

LEARN OPENCV BY EXAMPLES

OpenCV simplified for beginners by the use of examples. Learn OpenCV with basic implementation of different algorithms. Please visit LearnOpenCV.com for newer articles.

Home	For Beginners	Table of Contents	Keywords	Resources
------	---------------	-------------------	----------	-----------

Line Detection by Hough Line Transform

void **HoughLines**(InputArray image, OutputArray lines, double rho, double theta, int threshold, double srn=0, double stn=0)

Parameters:

- **image** – 8-bit, single-channel binary source image (use edge detectors)
- **lines** – Output vector of lines. Each line is represented by a two-element vector (ρ, θ) . ρ is the distance from the coordinate origin $(0,0)$ (top-left corner of the image). θ is the line rotation angle in radians ($0 \leq \theta < \pi$).
- **rho** – Distance resolution of the accumulator in pixels.
- **theta** – Angle resolution of the accumulator in radians.
- **threshold** – Accumulator threshold parameter. Only those lines are returned that get enough votes ($> \text{threshold}$).
- **srn** – For the multi-scale Hough transform, it is a divisor for the distance resolution ρ . The coarse accumulator distance resolution is ρ and the accurate accumulator resolution is ρ/srn . If both $\text{srn}=0$ and $\text{stn}=0$, the classical Hough transform is used. Otherwise, both these parameters should be positive.
- **stn** – For the multi-scale Hough transform, it is a divisor for the distance resolution θ .

A good example for Hough Line Transform is provided in [OpenCV Documentation](#).

Steps:

1. Load image and convert to gray-scale.
2. Apply the Hough Transform to find the lines. ([HoughLines](#))
3. Draw the detected lines. ([line](#))
4. Show the result

Functions:

[HoughLines](#), [line](#), [Canny](#), [cvtColor](#), [imshow](#), [imread](#), [namedWindow](#), [waitKey](#).

Example:

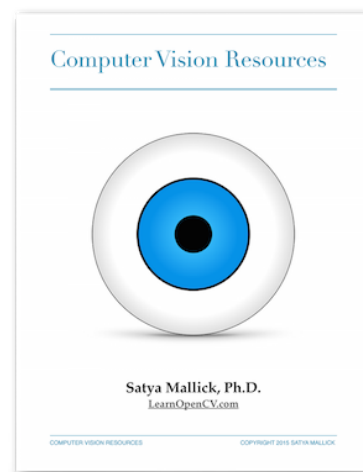
```

1  #include "opencv2/highgui/highgui.hpp"
2  #include "opencv2/imgproc/imgproc.hpp"
3  #include <iostream>
4
5  using namespace cv;
6  using namespace std;
7
8  int main()
9  {
10     Mat src = imread("building.jpg", 0);
11
12     Mat dst, cdst;
13     Canny(src, dst, 50, 200, 3);
14     cvtColor(dst, cdst, CV_GRAY2BGR);
15
16     vector<Vec2f> lines;
17     // detect lines
18     HoughLines(dst, lines, 1, CV_PI/180, 150, 0, 0 );
19
20     // draw lines
21     for( size_t i = 0; i < lines.size(); i++ )

```

COMPUTER VISION RESOURCES

To learn more about OpenCV books, libraries, and web APIs download our free resource guide.



SEARCH CONTENTS OF THIS BLOG

POPULAR POSTS

- 1 Find Contour
- 2 Basic drawing examples
- 3 Line Detection by Hough Line Transform
- 4 Face Detection using Haar-Cascade Classifier
- 5 Perspective Transform
- 6 OpenCV example to convert RGB to gray / other color spaces
- 7 Creating matrix in different ways
- 8 Canny Edge Detection
- 9 Bitwise AND, OR, XOR and NOT

```

22     {
23         float rho = lines[i][0], theta = lines[i][1];
24         Point pt1, pt2;
25         double a = cos(theta), b = sin(theta);
26         double x0 = a*rho, y0 = b*rho;
27         pt1.x = cvRound(x0 + 1000*(-b));
28         pt1.y = cvRound(y0 + 1000*(a));
29         pt2.x = cvRound(x0 - 1000*(-b));
30         pt2.y = cvRound(y0 - 1000*(a));
31         line( cdst, pt1, pt2, Scalar(0,0,255), 3, CV_AA);
32     }
33
34     imshow("source", src);
35     imshow("detected lines", cdst);
36
37     waitKey();
38     return 0;
39 }

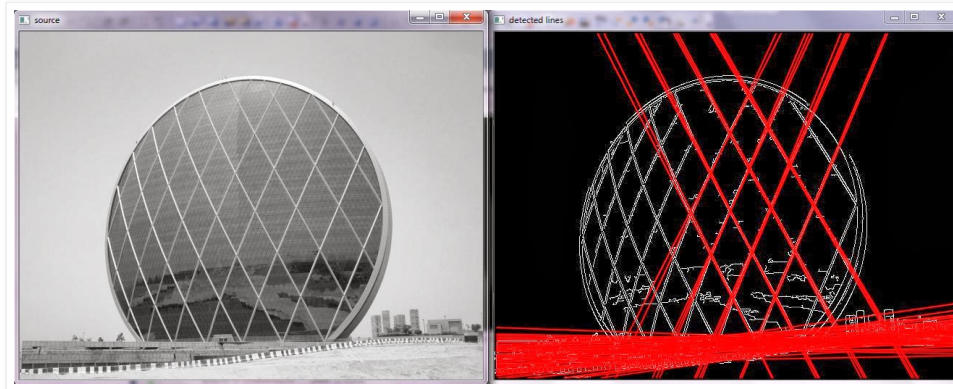
```

10 Kalman Filter Implementation (Tracking mouse position)

CATEGORIES

- [Accessory](#)
- [Applications](#)
- [Basics](#)
- [Edge Detection](#)
- [Feature Extraction](#)
- [Filter](#)
- [Miscellaneous](#)
- [Morphological Operation](#)

Result:



Extra Stuffs:

- What if you need to select some lines on the basis of prior knowledge of range of angles?

So, you know the range of angles in which your lines may present. Now, what you need to do is to add a conditional statement to filter out the lines detected in that angle range. For example,

if you want to detect vertical lines, use the following conditional statement **after line 23** of the above example.

```

1  if( theta>CV_PI/180*170 || theta<CV_PI/180*10) ?
2      { Point pt1, pt2;
3          .....
4          line( cdst, pt1, pt2, Scalar(0,0,255), 3, CV_AA);
5      }

```

if you want to detect horizontal lines, use

```

1  if( theta>CV_PI/180*80 && theta<CV_PI/180*100) ?
2      { Point pt1, pt2;
3          .....
4          line( cdst, pt1, pt2, Scalar(0,0,255), 3, CV_AA);
5      }

```

Labels: [Edge Detection](#)

21 comments:



Jeremy Borg March 11, 2014 at 12:06 AM

If i know that the lines im trying to detect will be in a specific range, e.g between 80 and 110 degrees, can I limit the range that houghLines searches through?

[Reply](#)

[Replies](#)

Kanha [March 11, 2014 at 5:54 AM](#)

Excellent question Borg.

I have updated the post. Please have a look.

All you need to do is restrict your angle range. i.e. choose the lines those fall in your angle range. This can be done using a conditional statement after line detection.



Zams Z [September 18, 2014 at 8:45 PM](#)

Hi Kanha,

Hope you can teach on the specific degree. What I had understand is totally weird

?

1
2
3
4
5

if(theta>CV_PI/180*170 || theta 170 degree || means OR in C++ .. and still confusing..
please help to details out then.. appreciate on yr response
zamani

Kanha [September 19, 2014 at 9:40 AM](#)

Here "theta" is in radian. So we have to convert degree to radian.

Here 170 degree = (CV_PI/180)*170 radian.

We may not get perfect horizontal lines. So we have to put some threshold. Perfect horizontal line occurs when theta = CV_PI/2 = 90 degree. So we use the threshold from 80~100 degrees to detect horizontal lines.

For vertical lines, the angle is 0 degree = 180 degree.

So we can write that if (angle<10 degree OR angle>170 degree), then it is a vertical line.

Hope this helps.



Zams Z [September 19, 2014 at 2:01 PM](#)

Hi Kanha,

Thanks for the details up and I had tested again and yes it worked as per define above.
TQSM!!

Just another question related to Hough Transform but for Probabilistic Hough Line Transform. In this function, the HT Prob

HoughLinesP(dst, lines, 1, CV_PI/180, 50, 50, 10);

with the arguments:

- dst: Output of the edge detector. It should be a grayscale image (although in fact it is a binary one)
- lines: A vector that will store the parameters (x_{start}, y_{start}, x_{end}, y_{end}) of the detected lines
- rho : The resolution of the parameter r in pixels. We use 1 pixel.
- theta: The resolution of the parameter \theta in radians. We use 1 degree (CV_PI/180)
- threshold: The minimum number of intersections to "detect" a line
- minLinLength: The minimum number of points that can form a line. Lines with less than this number of points are disregarded.
- maxLineGap: The maximum gap between two points to be considered in the same line.

Here can the rho and theta been employed again to get the selective degree? The lines now are vectors X start, Y start, X end and Y end which is different.

can I limit the range that houghLines probabilistic searches through? If yes, should I declare again using x start, y start, x end and y end?

Thanks
Zamani

Reply

Anonymous April 15, 2014 at 9:44 PM

Hi Kanha

if replace vector lines; by vector lines; then what are define rho and theta?

[Reply](#)

[Replies](#)

Anonymous April 15, 2014 at 9:45 PM

Vec2f by Vec4i

Kanha April 16, 2014 at 9:50 AM

These two datatypes are defined as:

```
typedef Vec Vec2f;
typedef Vec Vec4i;
```

So, vec2f can contain two float values. Similarly, vec4i can have four integer values.

However, we need two float values to represent a line, i.e. slope and distance from origin.

Therefore, vec2f is used.

[Reply](#)

Anonymous April 16, 2014 at 1:37 PM

Thanks for you reply, Kanha.

In my case, i used Vec4i and i want to detect vertical lines, but i don't know how to defined rho and theta.

This is my code

```
vector< Vec4i > lines; //don't space between < and Vec4i
int houghThreshold = 70;
if(imgGRAY.cols*imgGRAY.rows < 400*400)
    houghThreshold = 100;
```

```
cv::HoughLinesP(imgCanny, lines, 1, CV_PI/180, houghThreshold, 10,10);
```

```
while(lines.size() > MAX_NUM_LINES)
{
    lines.clear();
    houghThreshold += 10;
    cv::HoughLinesP(imgCanny, lines, 1, CV_PI/180, houghThreshold, 10, 10);
}
for(size_t i=0; i<lines.size(); i++)
{
    Point pt1, pt2;
    //defined rho and theta to detect vertical lines ???
    pt1.x = lines[i][0];
    pt1.y = lines[i][1];
    pt2.x = lines[i][2];
    pt2.y = lines[i][3];
    line(outputImg, pt1, pt2, CV_RGB(0,0,0), 2);
    aux.clear();
    aux.push_back(pt1);
    aux.push_back(pt2);
    lineSegments.push_back(aux);
}
```

Could you help me with this?

[Reply](#)



Vivek Goel October 18, 2014 at 8:42 PM

thanks allot brother, thanks allot!! :)

[Reply](#)

Anonymous February 27, 2015 at 4:13 PM

can you explain about those line :

```
pt1.x = cvRound(x0 + 1000*(-b));
pt1.y = cvRound(y0 + 1000*(a));
```

```
pt2.x = cvRound(x0 - 1000*(-b));
pt2.y = cvRound(y0 - 1000*(a));
Why a and b must be multiplied by 1000?
```

[Reply](#)**Francisco Alves** March 12, 2015 at 10:20 PM

Very good! It was very usefull to me. Thank you

[Reply](#)**Anonymous** June 4, 2015 at 11:39 PM

Is there any way to limit minimal length of a line?
Thanks for great article

[Reply](#)**mustafa göl** June 9, 2015 at 2:21 AM

Hi Kanha, your tutorial is very helpful. I use HoughLinesP and I want to merge or average neighbour lines. Do you have any suggestion? Thanks

[Reply](#)**ankit ramani** June 9, 2015 at 6:59 PM

Hi Kanha,

how can we measure the distance between two lines or how to get the desired line?
in my case i want to detect the tube and width of tube is 46 pixel, so i just want to draw line which has distance 46 between them and rest line should be neglect.

Any suggestion.

Thank you for tutorial.

[Reply](#)**Đào Quân** December 7, 2015 at 9:58 AM

dịch vụ thành lập doanh nghiệp công ty trọn gói
dịch vụ thành lập doanh nghiệp công ty tại thanh xuân
dịch vụ thành lập doanh nghiệp công ty tại hà đông
dịch vụ thành lập doanh nghiệp công ty tại long biên
dịch vụ thành lập doanh nghiệp công ty tại cầu giấy
dịch vụ thành lập doanh nghiệp công ty tại bắc ninh
dịch vụ thành lập doanh nghiệp công ty tại quận 3 tphcm
dịch vụ thành lập doanh nghiệp công ty tại quận đồ ng đa
dịch vụ thành lập doanh nghiệp công ty tại quận thủ đức
dịch vụ thành lập doanh nghiệp công ty tại huyện đông anh
dịch vụ thành lập doanh nghiệp công ty tại huyện nhà bè
dịch vụ thành lập doanh nghiệp công ty tại huyện hoà đức
dịch vụ thành lập doanh nghiệp công ty tại bình dương
dịch vụ thành lập doanh nghiệp công ty tại hưng yên

[Reply](#)**Anonymous** August 3, 2016 at 4:17 PM

Hello there i am new in opencv as well as in image processing
i want to detect the crack in biscuit so i used the haugh transform as above example so i can crack this.

Please tell me how to do this.

Here i am processing

1->take an image

2->adaptive segment

3->haugh line transform using above code.

but line is not detected.

red window displays.

[Reply](#)**Hari Prasad** January 10, 2018 at 1:10 PM

what is the role of threshold here? canyou explain

[Reply](#)



kritika mytectra September 18, 2018 at 4:25 PM

Python is a general-purpose interpreted, interactive, object-oriented and high-level programming language. Currently Python is the most popular Language in IT.

[python training in bangalore](#)

[aws training in bangalore](#)

[artificial intelligence training in bangalore](#)

[data science training in bangalore](#)

[machine learning training in bangalore](#)

[hadoop training in bangalore](#)

[devops training in bangalore](#)

[Reply](#)



kritika mytectra September 18, 2018 at 4:27 PM

[corporate training companies](#)

[corporate training companies in mumbai](#)

[corporate training companies in pune](#)

[corporate training companies in delhi](#)

[corporate training companies in chennai](#)

[corporate training companies in hyderabad](#)

[corporate training companies in bangalore](#)

[Reply](#)



kritika mytectra December 20, 2018 at 5:42 PM

Currently [Python](#) is the most popular Language in IT. [Python](#) adopted as a language of choice for almost all the domain in IT including Web Development, Cloud Computing (AWS, OpenStack, VMware, Google Cloud, etc..),[Read More](#)

[Reply](#)

Enter your comment...



Comment as:

stephtan2007 ▼

[Sign out](#)

[Publish](#)

[Preview](#)

☐ [Notify me](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

If you liked the articles on this blog, sign up for our weekly newsletter and receive a free **"Computer Vision Resources"** guide.

[Subscribe Now](#)