云南大学

本科实验报告

课程名称： 图像处理与机器视觉

实验名称： 实验六.图像描述实验

学院（系）： 信息学院

专 业： 智能科学与技术

班 级： 2019级

姓 名： 李品鑫

学 号： 20191060239

指导教师： 高赟

成 绩：

评 语：

2022年 6 月 7 日

**一. 实验目的**

通过编程实现能够使学生掌握常用特征提取方法的基本原理，如角点检测、边缘检测、纹理检测等方法，为计算机视觉更高层次的处理奠定基础。

**二. 实验内容**

给定一幅图像，编程实现以下特征提取算法：

（1）编程实现Harris角点特征提取算法；

（2）编程实现SIFT角点特征提取算法；

（3）采用Sobel算子编程实现边缘检测算法；

（4）采用Canny算子编程实现边缘检测算法；

（5）编程实现LBP纹理检测算法；

（6）编程实现HOG特征提取算法。

**三. 实验环境**

Matlab软件是图像处理领域广泛使用的仿真软件之一。本实验基于Matlab 2018b版本完成。

**四. 实验代码**（详细注释，Times New Roman/宋体 五号字体 单倍行距）

close all;

clear all;

clc;

filename = 'lena.jpg';

f = imread(filename);

Info = imfinfo(filename);

if (Info.BitDepth > 8)

f = rgb2gray(f);

end

%Harris角点检测

%计算图像亮度f(x,y)在点(x,y)处的梯度-----------------------------------------------

ori\_im = double(f) / 255;

fx = [-2 -1 0 1 2]; % x方向梯度算子

Ix = filter2(fx, ori\_im);

fy = [-2; -1; 0; 1; 2]; % y方向梯度算子

Iy = filter2(fy, ori\_im); % y方向滤波

%构造自相关矩阵---------------------------------------------------------------

Ix2 = Ix .^ 2;

Iy2 = Iy .^ 2;

Ixy = Ix .\* Iy;

clear Ix;

clear Iy;

h= fspecial('gaussian', [7 7], 2); % 产生7\*7的高斯窗函数，sigma=2

Ix2 = filter2(h,Ix2);

Iy2 = filter2(h,Iy2);

Ixy = filter2(h,Ixy);

%提取特征点---------------------------------------------------------------

height = size(ori\_im, 1);

width = size(ori\_im, 2);

result = zeros(height, width); % 纪录角点位置，角点处值为1

R = zeros(height, width);

Rmax = 0; % 图像中最大的R值

k = 0.06;

for i = 1 : height

for j = 1 : width

M = [Ix2(i, j) Ixy(i, j); Ixy(i, j) Iy2(i, j)];

R(i,j) = det(M) - k \* (trace(M)) ^ 2; % 计算R

if R(i,j) > Rmax

Rmax = R(i, j);

end

end

end

T = 0.01 \* Rmax;%固定阈值，当R(i, j) > T时，则被判定为候选角点

%进行局部非极大值抑制-------------------------------------

cnt = 0;

for i = 2 : height-1

for j = 2 : width-1

if (R(i, j) > T && R(i, j) > R(i-1, j-1) && R(i, j) > R(i-1, j) && R(i, j) > R(i-1, j+1) && R(i, j) > R(i, j-1) && ...

R(i, j) > R(i, j+1) && R(i, j) > R(i+1, j-1) && R(i, j) > R(i+1, j) && R(i, j) > R(i+1, j+1))

result(i, j) = 1;

cnt = cnt+1;

end

end

end

i = 1;

for j = 1 : height

for k = 1 : width

if result(j, k) == 1

corners1(i, 1) = j;

corners1(i, 2) = k;

i = i + 1;

end

end

end

[posc, posr] = find(result == 1);

figure(1)

subplot(121);

imshow(f);

title("原图")

subplot(122);

imshow(ori\_im);

hold on;

plot(posr, posc, 'r+');

title("Harris角点检测结果图")

%SIFT角点检测

pic = f;

[M N]=size(pic);%计算图像像素值

mask=[0 0 1 1 1 0 0;...

0 1 0 0 0 1 0;...

1 0 0 0 0 0 1;...

1 0 0 0 0 0 1;...

1 0 0 0 0 0 1;...

0 1 0 0 0 1 0;...

0 0 1 1 1 0 0];%制作16个圆周点单位矩阵

mask=uint8(mask);%把大于255的数强制为255，也就是把灰度值映射到0-255内

threshold=25;%定义阀值

figure(2);

subplot(121);

imshow(f);

title("原图")

subplot(122);

imshow(pic);%显示灰度图像f

title('SIFT角点检测结果图');hold on;

for i=4:M-3

for j=4:N-3%若I1、I9与中心I0的差均小于阈值，则不是候选点

%计算p1、p9、p5、p13与中心p的像素差

delta1=abs(pic(i-3,j)-pic(i,j))>threshold;

delta9=abs(pic(i+3,j)-pic(i,j))>threshold;

delta5=abs(pic(i,j+3)-pic(i,j))>threshold;

delta13=abs(pic(i,j-3)-pic(i,j))>threshold;

if sum([delta1 delta9 delta5 delta13])>=3%像素差超过阀值的个数

block=pic(i-3:i+3,j-3:j+3);

block=block.\*mask;%提取圆周16个点

pos=find(block);%查找圆周点中的非0点

block1=abs(block(pos)-pic(i,j))/threshold;

block2=floor(block1);%把block1中的数向负无穷方向取整

res=find(block2);

if size(res,1)>=12 %筛选

plot(j,i,'ro');

end

end

end

end

%sobel边缘检测

[high,width] = size(f); % 获得图像的高度和宽度

F2 = double(f);

U = double(f);

uSobel = f;

for i = 2:high - 1

for j = 2:width - 1

Gx = (U(i+1,j-1) + 2\*U(i+1,j) + F2(i+1,j+1)) - (U(i-1,j-1) + 2\*U(i-1,j) + F2(i-1,j+1));

Gy = (U(i-1,j+1) + 2\*U(i,j+1) + F2(i+1,j+1)) - (U(i-1,j-1) + 2\*U(i,j-1) + F2(i+1,j-1));

uSobel(i,j) = sqrt(Gx^2 + Gy^2);

end

end

subplot(121);imshow(f);title('原图');

subplot(122);imshow(im2uint8(uSobel));title('Sobel算子边缘检测结果图'); %画出边缘检测后的图像

%Canny算子边缘检测

I = im2double(f);

BW4=edge(I,'canny');

figure(4)

subplot(121),imshow(f),title("原图");

subplot(122),imshow(BW4),title("Canny算子边缘检测结果图");

%LBP纹理检测

x=size(f,1);

y=size(f,2);

texture=uint8(zeros(x,y));

for i=2:1:x-1

for j=2:1:y-1

neighbor=uint8(zeros(1,8));

neighbor(1,1)=f(i-1,j);

neighbor(1,2)=f(i-1,j+1);

neighbor(1,3)=f(i,j+1);

neighbor(1,4)=f(i+1,j+1);

neighbor(1,5)=f(i+1,j);

neighbor(1,6)=f(i+1,j-1);

neighbor(1,7)=f(i,j-1);

neighbor(1,8)=f(i-1,j-1);

center=f(i,j);

temp=uint8(0);

for k=1:1:8

temp =temp+ (neighbor(1,k) >= center)\* 2^(k-1);

end

texture(i,j)=temp;

end

end

figure(5)

subplot(121),imshow(f),title("原图")

subplot(122),imshow(texture),title("LBP纹理检测结果图");

%HOG特征提取

[featureVector,hogVisualization] = extractHOGFeatures(f);

figure(6);

subplot(121),imshow(f),title("原图");

subplot(122);

imshow(f),title("HOG特征提取结果图");

hold on;

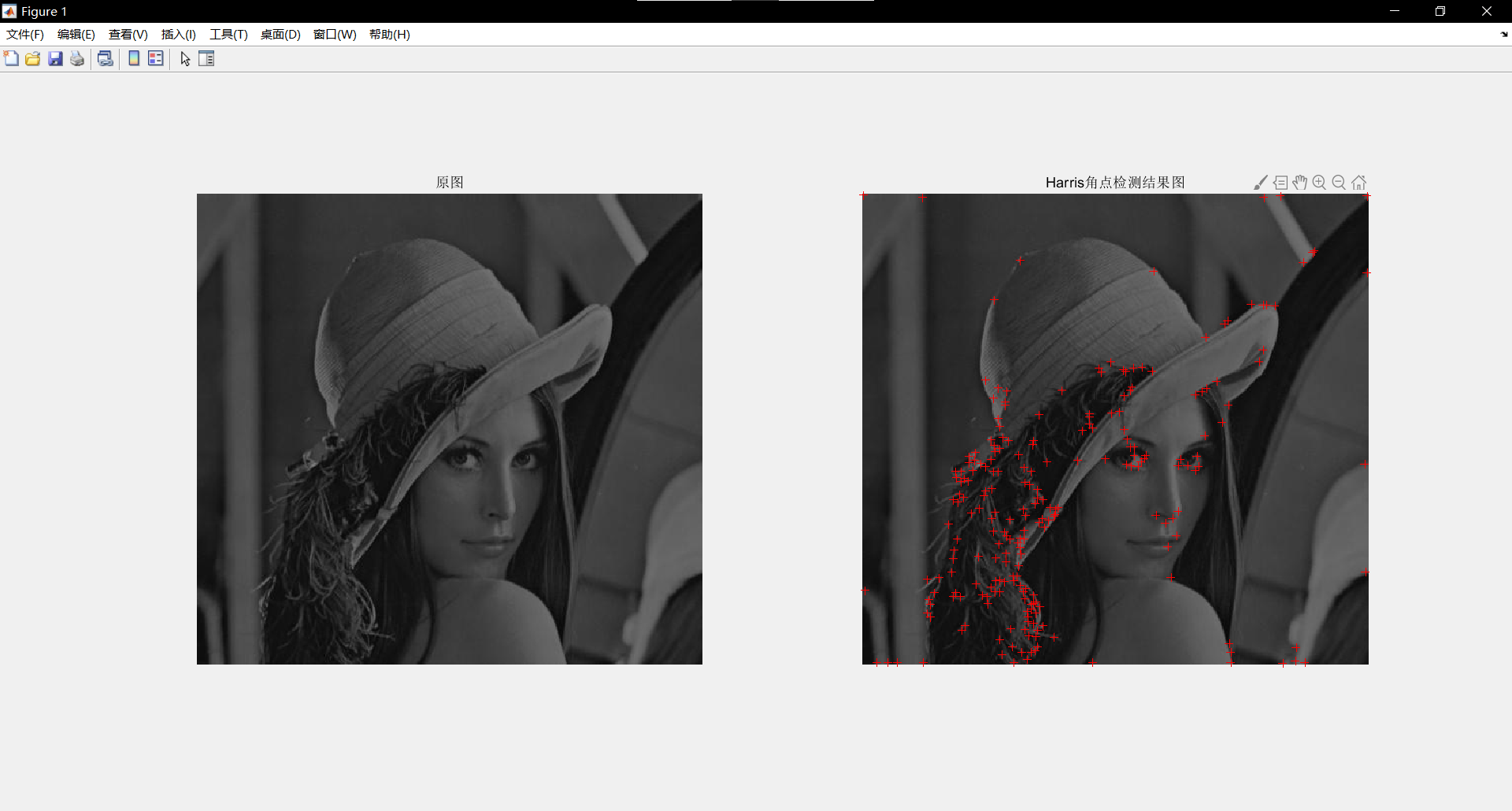
plot(hogVisualization);

附件（.m文件）：

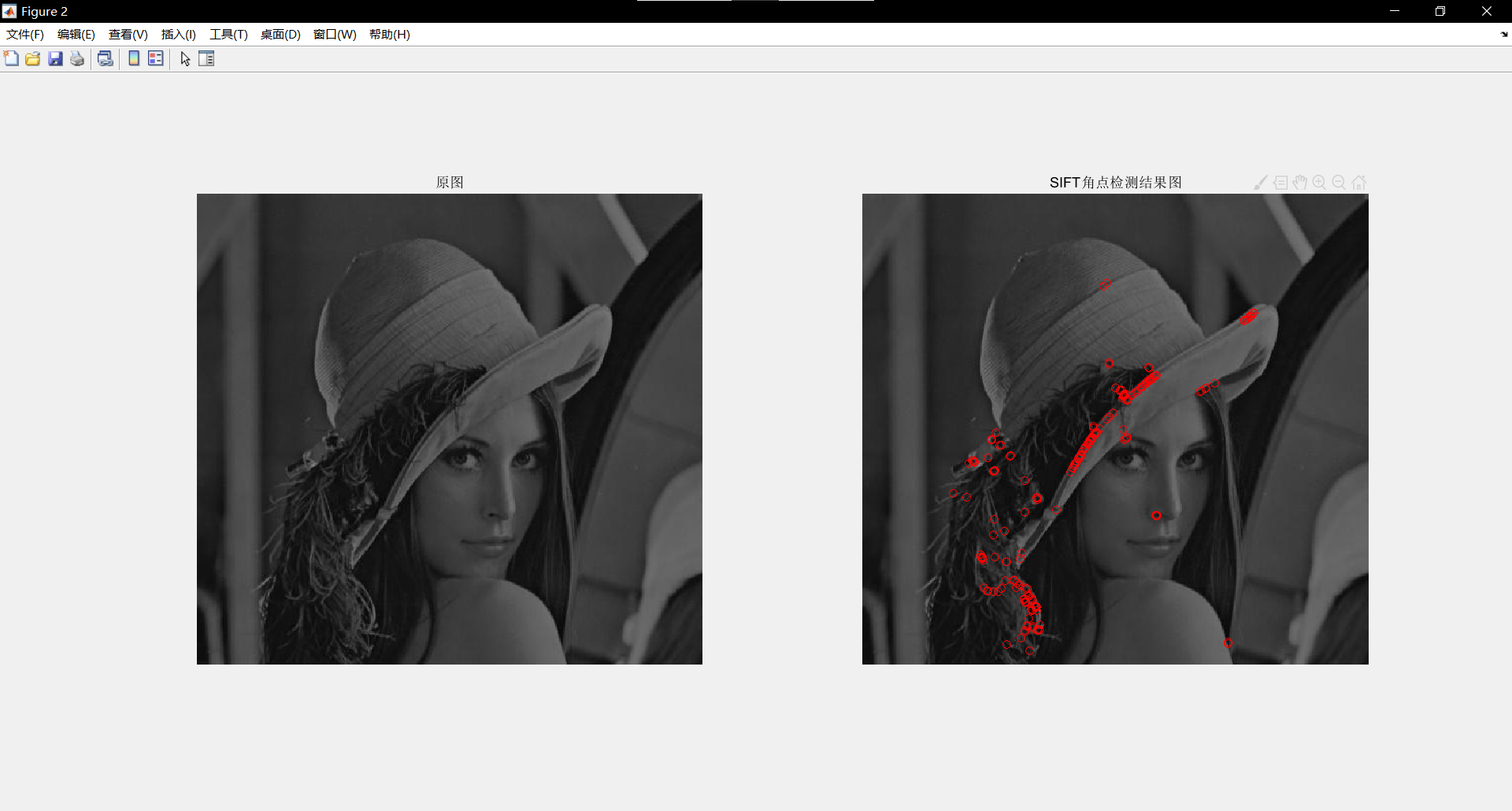


**五. 实验结果**

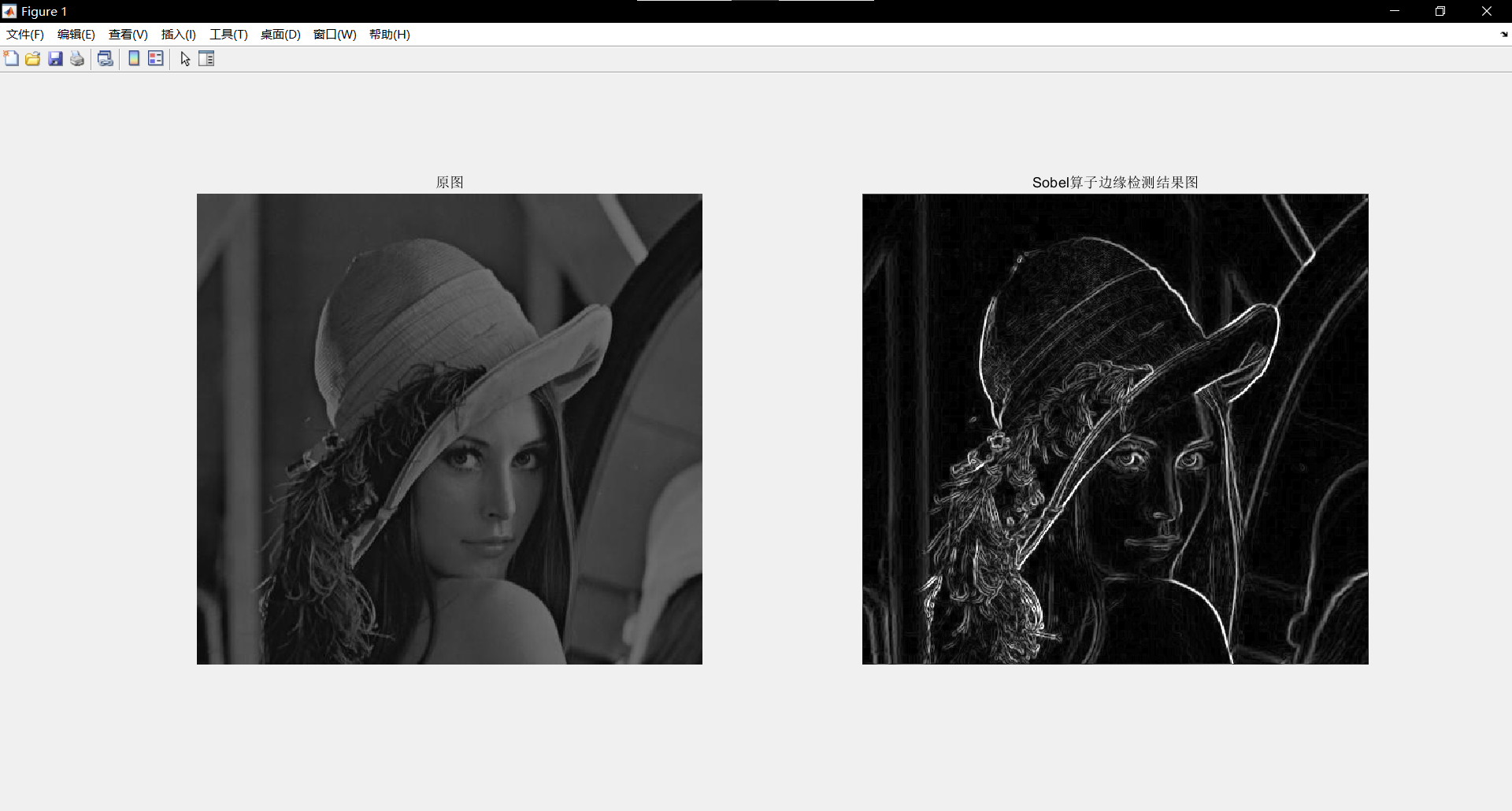
（1）Harris角点检测原图、关键过程及结果；



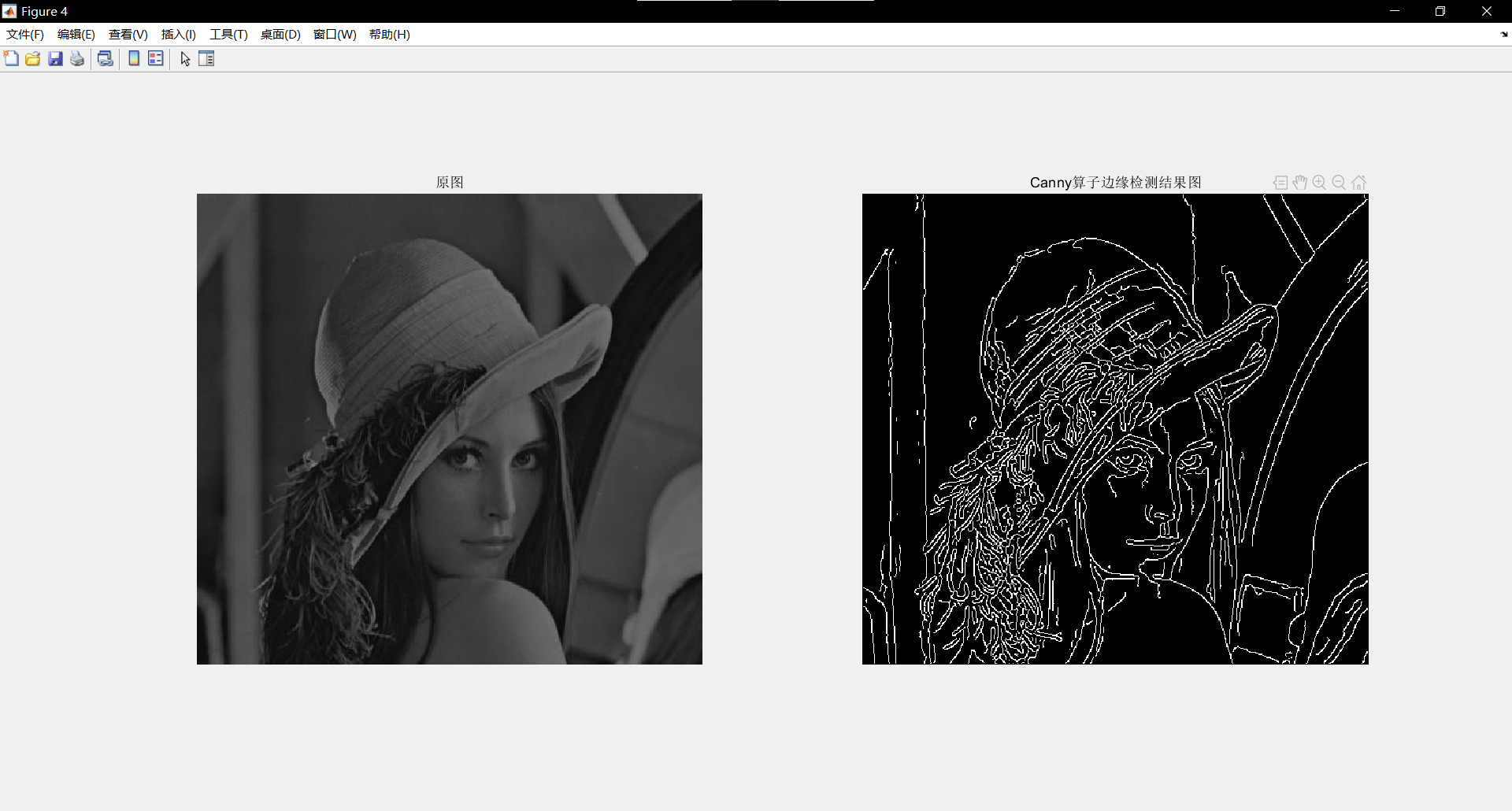
（2）SIFT角点检测原图、关键过程及结果；



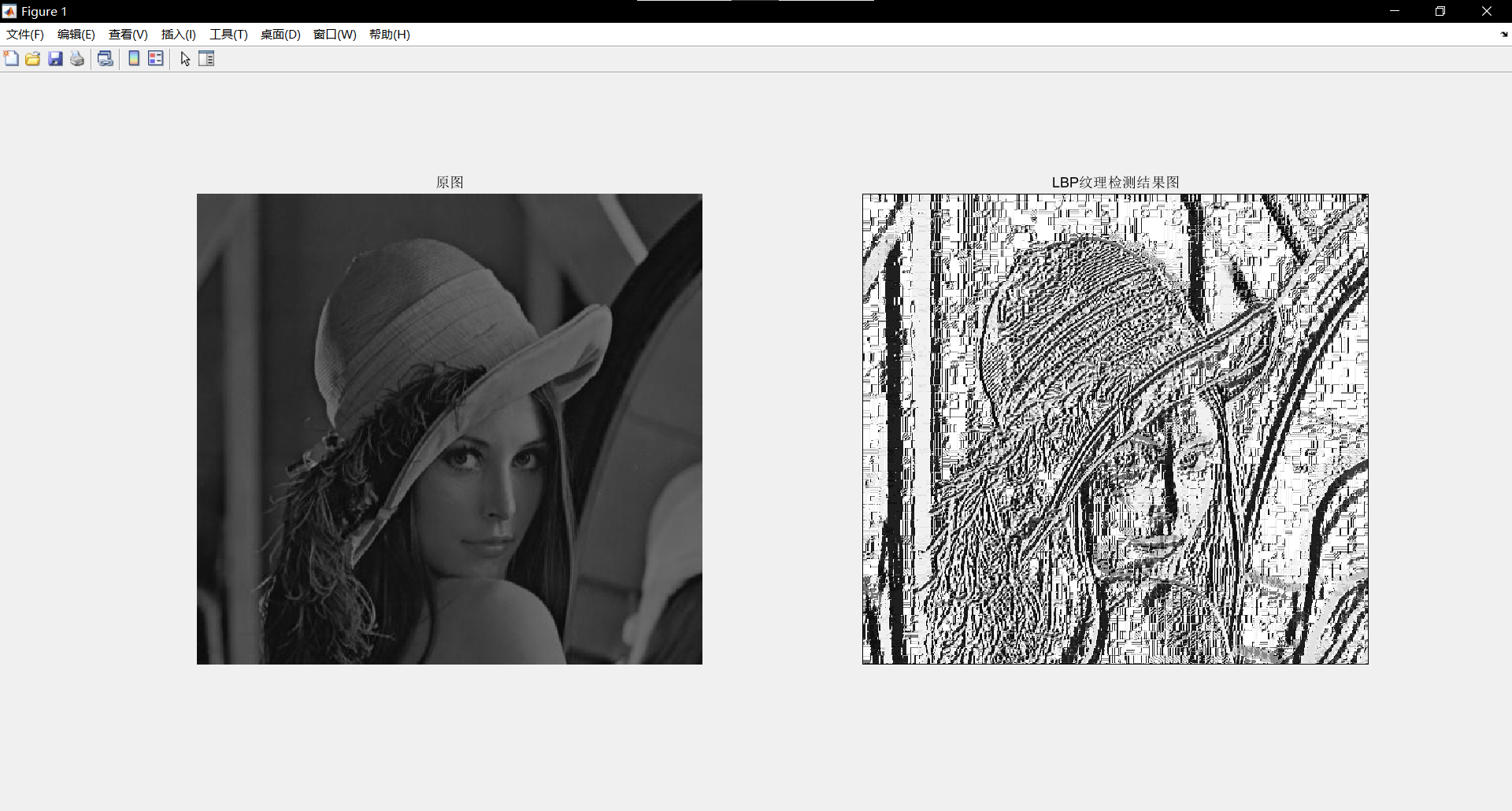
（3）Sobel算子边缘检测原图、关键过程及结果；



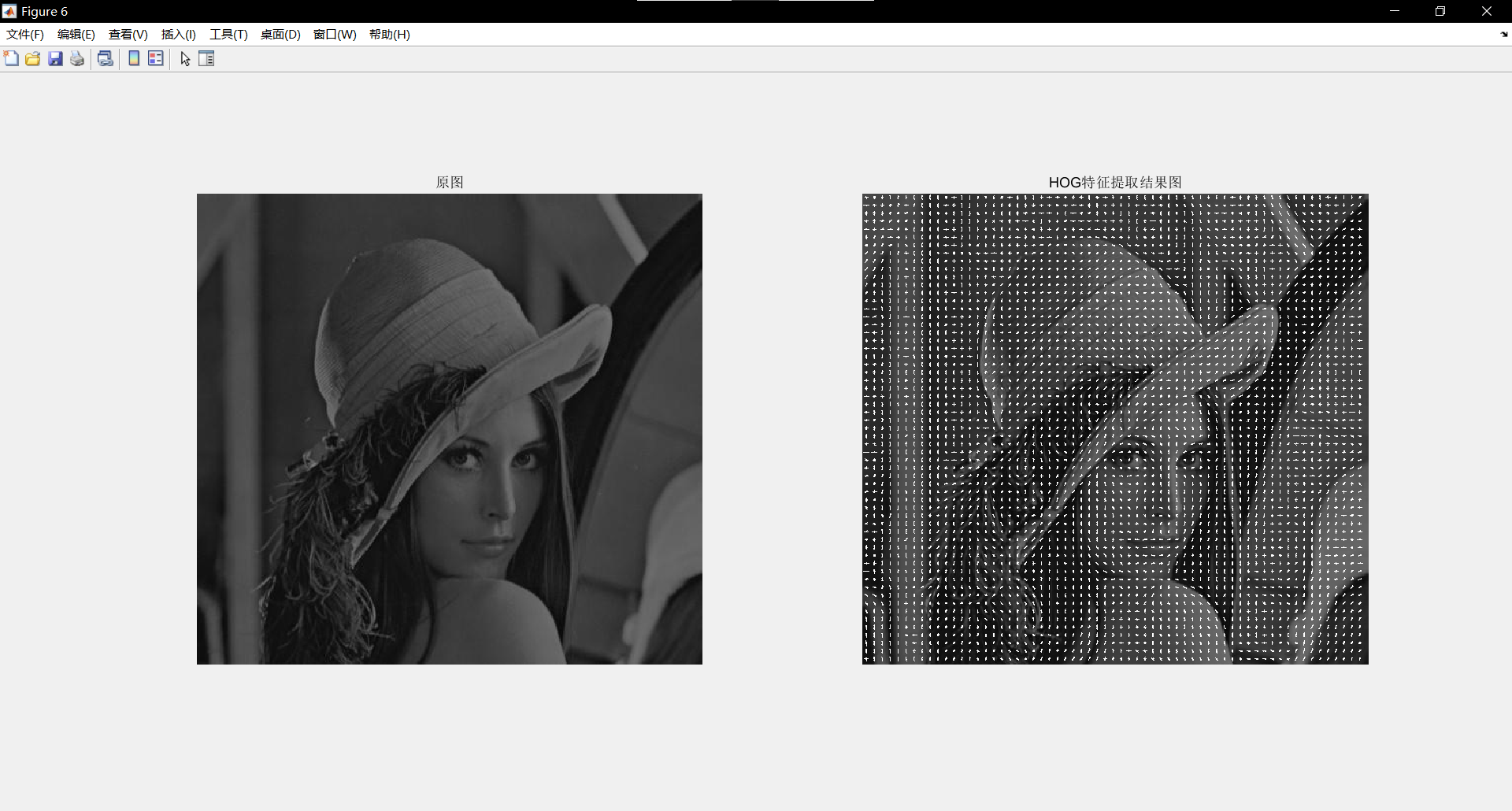
（4）Canny算子边缘检测原图、关键过程及结果；



（5）LBP纹理检测原图、关键过程及结果；



（6）HOG特征提取原图、关键过程及结果。



**六. 结果分析及体会**