

20180426 霍夫变换 直线检测

原文链接：<https://blog.csdn.net/songzitea/article/details/17027849>

背景引言

在图像处理中，如果图像由已知形状和大小的物体组成，需要找出物体的形状的问题。在解决这些问题的许多可能方法中，一种是在图像中移动一个合适形状和大小的掩模，寻找图像与掩模的相关性，因由于形状变形，旋转、缩放等原因，特殊的掩模常常与在特处于是的数据中特体的表示相差太大。一种非常有效的解决问题的方法是Hough变换，本节中介绍Hough变换直线检测原理和相关知识。

基本介绍

霍夫变换(Hough Transform)是图像处理中的一种特征提取技术，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[1]首次提出，最初的Hough变换是设计用来检测直线和曲线，起初的方法要求知道物体边界线的解析方程，但不需要有关区域位置的先验知识。这种方法的一个突出优点是分割结果的Robustness,即：对数据的不完全或噪声不是非常敏感。然而，要获得描述边界的解析表达常常是不可能的。后于1972年由Richard Duda & Peter Hart推广使用[2]，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。霍夫变换运用两个坐标空间之间的变换将在一个空间中具有相同形状的曲线或直线映射到另一个坐标空间的一个点上形成峰值，从而把检测任意形状的问题转化为统计峰值问题。

理论分析

Hough变换的方法基本思想可以从检测图像中的直线这个简单问题中看到。直线由两点 $A=(x_1, y_1)$ 和 $B=(x_2, y_2)$ 定义，所下图1(a)示。通过点A的所有直线由 $y=kx_1+q$ 表示， k 和 q 是某些值。这意味着同一个方程可以解释为参数空间 k, q 的方程。因此通过点A的所须直线可以表示为方程 $q=-x_1k+y_1$ 图1.(b)。类似地通过点B的直线可以表示 $q=-x_2k+y_2$ 。在参数空间 k 和 q 中，两条直线的唯一公共点是在原图像空间中表示连接点A和B的唯一存在的直线。

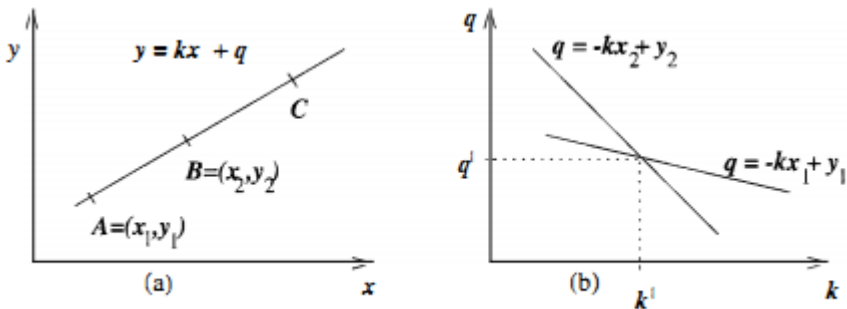


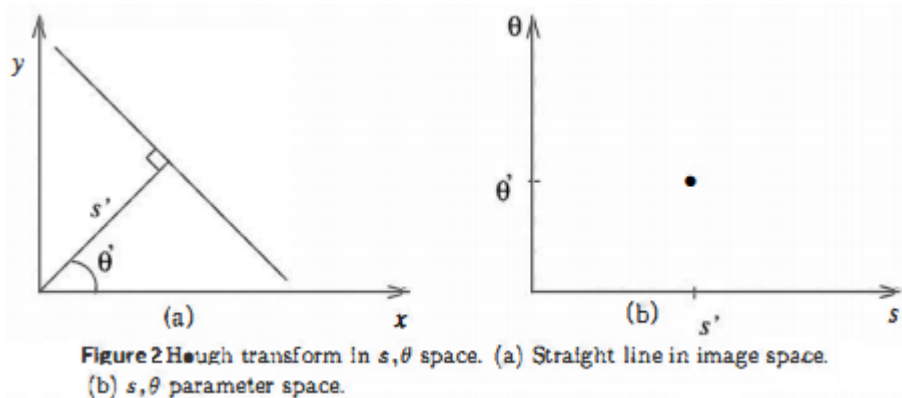
Figure 1 Hough transform principles. (a) Image space. (b) k, q parameter space.

这意味着图像中的每条直线在参数空 k, q 中由单独一个点表示，直线的任何一部分都变换为同一个点。直线检测的主要思想是确定图中所在的直线像素，将通过这些像素的所在直线变换到参数空间的对应点，在参数空间检测点 (k, q) ,此点是图像中出现的直线 $y=ax+b$ 的Hough变换的结果。

图像中所有可能的直线像素的检测，可以通过在图像中进行边缘检测子得到，所有边缘幅值超过某个阈值的像素都可以看作是可能的直线像素。在最一般的情况下，当我们没有任何有关图像中的直线信息，因此，所有方向的直线可能通过任何边缘像素。而在现在实现中，这些直线的数目是无限的，然而，为了实际目标，只能有限数目的直线方向。直线的可能方向定义了参数 k 的一个离散化，因此参数 q 也被采样为有限数目的值。所以参数空间不是连续的，而是被表示为矩形单元，称之为累计数组(accumulator array) A ，它的元素是累计单元(Accumulator cells) $A(k,q)$ 。对于每个边缘元素，确定其参数 k 和 q 。这些参数表示了通过此像素的允许方向的直线。对于每条这样的直线，直线参数 k 和 q 的值用来增加累计单元 $A(k,q)$ 的值。如果公式 $y=ax+b$ 所表示的直线出现在图像中， $A(a,b)$ 的值会被增加很多次，而次数等于直线 $y=ax+b$ 作为可能通过某个边缘像素的直线被检测到的数目。

对于任意像素 P ，通过它的直线可能是任何的方向 k ，但是第二个参数 q 受像素 P 图像坐标和方向 k 所约束。因此，存在于图像中的直线会引起图像中适合的累计单元值变大，而逼近边缘像素的其他直线，它们不对应于图像中存在的直线，对于每个边缘像素具有不同的参数 k 和 q ，所以对应的累计单元极少被增加。即：图像中存在的直线可以作为累计数组中的高值累计单元被检测出，检测到的直线参数由累计数组的坐标给出，结果是图像中直线的检测被为累计空间中的局部极值的检测。

我们可以注意到直线的参数方程 $y=kx+q$ 只适合解释Hough变换原理，在检测垂直线条和参数的非线性离散化时会遇到困难。如果直线表示成 $s=x\cos\theta+y\sin\theta$ 。Hough变换就没有这些局限性。直线还是被变换为单个点，因此可用该原理进行直线检测。如下图2所示：



我们要注意到Hough变换的重要性质是对图像中直线的残缺部分、噪声以及其它共存的非直线结构不敏感。因为从图像空间到累计空间的变换的Robustness引起的，直线残缺的部分只会造成较低的局部极值。

参考代码

OpenCV1.X版Hough Transform

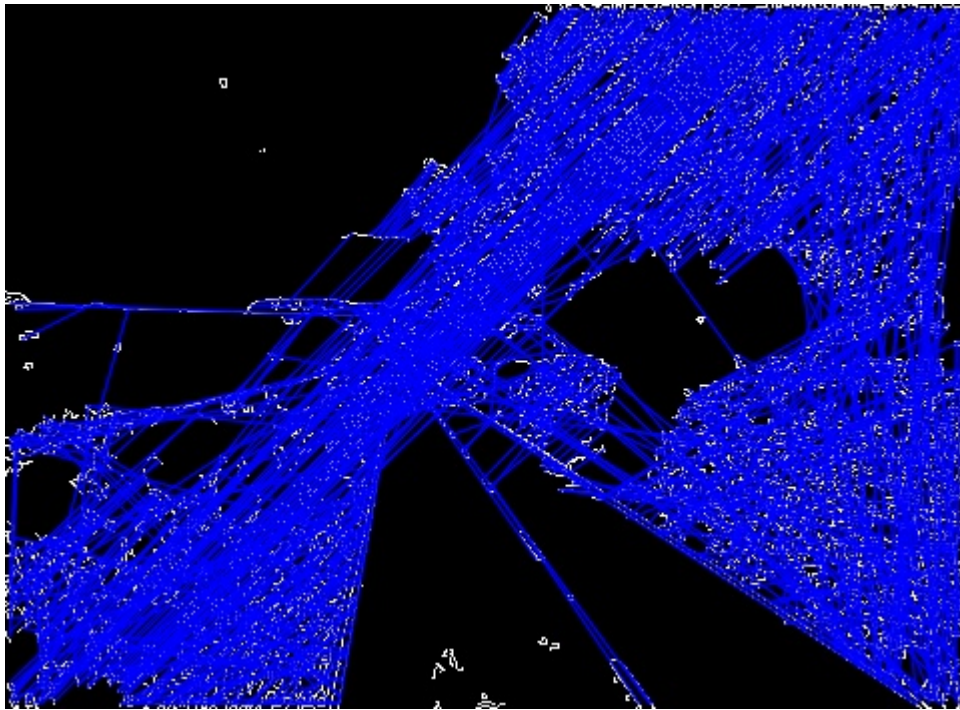
```
1  IplImage* ImageTrans::HoughLinesTransform(IplImage *pImg){
2      IplImage* src = 0,*color_dst,*dst;
3      CvMemStorage* storage = cvCreateMemStorage(0);
4      CvSeq* lines = 0;
5      int i;
6
7      /// 应该是图片通道处理相关的
8      if (pImg->nChannels == 3) {
9          src = cvCreateImage(cvGetSize(pImg), IPL_DEPTH_8U, 1);
10         cvCvtColor(pImg, src, CV_BGR2GRAY);
11     }
12     else {
13         src = cvCloneImage( pImg);
14     }
```

```
15
16    /// 创建目标图像？
17    dst = cvCreateImage( cvGetSize(src), 8 ,1 );
18    color_dst = cvCreateImage( cvGetSize(src), 8 ,3);
19
20    cvCanny( src, dst, 50, 120, 3 );
21    cvCvtColor(dst,color_dst,CV_GRAY2BGR);
22
23    /// 直接调用 霍夫变换的函数？
24    lines = cvHoughLines2(dst,storage,CV_HOUGH_PROBABILISTIC,1,CV_PI/180,20,20,30);
25
26    for( i = 0; i < lines->total; i++ ){
27        CvPoint* line = (CvPoint*)cvGetSeqElem(lines,i);
28        cvLine( color_dst, line[0], line[1], CV_RGB(0,0,255),1, CV_AA, 0 );
29    }
30    return color_dst;
31 }
```

****实验结果:**



(a) 原始图像 (b) 经Canny算子处理效果图



(c) Hough Transform处理效果图

MatLab版HoughTransform

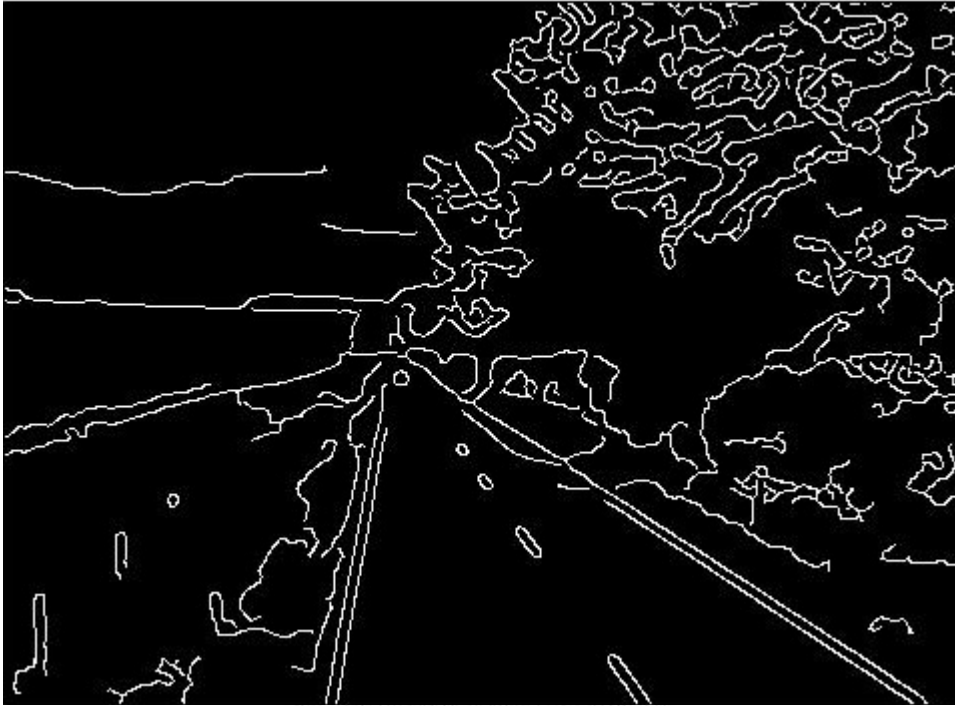
```

1  clc,close
2  I=imread('road.jpg');
3
4  I=rgb2gray(I);
5  thresh=[0.01,0.17];
6  sigma=2;
7  f = edge(double(I),'canny',thresh,sigma);
8  figure(1),imshow(f,[]);
9  title('canny Detect Result');
10
11 % 进行霍夫变换
12 [H, theta, rho]= hough(f,'RhoResolution', 0.5);
13 peak=houghpeaks(H,5);
14 hold on
15 lines=houghlines(f,theta,rho,peak);
16 figure,imshow(f,[]),title('Hough Transform Detect Result'),hold on
17 for k=1:length(lines)
18     xy=[lines(k).point1;lines(k).point2];
19     plot(xy(:,1),xy(:,2),'LineWidth',4,'Color',[.6 .6 .6]);
20 end

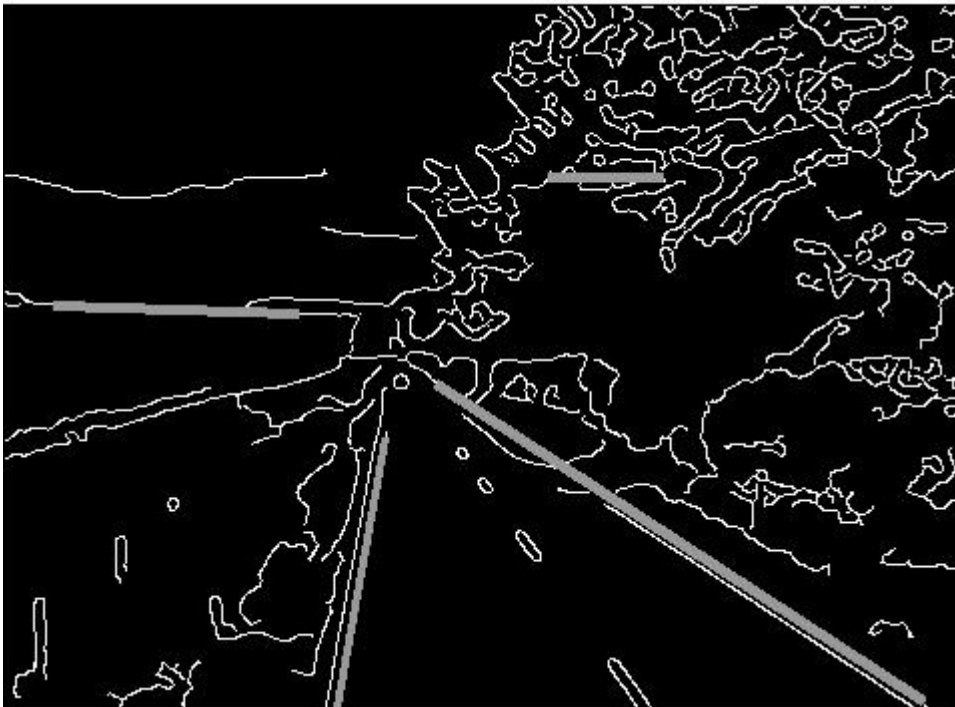
```

输出结果：

canny Detect Result



Hough Transform Detect Result



扩展代码

贴出OpenCV中对Hough Transform的直线检测功能，给各位同仁参考。代码如下：

```
1  /* Wrapper function for standard hough transform */
2  CV_IMPL CvSeq*
3  cvHoughLines2( CvArr* src_image, void* lineStorage, int method,
4                 double rho, double theta, int threshold,
5                 double param1, double param2 )
6  {
7      CvSeq* result = 0;
```

```

8
9   CvMat stub, *img = (CvMat*)src_image;
10  CvMat* mat = 0;
11  CvSeq* lines = 0;
12  CvSeq lines_header;
13  CvSeqBlock lines_block;
14  int lineType, elemSize;
15  int linesMax = INT_MAX;
16  int iparam1, iparam2;
17
18  img = cvGetMat( img, &stub );
19
20  if( !CV_IS_MASK_ARR(img))
21      CV_Error(CV_StsBadArg, "The source image must be 8-bit,single-channel");
22
23  if( !lineStorage )
24      CV_Error(CV_StsNullPtr, "NULL destination" );
25
26  if( rho <= 0 || theta <= 0 || threshold <= 0 )
27      CV_Error(CV_StsOutOfRange,"rho, theta and threshold must be positive");
28
29  if( method != CV_HOUGH_PROBABILISTIC )
30  {
31      lineType = CV_32FC2;
32      elemSize = sizeof(float)*2;
33  }
34  else
35  {
36      lineType = CV_32SC4;
37      elemSize = sizeof(int)*4;
38  }
39
40  if( CV_IS_STORAGE( lineStorage ))
41  {
42      lines = cvCreateSeq(lineType,sizeof(CvSeq),elemSize,(CvMemStorage*)lineStorage );
43  }
44  else if( CV_IS_MAT( lineStorage ))
45  {
46      mat = (CvMat*)lineStorage;
47
48      if( !CV_IS_MAT_CONT( mat->type ) || (mat->rows != 1 && mat->cols != 1) )
49          CV_Error(CV_StsBadArg,"The destination matrix should be continuous
50                  and have a single row or a single column" );
51
52      if( CV_MAT_TYPE( mat->type ) != lineType )
53          CV_Error( CV_StsBadArg,
54                  "The destination matrix data type is inappropriate, see the manual");
55
56      lines = cvMakeSeqHeaderForArray(lineType,sizeof(CvSeq),elemSize,mat->data.ptr,
57                                     mat->rows + mat->cols - 1,
58                                     &lines_header, &lines_block );
59      linesMax = lines->total;
60
61      cvClearSeq( lines );

```

```

61     }
62     else
63         CV_Error( CV_StsBadArg, "Destination is not CvMemStorage* nor CvMat*");
64
65     iparam1 = cvRound(param1);
66     iparam2 = cvRound(param2);
67
68     switch( method )
69     {
70     case CV_HOUGH_STANDARD:
71         icvHoughLinesStandard( img, (float)rho,
72                                (float)theta, threshold, lines, linesMax );
73         break;
74     case CV_HOUGH_MULTI_SCALE:
75         icvHoughLinesSDiv( img, (float)rho, (float)theta,
76                            threshold, iparam1, iparam2, lines, linesMax );
77         break;
78     case CV_HOUGH_PROBABILISTIC:
79         icvHoughLinesProbabalistic( img, (float)rho, (float)theta,
80                                     threshold, iparam1, iparam2, lines, linesMax );
81         break;
82     default:
83         CV_Error( CV_StsBadArg, "Unrecognized method id" );
84     }
85
86     if( mat )
87     {
88         if( mat->cols > mat->rows )
89             mat->cols = lines->total;
90         else
91             mat->rows = lines->total;
92     }
93     else
94         result = lines;
95
96     return result;
97 }

```

参考资料

- [1] P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures*, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959.
- [2] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM*, Vol. 15, pp. 11–15 (January, 1972).
- [3][HoughTransform](http://en.wikipedia.org/wiki/Hough_transform) From Wikipedia, the free encyclopedia.
- [4][Hough Transform](<http://planetmath.org/HoughTransform>) From PlanetMath ORG.
- [5][Hough Transform](<http://www.mathworks.cn/cn/help/images/ref/hough.html>) Documentation Center MathWorks.

[6][Hough Circle Transform]

(http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html) for OpenCV 2.4.7.0 documentation.

[7][Hough Transform 理论分析](<http://140.134.32.129/oe/oe/lcd/subject4-3.htm#Toc515070282>).

[8] Homepages Website:[Image Transforms - Hough Transform](#). Retrieved 2009.

关于Image Engineering & Computer Vision的更多讨论与交流，敬请关注本博和新浪微博[songzi tea](#).