#include "stdafx.h"

#include<opencv2\core\core.hpp>

#include<opencv2\highgui\highgui.hpp>

#include<opencv2\imgproc\imgproc.hpp>

#include <iostream>

using namespace std;

using namespace cv;

int main()

{

VideoCapture vid(1);

Mat image;

int threslow=50, threshigh=100;

namedWindow("Output", CV\_WINDOW\_AUTOSIZE);

createTrackbar("low", "Output", &threslow, 500);

createTrackbar("high", "Output", &threshigh, 500);

while (1)

{

vid.read(image);

//make it grey using histogram

Mat grey(image.rows, image.cols, CV\_8UC1);

cvtColor(image, grey, CV\_BGR2GRAY);

//find canny edges

Mat cannyedge(image.rows, image.cols, CV\_8UC1);

Mat cannyedge2(image.rows, image.cols, CV\_8UC1);

Canny(grey, cannyedge, threslow, threshigh, 3);

//find hough lines

for (int i = 0; i < image.rows; ++i)

{

for (int j = 0; j < image.cols; ++j)

{

if (i>image.rows/2)

cannyedge2.at<uchar>(i, j) = cannyedge.at<uchar>(i, j);

else

cannyedge2.at<uchar>(i, j) = 0;

}

}

vector<Vec2f> lines2; //lines is a vector of r rows and 2 columns storing float values

int thres2 = 50;

Mat hline2(image.rows, image.cols, CV\_8UC1);

HoughLines(cannyedge2, lines2, 1, CV\_PI / 180, thres2);

int count = 0;

double sum = 0.0;

for (int i = 0; i < lines2.size(); ++i)

{

double rho = lines2[i][0];

double theta = lines2[i][1];

sum += theta;

count++;

Point pt1, pt2;

double a = cos(theta), b = sin(theta);

double x0 = a\*rho, y0 = b\*rho;

pt1.x = cvRound(x0 + 1500 \* (-b));

pt1.y = cvRound(y0 + 1500 \* (a));

pt2.x = cvRound(x0 - 1500 \* (-b));

pt2.y = cvRound(y0 - 1500 \* (a));

line(hline2, pt1, pt2, Scalar(0, 0, 255), 1, CV\_AA);

}

if (count == 0)

count = 1;

double final = sum / count;

if (final > CV\_PI / 2.0 + 5 \* CV\_PI / 180.0)

cout << "a" << endl;

else if (final < CV\_PI / 2.0 - 5 \* CV\_PI / 180.0)

cout << "d" << endl;

else cout << "w"<<endl;

imshow("Output", hline2);

if (waitKey(30) == 27)

break;

}

waitKey(0);

return 0;

}