

数字图像处理

实验指导书

数字图像处理教研组 主编

系别： 自动化测试与控制系

班级: _____

姓名:

学号:

哈尔滨工业大学

自动化测试与控制系

2018 年 10 月

实验 1. 图像的正交变换实验

一、实验目的

通过观察比较傅里叶变换的频谱和离散余弦变换频谱，掌握了解各频谱分量与图像的关系，并学习正交变换有关性质，进一步了解其在图像处理中的作用。

二、实验内容

图像素材：使用 CCD 拍摄的人脸头像，转化为大小为 256×256 ，灰度为 256 级的灰度图像。完成以下计算内容：

1. 获得傅里叶变换的幅值谱图、相位谱图；
2. 编写离散余弦变换及滤除某些成分后的复原图像。

三、实验原理

1. 离散傅里叶变换

一幅 $N \times N$ 的数字图像 $f(x, y)$ 的二维离散傅里叶变换 $F(u, v)$ 与反变换 $f(x, y)$ 分别定义如下：

$$\begin{aligned} F(u, v) &= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[\frac{-2\pi j}{N} (ux + vy) \right] \\ &= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) W_N^{-ux} W_N^{-vy}, \quad u, v = 0, 1, K, N-1 \\ f(x, y) &= \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[\frac{2\pi j}{N} (ux + vy) \right] \\ &= \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) W_N^{ux} W_N^{vy}, \quad x, y = 0, 1, K, N-1 \end{aligned}$$

这里， $W_N = \exp(2\pi j / N)$ 。变换的幅度谱： $|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)}$ ，相位谱： $\phi(u, v) = \tan^{-1}[I(u, v) / R(u, v)]$ ， $R(u, v)$ 和 $I(u, v)$ 分别为 $F(u, v)$ 的实部和虚部。

设图像空间的采样间隔为 dx 和 dy ，那么频域的坐标间隔为：

$$df_x = \frac{1}{Ndx}, \quad df_y = \frac{1}{Ndy},$$

频率的坐标点可以表示为:

$$f_{x,n} = df_x \cdot [[0:N-1]-N/2], \quad f_{y,n} = df_y \cdot [[0:N-1]-N/2]$$

2. 离散余弦变换

将一幅 $N \times N$ 的数字图像 $f(x, y)$ 沿水平方向对折镜像, 再沿垂直方向对折镜像, 可成为一个 $2N \times 2N$ 的偶函数图像, 那么它的二维正反余弦变换由下面两式定义:

$$C(u, v) = a(u)a(v) \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]$$

$$u, v = 0, 1, K, N-1$$

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} a(u)a(v)C(u, v) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right]$$

$$\text{式中: } a(v) = \begin{cases} \frac{1}{\sqrt{2}} & v = 0 \\ 1 & v = 1, 2, 3, L, N-1 \end{cases} \quad x, y = 0, 1, K, N-1$$

可见, 离散余弦变换是离散傅里叶变换的特例, 离散余弦变换是简化傅里叶变换的重要方法。

相关 MATLAB 函数有 fft, fft2, dct, dct2, fftshift 等。在使用这些函数时, 请认真阅读相关的帮助信息。

四、实验步骤

1. 分析傅里叶变换谱

- (1) 选择一幅 256×256 像素的灰度图像使用 MATLAB 函数 imread 读入;
- (2) 利用函数 fft2 进行离散傅里叶变换, 并使用函数 fftshift 进行频谱重整, 使结果与实际频谱相吻合。
- (3) 确定变换谱的空间频率和采样间隔之间的关系。
- (4) 使用 imshow, imagesc, mesh 等 MATLAB 函数显示傅氏变换的频谱信

息，如实部、虚部、振幅和相位等。

2. 离散余弦变换

在 MATLAB2012b 版本中，使用函数 `blkproc` 进行离散余弦变换。下面是该函数的应用例子：

```
I = imread('liftingbody.png');  
fun = @(x) std2(x)*ones(size(x));  
I2 = blkproc(I,[32 32],fun);  
figure, imshow(I), figure, imshow(I2,[]);
```

具体步骤如下：

- (1) 将拍摄的图像进行 DCT 变换；
- (2) 利用例子模板进行滤波后，进行 DCT 反变换，试着更改滤波模板形式（高通或低通）；
- (3) 对结果图像进行比较分析。

五、实验报告要求

1. 实验 1

- 1) 画出实验结果图；
- 2) 说明实验中采用的具体参数；
- 3) 对结果图给出适当分析和解释。

2. 实验 2

- 1) 说明程序的工作原理，针对程序加注释；
- 2) 对实验结果进行分析说明。

实验 2. 边缘检测与轮廓跟踪

一、实验目的

了解图像边缘检测算子的数学模型，掌握利用常用的图像边缘检测算子进行图像边缘检测的方法，认识其功能及适用场合，了解和掌握轮廓跟踪的基本思想和方法。

二、实验内容

- 1. 对不同图像采用不同类型的图像边缘检测算子进行边缘检测；
- 2. 观察和对比边缘检测前后的图像，掌握不同的图像边缘检测算子的特点。
- 3. 对同一图像分别采用边缘跟踪和边界跟踪方法进行边缘提取。

三、基本原理

1. 图像的边缘检测

Sobel 在 Prewitt 算子的基础上，对 4-邻域采用带权的方法计算差分，对应的模版如图 3 所示，该算子不仅能检测边缘点，而且能进一步抑制噪

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

图 1 Prewitt 算子

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

图 2 Sobel 算子

声的影响，但检测的边缘较宽。

2. Canny 算子

Canny 提出的检测边缘的三个准则是：（1）低误判率，即尽可能少地把边缘点误认为是非边缘点；（2）高定位精度，即准确的把边缘点定位在灰度变化最大的像素上；（3）抑制虚假边缘。对阶跃型边缘，Canny 算子所取得 Gaussian 函数为：

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

在其某一方向 n 上， $G(x,y)$ 的一阶方向导数为：

$$G(x, y)_n = \frac{\partial G(x, y)}{\partial n} = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial G(x, y)}{\partial x} \\ \frac{\partial G(x, y)}{\partial y} \end{bmatrix}$$

将图像 $f(x, y)$ 与 G_n 进行卷积，改变 n 的方向，使 $f(x, y) * G_n$ 取得最大值的方向就是梯度方向（正交于边缘走向）：

$$\frac{\partial [f(x, y) * G(x, y)_n]}{\partial \theta} = \frac{\partial \left[\cos \theta \frac{\partial G(x, y)}{\partial x} * f(x, y) + \sin \theta \frac{\partial G(x, y)}{\partial y} * f(x, y) \right]}{\partial \theta}$$

取最大值时导数为零，得到：

$$\tan \theta = \frac{\left[\frac{\partial G(x, y)}{\partial y} \right] * f(x, y)}{\left[\frac{\partial G(x, y)}{\partial x} \right] * f(x, y)}$$

此方向上的最大值为：

$$\begin{bmatrix} \frac{\partial G(x, y)}{\partial x} \\ \frac{\partial G(x, y)}{\partial y} \end{bmatrix} * f(x, y)$$

其中：

$$\begin{aligned} \frac{\partial G(x, y)}{\partial x} &= \frac{\partial \left(\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right)}{\partial x} = \frac{x}{-2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}; \\ \frac{\partial G(x, y)}{\partial y} &= \frac{\partial \left(\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right)}{\partial y} = \frac{y}{-2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \end{aligned}$$

边缘强度为：

$$A = \sqrt{\left[\frac{\partial G(x, y)}{\partial x} \right]^2 + \left[\frac{\partial G(x, y)}{\partial y} \right]^2}$$

方向为：

$$\alpha = \left[\frac{\partial G(x, y)}{\partial y} \right] / \left[\frac{\partial G(x, y)}{\partial x} \right]$$

边缘点的判断条件：

此点的边缘强度大于沿梯度方向的两个相邻像素的边缘强度；

与该像素梯度方向上相邻两点的方向差小于 45° ；

以该像素为中心的 3×3 领域中的边缘强度极大值小于某个给定的阈值。

Canny 这种算法可以减少小模版检测中的边缘中断，有利于得到较完整的线段。

3.边缘跟踪

将检测的边缘点连接成线就是边缘跟踪，线是图像分析中的一个基本而重要的内容，它是图像的一种中层符号描述，它使图像的表述更简洁。由边缘形成线特征包括两个过程：（1）可构成线特征的边缘提取；（2）将边缘连成线。连接边缘的方法很多，这里介绍光栅跟踪扫描。

光栅扫描跟踪是一种采用电视光栅行扫描顺序对遇到的像素进行分析，从而确定是否为边缘的跟踪方法，光栅跟踪的具体步骤如下：

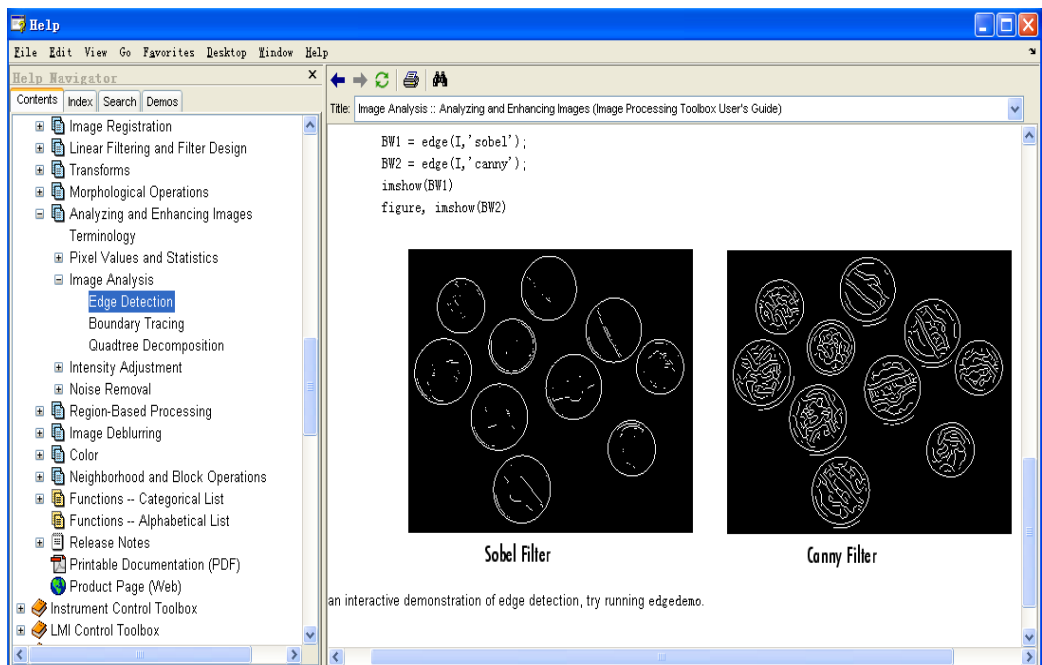


图 3 Matlab 边缘提取

1) 确定一个比较高的阈值 d ，把高于该阈值的像素作为对象点，称该阈值为“检测阈值”；

2) 用检测阈值 d 对图像第一行像素进行检测，凡超过 d 的点都接受为对象点，并作为下一步跟踪的起始点；

3) 选取一个比较低的阈值 t 作为跟踪阈值，该阈值可以根据不同准则来选择。

4. 边界跟踪

该方法是基于区域来进行的。首先需将区域进行分割，然后进行二值化处理，最后利用边界跟踪的方法，描绘出区域边界。

边界跟踪方法是：进行光栅扫描，找到像素由 0 变到 1 时的 1 像素点 (i,j) 为边界起点，以该点为原点，再以 $(i,j-1)$ 为圆的起点逆时针画半径为 1 的圆，所遇到的第一个 1 像素点为第二个边界点，由此依次顺序执行，直到回到边界起点为止。

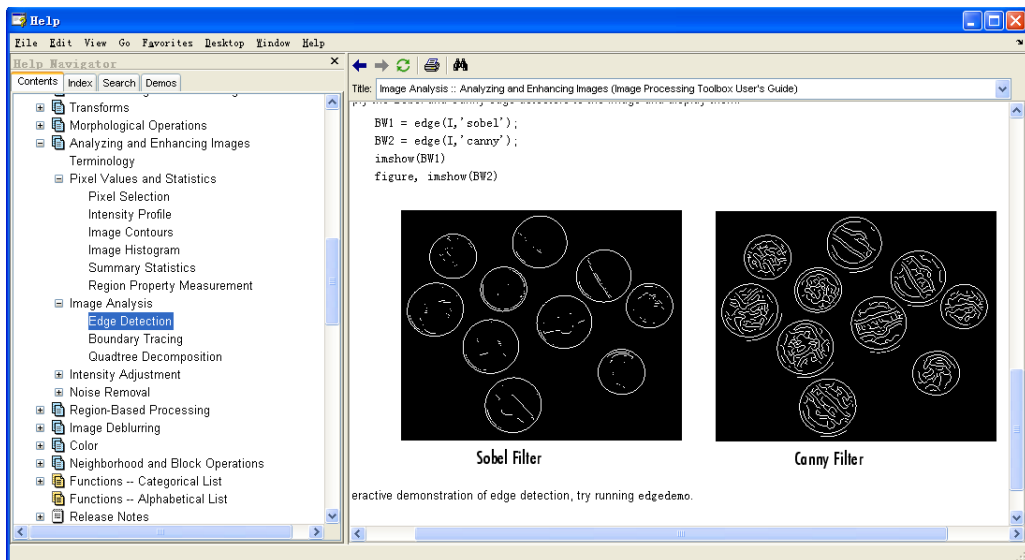


图 4 MATLAB 中图像处理工具图

四、实验步骤

1. 运行 Matlab 软件，在 Matlab 命令窗口下输入“help”，弹出 Matlab 演示程序；

2. 在 Matlab help 的左侧窗口选择“Toolboxes”->“image processing”，此时右侧窗口显示所有 Matlab 的图像处理的演示程序，选择“Edge detection”选项如图 4 所

示：

3. 双击“Edge detection”选项进入图像处理边缘检测演示程序如图 4 所示；
4. 依照例子将自己的人脸图像作为输入图像，进行人脸的跟踪处理；
5. 实现轮廓跟踪，在 Matlab 的“Help”菜单下点击“Matlab Help”，进入帮助界面如图 5 所示：
6. 在 Matlab help 的左侧窗口选择“Toolboxes”->“image processing”，此时右侧窗口显示所有 Matlab 的图像处理的演示程序，选择“Boundary Tracing”选项如图 5 所示：

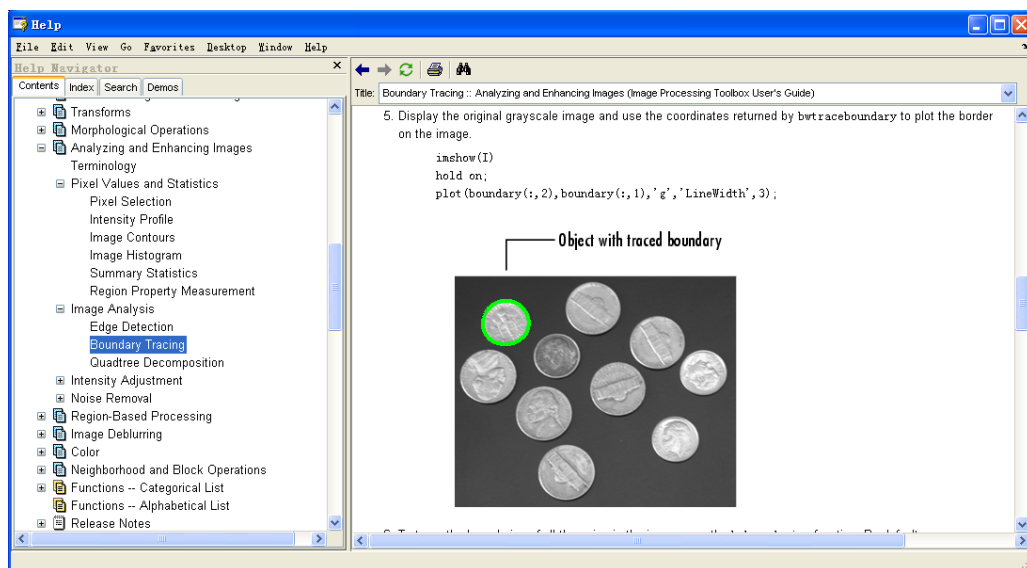


图 5 MATLAB 帮助界面

7. 按照上述办法对人脸图像进行轮廓跟踪，观察图像的轮廓跟踪结果。

相关代码：

```
I = imread('coins.png');
figure(1);imshow(I,[]);
BW = im2bw(I);
figure(2);imshow(BW,[]);
dim = size(BW);
col = round(dim(2)/2)-90;
row = min(find(BW(:,col)));
boundary = bwtraceboundary(BW,[row, col],'N');
figure(3);imshow(I,[]);
hold on;
```

```

plot(boundary(:,2),boundary(:,1),'g','LineWidth',3);

BW_filled = imfill(BW,'holes');
boundaries = bwboundaries(BW_filled);
figure(4);imshow(I,[]);hold on;
for k=1:10
    b = boundaries{k};
    plot(b(:,2),b(:,1),'g','LineWidth',3);
end

```

五、实验报告要求

1. 使用不同的边缘检测算子对个人采集的人脸图像进行边缘检测，打印输入图像和边缘检测的结果图像，并进行比较说明。
2. 使用区域分割，将人脸图像进行二值化处理，并利用边界跟踪方法进行边缘提取，给出二值化人脸图像和人脸的轮廓图像。
3. 尝试拍摄背景、以及背景+人脸图像结合的方式(得到 2 幅图像)，使用减法运算得到插值。利用形态学操作得到人脸区域，然后提取人脸区域。