#include "stdafx.h"  
#include <stdio.h>  
#include <iostream>  
#include <math.h>  
#include <opencv2\opencv.hpp>  
  
#define PI 3.14159265  
  
using namespace cv;  
using namespace std;  
  
int lowthres, highthres;  
Mat img = imread("C:\\lena.jpg");  
Mat edgeimg(img.rows, img.cols, CV\_8UC3);  
  
void houghtransform(Mat edgeimg)  
{  
 //Hough transform on edgeimg  
 int i, j, a[1000][72] = { 0 }; //a[1000][72] is the (r,theta) line array  
 int r, theta; //r varies from 0-999,theta from 0-355 degree  
 int nooflinethres = 100;  
 int valuethres = 2, value;  
 for (i = 0; i < edgeimg.rows; i++)  
 {  
 for (j = 0; j < edgeimg.cols; j++)  
 {  
 if (edgeimg.at<uchar>(i, j) != 0)  
 {  
 for (theta = 0; theta <= 355; theta += 5)  
 {  
 r = floor(i\*cos(theta\*PI / 180) + j\*sin(theta\*PI / 180));  
 if (r >= 0 && r <= 999)  
 a[r][theta / 5]++;  
 }  
 }  
 }  
 }  
 for (r = 0; r < 1000; r++)  
 {  
 for (theta = 0; theta <= 355; theta += 5)  
 {  
 if (a[r][theta / 5] >= nooflinethres)  
 {  
 for (i = 0; i < edgeimg.rows; i++)  
 {  
 for (j = 0; j < edgeimg.cols; j++)  
 {  
 value = floor(i\*cos(theta\*PI / 180) + j\*sin(theta\*PI / 180));  
 if (value >= r - valuethres&&value <= r + valuethres)  
 edgeimg.at<uchar>(i, j) = 255;  
 }  
 }  
 }  
 }  
 }  
 imshow("Hough", edgeimg);  
}  
  
void edgedetectlow(int lowthres, void \*x)  
{  
 Canny(img, edgeimg, lowthres, highthres, 3);  
 imshow("Edge map", edgeimg);  
 houghtransform(edgeimg);  
}  
  
void edgedetecthigh(int highthres, void \*x)  
{  
 Canny(img, edgeimg, lowthres, highthres, 3);  
 imshow("Edge map", edgeimg);  
 houghtransform(edgeimg);  
}  
  
int main()  
{  
 lowthres = highthres = 0;  
 imshow("Original", img);  
 namedWindow("Edge map", WINDOW\_NORMAL);  
 createTrackbar("Lower", "Edge map", &lowthres, 255, edgedetectlow);  
 createTrackbar("Higher", "Edge map", &highthres, 255, edgedetecthigh);  
 waitKey(0);  
 return 0;  
}