

QIN, Yujie mail@qinyujie.net

Part 1:

## 1.Function Codes

```

function [template] = makeObjectTemplate(templateImg, gradKernelSize,
templateThresh)

%Turn the gray image into balck/white image
bwLevel = graythresh(templateImg);
bwTemplateImg = im2bw(templateImg, bwLevel);
%Calculate Gradients
[gradTemplate] = calcDirectionalGrad(bwTemplateImg, gradKernelSize);
%Get the edge of the shape from the image
edgeTemplateImg = im2bw(abs(gradTemplate), templateThresh);
%Generate Template
[lx, ly] = size(templateImg);
template = zeros(lx, ly, 2);
template(:, :, 1) = edgeTemplateImg;
template(:, :, 2) = gradTemplate;
end

function [fftMask]=makeFFTObjectMask(template, scale, angle, imgSize)

binObj = template(:, :, 1);
objGrad = template(:, :, 2);
%Change in Scale
binObj = imresize(binObj, scale, 'bicubic');
objGrad = imresize(objGrad, scale, 'bicubic');
%Rotation
degrees = mod((180*angle/pi),360);
binObj = imrotate(binObj, degrees, 'bicubic', 'loose');
i = sqrt(-1);
objGrad = exp(i .* angle) .* imrotate(objGrad, degrees, 'bicubic', 'loose');
%Normalisation
sumAbsGrad = sum(sum(abs(objGrad)));
objGrad = objGrad ./ sumAbsGrad;

%construct origin Object Mask
oriObjMask = binObj .* objGrad;
[oomx, oomy] = size(oriObjMask);
%Centre Object Mask in size imgSize
newObjMask = zeros(imgSize);
newObjMask(1:oomx, 1:oomy) = oriObjMask;
newObjMask = circshift(newObjMask, [-round(oomx ./ 2), -round(oomy ./ 2)]);
%Transfer into frequncy domain
fftMask = fft2(newObjMask);
end

function [houghSpace] = generalHough(gradImg, template, scaleSteps, scaleRange,
angleSteps, angleRange)

%Get image size
sizeImg = size(gradImg);
%Get image spectrum
fftImg = fft2(gradImg);
%Get iteration times
scaleIncr = (scaleRange(2)-scaleRange(1))/(scaleSteps-1);
angleIncr = (angleRange(2)-angleRange(1))/(angleSteps-1);
ist = 0;
iat = 0;

```

```

%Initialize Hg
houghSpace = zeros(sizeImg(1), sizeImg(2), scaleSteps, angleSteps);
%Iteration
for scale = scaleRange(1):scaleIncr:scaleRange(2)
    ist = ist + 1;
    iat = 0;
    for angel = angleRange(1):angleIncr:angleRange(2)
        iat = iat + 1;
        %Get the complex conjugate in spectrum with scale and angle
        [fftMask] = conj(makeFFTObjectMask(template, scale, angel, sizeImg));
        %Get result
        if (size(fftMask) - size(fftImg) == 0)
            fftMask = fftImg .* fftMask;
            Hsa = ifft2(fftMask);
        else
            %[size(fftMask) size(fftImg) scale angel]
        end
        houghSpace(:, :, ist, iat) = abs(Hsa);
    end
end
end
end

```

## 2.Main function and Grafiks

```

function part1

clear all;
close all;

imgTpl = imread('testTemplate.jpg');
imgTpl = rgb2gray(imgTpl);
[tplImgTpl] = makeObjectTemplate(imgTpl, 3, 0.5);

[imgDetect] = makeTestImg(imgTpl, 6 * pi / 10, 2, [1.7, 1.7]);
gradImgDetect = calcDirectionalGrad(imgDetect, 3);

%imgDetect = imread('testTemplate_2.jpg');
%imgDetect = rgb2gray(imgDetect);
%imgDetect = flipud(imgDetect);
%gradImgDetect = calcDirectionalGrad(imgDetect, 3);

%%plot
figure;hold on;
subplot(2, 3, 1);
imshow(imgTpl);
subplot(2, 3, 4);
imshow(imgDetect);
subplot(2, 3, 2);
imshow(tplImgTpl(:, :, 1), []);
subplot(2, 3, 3);
imshow(abs(tplImgTpl(:, :, 2)), []);
%paras
scaleSteps = 20;
scaleRange = [0.5, 2.5];
angleSteps = 20;
angleRange = [0, 2 .* pi];
[houghSpace] = generalHough(gradImgDetect, tplImgTpl, scaleSteps, scaleRange,
angleSteps, angleRange);
maxSA = max(max(houghSpace, [], 4), [], 3);
subplot(2, 3, 5);
imagesc(maxSA);

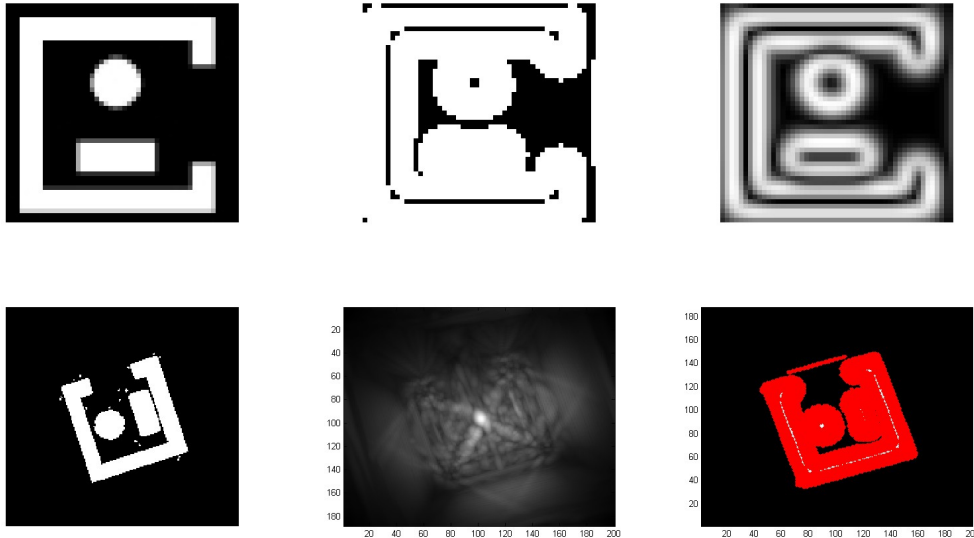
```

```

subplot(2, 3, 6);
% Find the linear index of the global maximum in Hough space
[ignore, maxInd] = max(houghSpace(:));
% Convert the index to a conventional 4D coordinate
objectList = convertLinearInd(size(houghSpace), maxInd);
plotHoughDetectionResult(imgDetect, tplImgTpl, objectList, scaleSteps,
scaleRange, angleSteps, angleRange);
hold off;

end

```



Part 2:

### 1. Function Codes

```

function findMoney()

close all;
clear all;

imgTpl = imread('moneyTemplate.jpg');
%imgTpl = rgb2gray(imgTpl);
[tplImgTpl] = makeObjectTemplate(imgTpl, 1, 0.5);

imgDetect = imread('poker.jpg');
%imgDetect = rgb2gray(imgDetect);
imgDetect = flipud(imgDetect);
gradImgDetect = calcDirectionalGrad(imgDetect, 1);

%%plot
figure;hold on;
subplot(2, 3, 1);
imshow(imgTpl);
subplot(2, 3, 4);
imshow(imgDetect);
subplot(2, 3, 2);
imshow(tplImgTpl(:,:,1),[]);
subplot(2, 3, 3);
imshow(abs(tplImgTpl(:,:,2)),[]);

```

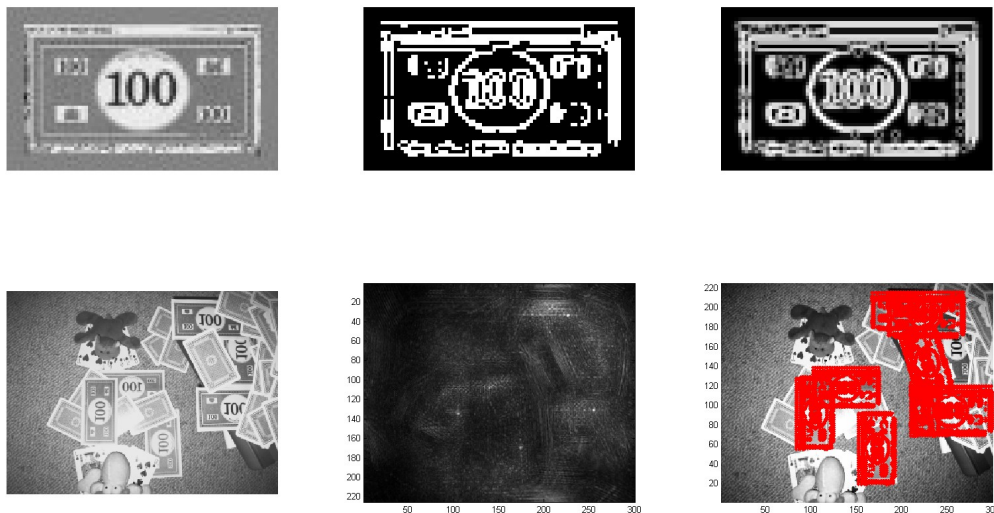
```

%paras
scaleSteps = 20;
scaleRange = [0.5, 2];
angleSteps = 20;
angleRange = [0, 2 .* pi];
[houghSpace] = generalHough(gradImgDetect, tplImgTpl, scaleSteps, scaleRange,
angleSteps, angleRange);
maxSA = max(max(houghSpace, [], 4), [], 3);
subplot(2, 3, 5);
imagesc(maxSA);

subplot(2, 3, 6);
FirstN = 8;
objectList = zeros(FirstN, 4);
sizeSA = size(maxSA);
subSA = zeros(scaleSteps, angleSteps);
maxInd = [0,0];
maxInd2 = [0,0];
for i = 1:FirstN
    maxValue = max(max(maxSA));
    [maxInd(1), maxInd(2)] = find(maxSA == maxValue, 1);
    subSA(:, :) = houghSpace(maxInd(1), maxInd(2), :, :);
    maxValue = max(max(subSA));
    [maxInd2(1), maxInd2(2)] = find(subSA == maxValue, 1);
    objectList(i, :) = [maxInd(1), maxInd(2), maxInd2(1), maxInd2(2)];
    deltaX = [max(1, maxInd(1) - 25), min(sizeSA(1), maxInd(1) + 25)];
    deltaY = [max(1, maxInd(2) - 25), min(sizeSA(2), maxInd(2) + 25)];
    maxSA(deltaX(1) : deltaX(2), deltaY(1) : deltaY(2)) = 0;
end
plotHoughDetectionResult(imgDetect, tplImgTpl, objectList, scaleSteps,
scaleRange, angleSteps, angleRange);
hold off;
end

```

2.Grafik



## Part 3:

## 1. Function Codes

```

function [template]=makeMarkerTemplate(templateEdgeLen)

edgeTemplateImg = zeros(templateEdgeLen);
midy = round(templateEdgeLen ./ 2);
edgeTemplateImg(:, midy) = ones(templateEdgeLen, 1);

gradTemplate = zeros(templateEdgeLen);
midx = round(templateEdgeLen ./ 2);
midy = round(templateEdgeLen ./ 2);
gradTemplate(1:midx - 1, midy) = ones(midx - 1, 1);
gradTemplate(midx + 1:end, midy) = -ones(midx - 1, 1);

template = zeros(templateEdgeLen, templateEdgeLen, 2);
template(:, :, 1) = edgeTemplateImg;
template(:, :, 2) = gradTemplate;

end

function findMarkers()

close all;
clear all;

[tplImgTpl] = makeMarkerTemplate(101);

imgDetect = imread('markers_contrast.jpg');
imgDetect = flipud(imgDetect);
gradImgDetect = calcDirectionalGrad(imgDetect, 1);

%%plot
figure;hold on;
subplot(2, 3, 3);
imshow(imgDetect);

%paras
scaleSteps = 10;
scaleRange = [0.205, 0.210];
angleSteps = 80;
angleRange = [0, 2 .* pi];
[houghSpace] = generalHough(gradImgDetect, tplImgTpl, scaleSteps, scaleRange,
angleSteps, angleRange);
maxSA = max(max(houghSpace, [], 4), [], 3);
subplot(2, 3, 6);
imagesc(maxSA);

subplot(2, 3, [1, 2, 4, 5]);
FirstN = 60;
objectList = zeros(FirstN, 4);
sizeSA = size(maxSA);
subSA = zeros(scaleSteps, angleSteps);
maxInd = [0,0];
maxInd2 = [0,0];
for i = 1:FirstN
    maxValue = max(max(maxSA));
    [maxInd(1), maxInd(2)] = find(maxSA == maxValue, 1);
    subSA(:, :) = houghSpace(maxInd(1), maxInd(2), :, :);
    maxValue = max(max(subSA));
    [maxInd2(1), maxInd2(2)] = find(subSA == maxValue, 1);

```

```

    objectList(i * 2 - 1, :) = [maxInd(1), maxInd(2), maxInd2(1), maxInd2(2)];
    deltaA = [mod(maxInd2(2) - angleSteps ./ 10, angleSteps) + 1,
mod(maxInd2(2) + angleSteps ./ 10, angleSteps) + 1];
    if (deltaA(1) < deltaA(2))
        subSA(:, deltaA(1) : deltaA(2)) = 0;
    else
        subSA(:, deltaA(1) : end) = 0;
        subSA(:, 1 : deltaA(2)) = 0;
    end
    deltaA = [mod(maxInd2(2) + angleSteps ./ 2 - angleSteps ./ 10, angleSteps)
+ 1, mod(maxInd2(2) + angleSteps ./ 2 + angleSteps ./ 10, angleSteps) + 1];
    if (deltaA(1) < deltaA(2))
        subSA(:, deltaA(1) : deltaA(2)) = 0;
    else
        subSA(:, deltaA(1) : end) = 0;
        subSA(:, 1 : deltaA(2)) = 0;
    end
    end
    maxValue = max(max(subSA));
    [maxInd(1), maxInd(2)] = find(subSA == maxValue, 1);
    objectList(i * 2, :) = [maxInd(1), maxInd(2), maxInd2(1), maxInd2(2)];

    deltaX = [max(1, maxInd(1) - 20), min(sizeSA(1), maxInd(1) + 20)];
    deltaY = [max(1, maxInd(2) - 20), min(sizeSA(2), maxInd(2) + 20)];
    maxSA(deltaX(1) : deltaX(2), deltaY(1) : deltaY(2)) = 0;
end
plotHoughDetectionResult(imgDetect, tplImgTpl, objectList, scaleSteps,
scaleRange, angleSteps, angleRange);
objectList
hold off;
end

```

## 2.Grafik

