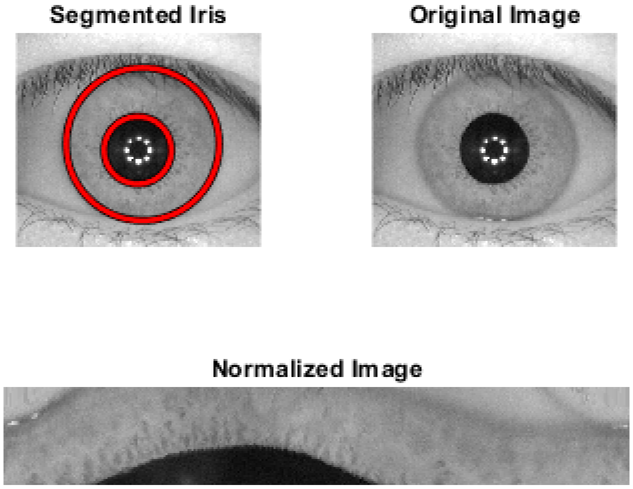
```matlab
%Stefana Rusu - Iris Segmentation Project
%CSCI446 - Advances in Biometrics
%this project uses the Circular Hough Transform method

close all;
clear all;
%read the grayscale image from the path
I = imread('iris.png');

%convert image to binary
BW = im2bw(I);

%find the edges using the Canny operator
Image = edge(BW, 'canny');

%find the center coordinates and radius for the pupillary boundary
[pupilCenterX, pupilCenterY, pupilRadius] =
 findPupillaryBoundary(Image);

%find the center coordinates and radius for the limbus boundary
[limbusCenterX, limbusCenterY, limbusRadius] =
 findLimbusBoundary(Image);

%display the original image
imshow(I);
subplot(2,2,1);
imshow(I);
title('Segmented Iris');
hold on;

%draw circles on the original image
c1 = viscircles([pupilCenterX, pupilCenterY], pupilRadius);
c2 = viscircles([limbusCenterX, limbusCenterY], limbusRadius);
subplot(2,2,2);
imshow(I);
title('Original Image');
hold on;

% decide the size of the rectangular normalized image (50x320)
m = 50;
n = 320;

% initialize the normalized image
Im = zeros(m,n);
Im = uint8(Im); %convert it to uint8

%loop through the pixels of the original image
for j = 1:n %columns
    theta = (2*pi/n)*j;
    limbusX = limbusRadius * cos(theta);
    limbusY = limbusRadius * sin(theta);
    pupilX = pupilRadius * cos(theta);
    pupilY = pupilRadius * sin(theta);

    xMinusH = limbusX - pupilX;
    yMinusK = limbusY - pupilY;

    x = xMinusH/m;
    y = yMinusK/m;

    for i = 1:m %rows
        xcoord = limbusX - (x * i) + limbusCenterX;
        ycoord = limbusY - (y * i) + limbusCenterY;
        xcoord = uint8(xcoord);
        ycoord = uint8(ycoord);
        Im(i,j) = I(xcoord, ycoord); %create the normalized image
    end
end

subplot(2,1,2);
imshow(Im);
title('Normalized Image');
hold on;

%Below are the functions to find the circle coordinates and radii of
 the two boundaries
%using trial and error, the radius range was found to be 37-45 pixels
 for
%the pupil and 97-105 pixels for the limbus

%Terms:
%'two stage' = the method used in two-stage circular Hough transform
%'object polarity' = indicates whether the circular objects are
 brighter or darker than the background
%'sensitivity' = sensitivity for the circular Hough transform
 accumulator array

function [pCenterX, pCenterY, pRadius] =
 findPupillaryBoundary(grayscaleImage)
[center, radius] = imfindcircles(grayscaleImage,[37
 45],'ObjectPolarity','dark','Sensitivity',1);
pCenterX = center(1,1);
pCenterY = center(1,2);
pRadius  = radius(1);
end

function [lCenterX, lCenterY, lRadius] =
 findLimbusBoundary(grayscaleImage)
[center, radius] = imfindcircles(grayscaleImage,[97
 105], 'ObjectPolarity','dark','Sensitivity',1,'Method','twostage');
lCenterX = center(1,1);
lCenterY = center(1,2);
lRadius  = radius(1);
end
```



Segmented Iris

Original Image

Normalized Image

Published with MATLAB® R2018a