)+f(x,y) \\ f(x,y-1)+f(x,y+1)+f(x+1,y-1)+f(x+1,y)+f(x+1,y+1)]$$

4、运动检测原理：在序列图像中，通过逐像素比较可直接求取前后两帧图像之间的差别。假设照明条件在多帧图像间基本不变化，那么差图像中的不为零处表明该处的像素发生了移动。换句话说，对时间上相邻的两幅图像求差可以将图像中目标的位置和形状变化突出出来。

5、Matlab 中的 `fft2` 函数

语法：

```
Y = fft2(X)
Y = fft2(X,m,n)
```

6、Matlab 中的 `ifft2` 函数

语法：

```
Y = ifft2(X)
```

	<p>$Y = \text{ifft2}(X, m, n)$</p> <p>7、 二维傅立叶变换的平移性质。</p> <p>8、 Matlab 中二维离散余弦变换函数 <code>dct2</code></p> <p>语法：</p> <p>$B = \text{dct2}(A)$</p> <p>$B = \text{dct2}(A, m, n)$</p> <p>$B = \text{dct2}(A, [m \ n])$</p> <p>9、 Matlab 中的函数 <code>dctmtx</code></p> <p>此函数可以用来求离散余弦变换的变换矩阵 D。</p> <p>语法：</p> <p>$D = \text{dctmtx}(n)$</p> <p>10、 DCT 域的图像压缩原理。</p> <p>离散余弦变换 DCT (Discrete Cosine Transform) 是数码率压缩的一个常用的变换编码方法。DCT 是先将整体图像分成 $N \times N$ 像素块，然后对 $N \times N$ 像素块逐一进行 DCT 变换。在变换编码中一个很重要的因素是块的大小。最通用的大小是 8×8 和 16×16，由于计算方面的原因，两个都是 2 的乘方。若是采用 8×8 的变换块，则得到的变换为 $C(u, v)$ 称为 DCT 系数。</p> <p>此矩阵的左上角系数 $G(1,1)$ 相当于最低频率的系数，简称为子图像的直流 (DC) 系数或直流成分；除此之外的 $C(u, v)$ 均为交流系数。在右下角的系数 $G(8,8)$ 是最高频率的系数；在右上角的系数 $G(1,8)$，表示水平方向频率最高，垂直方向频率最低的系数；在左下角的系数 $G(8,1)$，表示垂直方向频率最高，水平方向频率最低的系数。</p> <p>另外在进行 DCT 变换以前，二维空间图像亮度数据通常较高，应该先进行向下电平移位。如果是 256 灰度级，则将亮度值剪掉 128。</p> <p>由于大多数图像的高频分量较少，相应于图像高频分量的系数经常为零，加之人眼对高频成分的失真不太敏感，所以可用更粗的量化。因此，传送变换系数的数码率要大大小于传送图像像素所用的数码率。到达接收端后通过反离散余弦变换回到原值。虽然会有一定的失真，但人眼是可以接受的。</p> <p>例：有一幅灰度图像为：</p>
--	--

	139 144 149 153 155 155 155 155
	144 151 153 156 159 156 156 156
	150 155 160 163 158 156 156 156
	159 161 162 160 160 159 159 159
	159 160 161 162 162 155 155 155
	161 161 161 161 160 157 157 157
	162 162 161 163 162 157 157 157
	162 162 161 161 163 158 158 158
	电平移位后为：
	11 16 21 25 27 27 27 27
	16 23 25 28 31 28 28 28
	22 27 32 35 30 28 28 28
	31 33 34 32 32 31 31 31
	31 32 33 34 34 27 27 27
	33 33 33 33 32 29 29 29
	34 34 33 35 34 29 29 29
	34 34 33 33 35 30 30 30
	DCT 系数
	235.6 1.0 12.1 5.2 2.1 1.7 2.7 1.3
	22.6 17.5 6.2 3.2 2.9 0.1 0.4 1.2
	10.9 9.3 1.6 1.5 0.2 0.9 0.6 0.1
	7.1 1.9 0.2 1.5 0.9 0.1 0.0 0.3
	0.6 0.8 1.5 1.6 0.1 0.7 0.6 1.3
	1.8 0.2 1.6 0.3 0.8 1.5 1.0 1.0
	1.3 0.4 0.3 1.5 0.5 1.7 1.1 0.8
	2.6 1.6 3.8 1.8 1.9 1.2 0.6 0.4
	从上面的 DCT 变换系数矩阵可以看出高频分量占得很少。所以对于系数小的我们可以分配较少的比特，大的系数可以分配较多的比特，对其进行量化编码。

实验仪器	<ol style="list-style-type: none"> 硬件环境 计算机 一台 软件环境 MATLAB 2016b Windows10 64bit Professional
实验步骤	<ol style="list-style-type: none"> 读取一个彩色图像文件并显示。将此图像转换为 256 色的灰度图像。用 <code>subimage</code> 函数将彩色图像和灰度图像同时显示。 读取灰度图像。用 <code>imnoise</code> 函数加入椒盐噪声；采用 8 邻域像素平均的方法对有噪声的图像去噪。同时显示原图像，噪声图像，和恢复后的图像。加入不同参数的椒盐噪声比较实验结果。 编写程序。产生一个 50*50 的矩阵。其中第 (20,10) 到 (20,15) 的值为 255。其余矩阵元素为 0。对该矩阵做二维傅立叶变换，并显示其频谱图。将该图像中的目标向 y 方向平移。即将 (20,35) 到 (20,40) 的元素置为 255。其余值为 0。对该矩阵做二维傅立叶变换，并显示其频谱图。对两频谱图进行比较，得出结论。依照以上方法验证傅立叶变换的 X 方向的平移性质。 对给定图像进行二维 DCT 变换，并将 DCT 系数矩阵中值小于给定阈值的元素置为 0，然后使用反离散余弦变换重建图像。同时显示原图像和重建图像。
实验内容	<ol style="list-style-type: none"> 彩色图像文件的读取和显示； 利用 8 邻域平均法对图像去噪； 编写程序。读取图像 <code>manface1.jpg</code> 和 <code>manface2.jpg</code>。用两图像相减的方式获取运动轨迹； 二维离散傅立叶变换的平移性质验证； 二维离散余弦变换的应用。
实验数据	<p>实验代码：</p> <pre> %% initialization close all clear, clc %% Q1 im = imread('..\data\jingjing.jpg'); % convert to gray image with a 256 degree </pre>

```

imGray = uint8(rgb2gray(im));

figure(1)
subplot(1, 2, 1)
imshow(im)
title('Original Image')
subplot(1, 2, 2)
imshow(imGray)
title('256 Degree Gray Image')

%% Q2
imWithGaussianNoise = imnoise(imGray, 'gaussian', 0, 0.2);
imWithSaltPepperNoise = imnoise(imGray, 'salt & pepper', 0.2);

imFilterGaussianNoise = eightNeighborFilter(imWithGaussianNoise);
imFilterSaltPepperNoise = eightNeighborFilter(imWithSaltPepperNoise);

figure(2)

subplot(3, 2, 1:2)
imshow(imGray)
title('Original Image')

subplot(3, 2, 3)
imshow(imWithGaussianNoise)
title('Gaussian Noise Mean:0 Var:0.2')
subplot(3, 2, 4)
imshow(imFilterGaussianNoise)
title('After Filter')

subplot(3, 2, 5)
imshow(imWithSaltPepperNoise)
title('Salt & Pepper Noise d:0.2')
subplot(3, 2, 6)
imshow(imFilterSaltPepperNoise)
title('After Filter')

%% Q3
mat = zeros(50, 50);
mat(20, 10:15) = 255;
figure(3)
subplot(4,2,1)
imshow(mat)
title('Original Image')

```

```
subplot(4,2,2)
matFFT = abs(fft2(mat));
% matFFT = 10.*log(matFFT);
matFFT = mat2gray(fftshift(matFFT));
imshow(matFFT)
title('After FFT')
```

```
mat = zeros(50, 50);
mat(20, 35:40) = 255;
subplot(4,2,3)
imshow(mat)
title('Original Image')
```

```
subplot(4,2,4)
matFFT = abs(fft2(mat));
% matFFT = 10.*log(matFFT);
matFFT = mat2gray(fftshift(matFFT));
imshow(matFFT)
title('After FFT')
```

```
mat = zeros(50, 50);
mat(10, 10:15) = 255;
subplot(4,2,5)
imshow(mat)
title('Original Image')
```

```
subplot(4,2,6)
matFFT = abs(fft2(mat));
% matFFT = 10.*log(matFFT);
matFFT = mat2gray(fftshift(matFFT));
imshow(matFFT)
title('After FFT')
```

```
mat = zeros(50, 50);
mat(10:15, 10) = 255;
subplot(4,2,7)
imshow(mat)
title('Original Image')
```

```
subplot(4,2,8)
matFFT = abs(fft2(mat));
% matFFT = 10.*log(matFFT);
matFFT = mat2gray(fftshift(matFFT));
imshow(matFFT)
```

```

title('After FFT')

%% Q4

imDct = dct2(imGray);
figure(4)
subplot(1,3,1)
imshow(imGray)
title('Gray Image')
subplot(1,3,2)
imshow(log(abs(imDct)), [])
colormap(gca,jet(64))
title('After DCT')
colorbar
subplot(1,3,3)
imRe = imDct;
imRe(abs(imRe) < 0.01) = 0;
imshow(idct2(imRe), [0, 255])
title('Re-construction')

function re = eightNeighborFilter(im)
[rows, cols] = size(im);
C = im;
for i = 2:rows-1
    for j = 2:cols-1
        C(i, j) = sum(sum( C(i-1:i+1, j-1:j+1) ))/9;
    end
end
re = C;
end

```

实验图像结果：

Original Image



256 Degree Gray Image



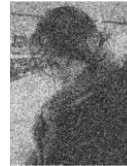
Original Image



Gaussian Noise Mean:0 Var:0.2



After Filter



Salt & Pepper Noise d:0.2



After Filter



Original Image



After FFT



Original Image



After FFT



Original Image



After FFT



Original Image



After FFT



	<div><div><div>Gray Image</div></div><div><div>After DCT</div></div><div><div>Re-construction</div></div></div>
实验总结	<p>在本次实验中，练习使用了图像处理的基本函数。在这些函数中，值得注意的是 <code>fft2</code> 和 <code>dct2</code>。由于其变换时会产生复数，所以对于这两个函数的结果可视化中，不能直接使用绘图函数绘图，而是应该先取绝对值（<code>abs</code>）后再绘图，即查看幅度响应。</p> <p>时域图像在平移时频谱不会发生变化，但是在旋转时频谱也会随之旋转。其旋转关系为逆方向旋转。</p>
指导教师意见	<div>签名：_____ 年 月 日</div>