

JackyChuan

一个业余硅元素长方体空间移动师的博客。

目录视图 摘要视图 RSS 订阅 发布Chat

个人资料



看图算命

访问： 13199次
积分： 211
等级： **BLOG > 2**
排名： 千里之外

原创： 8篇 转载： 0篇
译文： 0篇 评论： 0条

文章搜索

文章分类

opencv (7)
视觉基础知识 (1)

文章存档

2016年12月 (2)
2016年11月 (2)
2016年10月 (4)

阅读排行

Opencv学习笔记-----霍夫 (5446)
Opencv学习笔记-----求取 (3523)
Opencv学习笔记-----Filel (1333)
Opencv学习笔记-----PC (1229)
Opencv学习笔记 ---open (799)
基础知识填坑---矢量的叉 (292)
Opencv学习笔记-----理解 (262)
Opencv学习笔记-----图像 (209)

评论排行

Opencv学习笔记 ---open (0)
Opencv学习笔记-----Filel (0)
Opencv学习笔记-----霍夫 (0)
Opencv学习笔记-----理解 (0)
Opencv学习笔记-----PC (0)

异步赠书：10月Python畅销书升级 【线路图】人工智能到底学什么？！ 程序员9月书讯 每周荐书（京东篇）：618取胜之道、质量保障、技术解密)

Opencv学习笔记-----霍夫变换直线检测及原理理解

标签： opencv 霍夫变换 直线检测 2016-10-27 12:33 5462人阅读 评论

分类：
opencv (6)

版权声明：本文为博主原创文章，未经博主允许不得转载。

霍夫变换(Hough Transform)是图像处理中的一种特征提取技术，它通过一种投票方式识别具有特定形状的物体。该过程在一个参数空间中通过计算累计结果的局部最大值得到一个符合该特定形状的集合作为霍夫变换结果。霍夫变换于1962年由Paul Hough 首次提出[53]，后于1972年由Richard Duda和Peter Hart推广使用[54]，经典霍夫变换用来检测图像中的直线，后来霍夫变换扩展到任意形状物体的识别，多为圆和椭圆。
经过几天的学习，发现各位大牛的理解方式之前都是有一些区别的，但是核心的思想没有变化，因此记录一下自己对霍夫变换直线检测的认识。

一、原理介绍：

- 1、对于直角坐标系中的任意一点A(x0,y0)，经过点A的直线满足Y0=k*X0+b.(k是斜率，b是截距)
- 2、那么在X-Y平面过点A(x0,y0)的直线簇可以用Y0=k*X0+b表示，但对于垂直于X轴的直线斜率是无穷大的则无法表示。因此将直角坐标系转换到极坐标系就能解决该特殊情况。
- 3、在极坐标系中表示直线的方程为ρ=xCosθ+ySinθ（ρ为原点到直线的距离）,如图所示：



人工植发



Opencv学习笔记-----图像 (0)

基础知识填坑---矢量的叉 (0)

Opencv学习笔记-----求取 (0)

推荐文章

* CSDN邀请你来GitChat赚钱啦！

* 行为驱动开发 (BDD) 你准备好了吗？

* 如何更加安全、高效地利用开源项目？

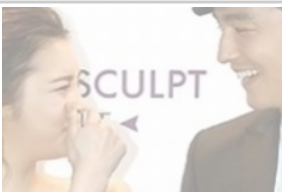
* 程序员业余时间修炼指南

* DevOps 在公司项目中的实践落地

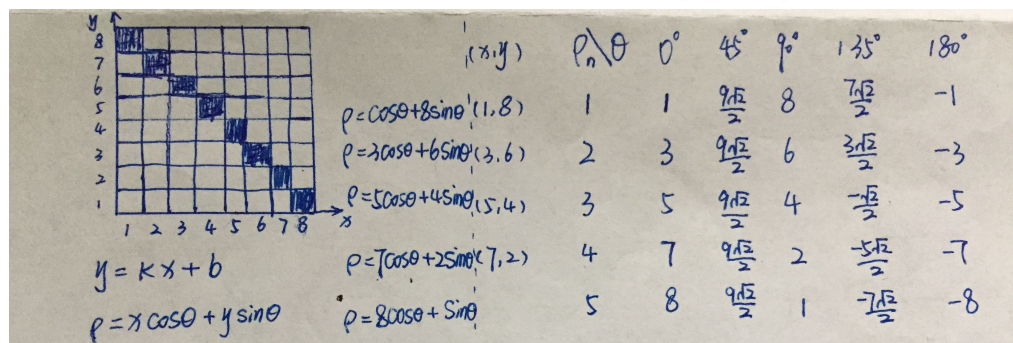
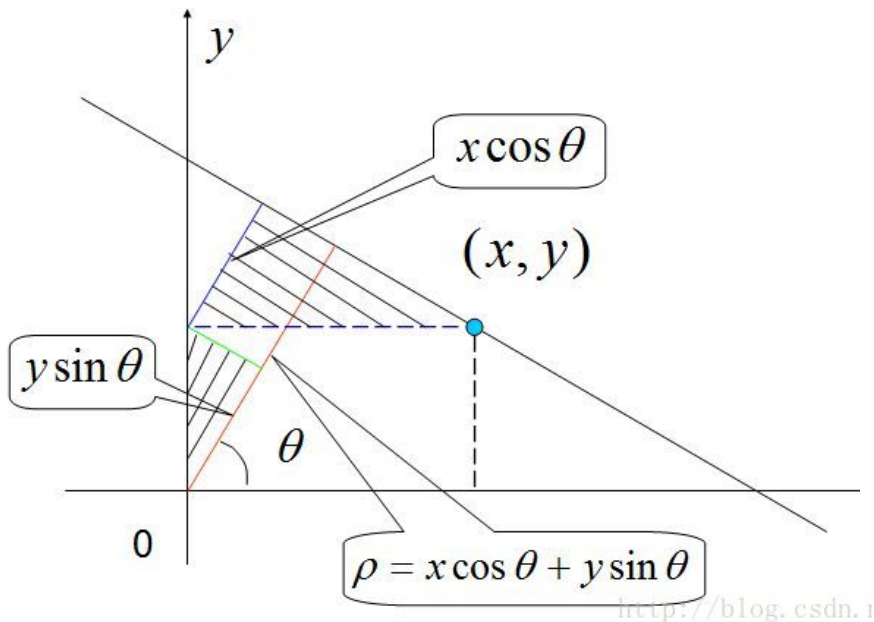
* Jenkins + Django 完整实战，细化到每一步操作



液态硬盘



去除口臭的方法



4、如上图，假定在一个8*8的平面像素中有一条直线，并且从左上角(1,8)像素点开始分别计算 θ 为 0° 、 45° 、 90° 、 135° 、 180° 时的 ρ ，图中可以看出 ρ 分别为1、 $(9\sqrt{2})/2$ 、8、 $(7\sqrt{2})/2$ 、-1，并给这5个值分别记一票，同理计算像素点(3,6)点 θ 为 0° 、 45° 、 90° 、 135° 、 180° 时的 ρ ，再给计算出来的5个 ρ 值分别记一票，此时就会发现 $\rho = (9\sqrt{2})/2$ 的这个值已经记了两票了，以此类推，遍历完整个8*8的像素空间的时候 $\rho = (9\sqrt{2})/2$ 就记了5票，别的 ρ 值的票数均小于5票，所以得到该直线在这个8*8的像素坐标中的极坐标方程为 $(9\sqrt{2})/2 = x \cdot \cos 45^\circ + y \cdot \sin 45^\circ$ ，到此该直线方程就求出来了。(PS：但实际中 θ 的取值不会跨度这么大，一般是 $\pi/180$)。

二、Opencv实现直线检测：

1、Opencv1.0版本：

```
[cpp]
01. #include<cv.h>
02. #include<highgui.h>
03.
04. int main()
05. {
06.     IplImage* pImgSrc = NULL;    //源图像
07.     IplImage* pImg8u = NULL;    //灰度图
08.     IplImage* pImgCanny = NULL; //边缘检测后的图
09.     IplImage* pImgDst = NULL;   //在图像上画上线检测到的直线
10.     CvSeq* lines = NULL;
11.     CvMemStorage* storage = NULL;
12.
13.     /*边缘检测*/
14.     pImgSrc = cvLoadImage("../res/street.jpg", 1);
15.     pImg8u = cvCreateImage(cvGetSize(pImgSrc), IPL_DEPTH_8U, 1);
16.     pImgCanny = cvCreateImage(cvGetSize(pImgSrc), IPL_DEPTH_8U, 1);
17.     pImgDst = cvCreateImage(cvGetSize(pImgSrc), IPL_DEPTH_8U, 1);
18.     cvCvtColor(pImgSrc, pImg8u, CV_BGR2GRAY);
19.     cvCanny(pImg8u, pImgCanny, 20, 200, 3);
20.
21.     /*检测直线*/
22.     storage = cvCreateMemStorage(0);
23.     lines = cvHoughLines2(pImgCanny, storage, CV_HOUGH_LINES,
```



```
24.     pImgDst = cvCreateImage(cvGetSize(pImgSrc), IPL_DEPTH_8U, 3);
25.     cvCvtColor(pImg8u, pImgDst, CV_GRAY2BGR);
26.
27.     /*在pImgDst上画出检测到的直线*/
28.     for (int i = 0; i < lines->total; i++)
29.     {
30.         CvPoint* line = (CvPoint*)cvGetSeqElem(lines, i);
31.         cvLine(pImgDst, line[0], line[1], CV_RGB(255, 0, 0), 3, 8);
32.     }
33.
34.     cvNamedWindow("src", 1);
35.     cvNamedWindow("canny", 1);
36.     cvNamedWindow("hough", 1);
37.     cvShowImage("src", pImgSrc);
38.     cvShowImage("canny", pImgCanny);
39.     cvShowImage("hough", pImgDst);
40.
41.     cvWaitKey(0);
42.
43.     cvReleaseImage(&pImgSrc);
44.     cvReleaseImage(&pImg8u);
45.     cvReleaseImage(&pImgCanny);
46.     cvReleaseImage(&pImgDst);
47.     cvReleaseMemStorage(&storage);
48.
49.     return 0;
50. }
```



液态硬盘



去除口臭的方法



2、Opencv2.4.9版本：

```
[cpp]
01. #include<opencv2\imgproc\imgproc.hpp>
02. #include<opencv2\opencv.hpp>
03. #include<opencv2\highgui\highgui.hpp>
04. using namespace std;
05. using namespace cv;
06.
07. int main()
08. {
09.     Mat Image = imread("../res//street.jpg", 0);
10.     Mat CannyImg;
11.     Canny(Image, CannyImg, 140, 250, 3);
12.     imshow("CannyImg", CannyImg);
13.
14.     Mat DstImg;
15.     cvtColor(Image, DstImg, CV_GRAY2BGR);
16.
17.     vector<Vec4i> Lines;
18.     HoughLinesP(CannyImg, Lines, 1, CV_PI / 360, 170,30,15);
19.     for (size_t i = 0; i < Lines.size(); i++)
20.     {
21.         line(DstImg, Point(Lines[i][0], Lines[i][1]), Point(Lines[i][2], Lines[i]
22. [3]), Scalar(0, 0, 255), 2, 8);
23.     }
24.     imshow("HoughLines_Detect", DstImg);
25.     imwrite("../res//HoughLines_Detect.jpg", DstImg);
26.     waitKey(0);
27.     return 0;
28. }
```

3、效果图：



人工植发

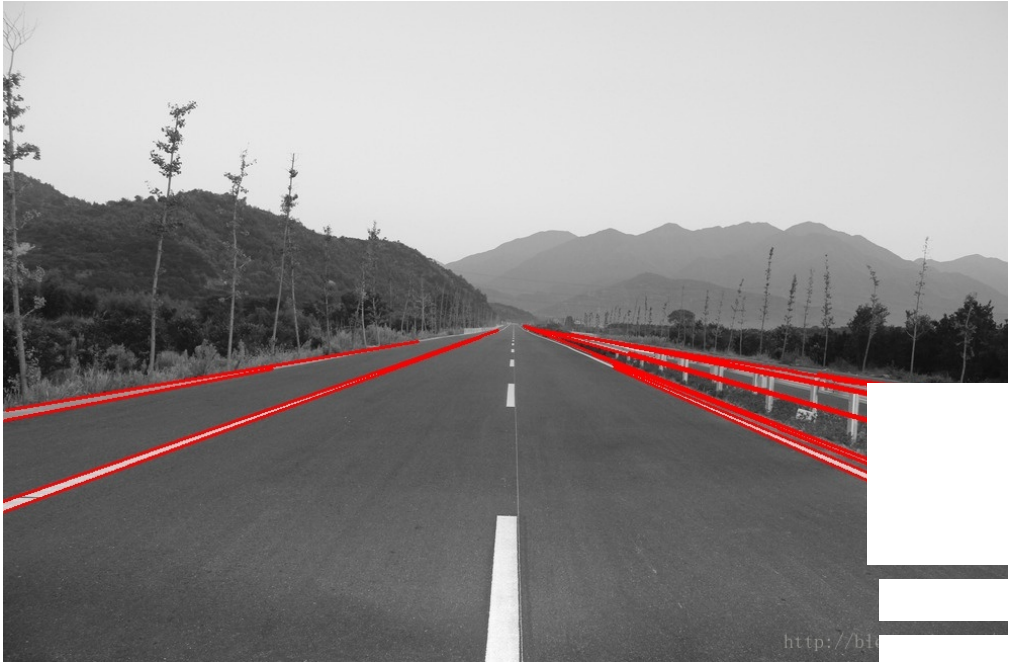




液态硬盘



去除口臭的方法



三、推荐一个国外公路直线检测的大神的个人主页（有很多资源和源码）：
Mohamed Aly：<http://www.vision.caltech.edu/malaa/research/>
他研究了公路上的直线（斑马线）等的检测。


顶 踩
4 0

上一篇 Opencv学习笔记-----FilePath的相对路径设置方法
下一篇 Opencv学习笔记-----理解Mat


相关文章推荐

- opencv直线检测直线提取算法与总结
- 机器学习之数学基础系列--AI100
- 【OpenCV】利用霍夫变换进行直线检测
- 使用Keras快速构造自己的深度学习模型--谢梁
- 霍夫变换检测直线--原理和Matlab实现
- 跳过Java开发的各种坑
- 图像处理算法基础（八）---霍夫变换直线检测
- Android自动化刷量、作弊与防作弊


- YV12转换到RGB32[]
- Retrofit 从入门封装到源码解析
- Hough Transform 霍夫变换检测直线
- 使用Pandas与Matplotlib分析科比职业生涯数据
- OpenCV 学习（Hough 变换提取直线）
- 霍夫变换
- 图像处理之
- Opencv学




能种头发吗



种头发原理



每日德语



你好 的德语

人工植发





暂无评论

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



液态硬盘



去除口臭的方法



关闭



人工植发

