1 Line detection

Hough变换之直线检测

Hough 变换之直线检测

⋆ Hough Transform的算法思想

在直角坐标系和极坐标系中，点、线是对偶关系。

即直角坐标系中的点是极坐标系中的线，直角坐标系中的线是极坐标系中的点。反之也成立。

如下图1所示，想要检测图像中的直线，可以转化为检测极坐标中点(θ, r)。

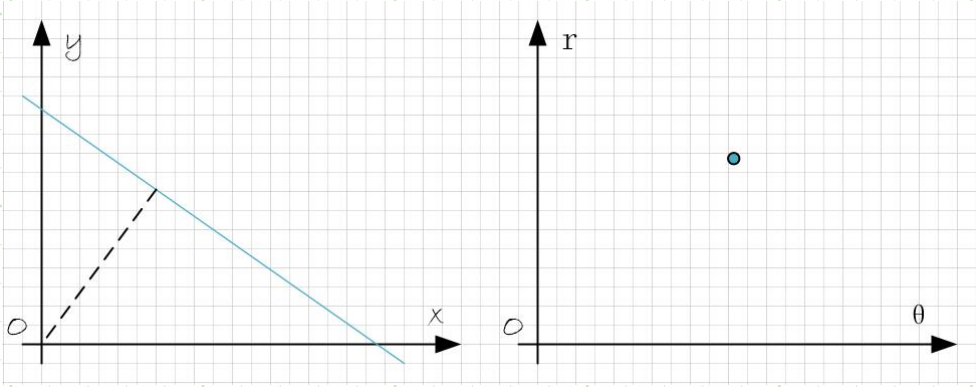

Figure 1: rectangular coordinate vs polar coordinate

⋆ Hough空间的表示

如下图2所示，图像中直线的表示，由斜率和截距表示，而极坐标中用(θ, r)表示，并且存在下式关系：

$$r = \cos(\theta) \cdot x + \sin(\theta) \cdot y$$

对于点(x_0, y_0)，代入上式，在极坐标中就是一条线(很多对(θ, r)点)：

$$r = \cos(\theta) \cdot x_0 + \sin(\theta) \cdot y_0$$

r，θ就是一对Hough空间的变量表示。

旋转的θ不容易表示，若将r，θ看成直角坐标空间。一个点(x_0, y_0)就是一个正弦曲线。

$$r = \cos(\theta) \cdot x_0 + \sin(\theta) \cdot y_0$$

如下图3所示，左图直角坐标系中的一个点，对应于右图$r-\theta$空间的一条正弦曲线。

如下图4，直角坐标系中的多个点，对应于$r-\theta$空间的多条正弦曲线。

直角坐标系的三点共线，对应于$r-\theta$空间的多线共点。

因此，我们可以通过检测$r-\theta$空间的交集点，来检测原始空间的线段。

接下来，就是要考虑将r，θ离散化，形成离散化的Hough空间，类似于一个矩阵/图像(如下图5)，用于统计交点的个数。

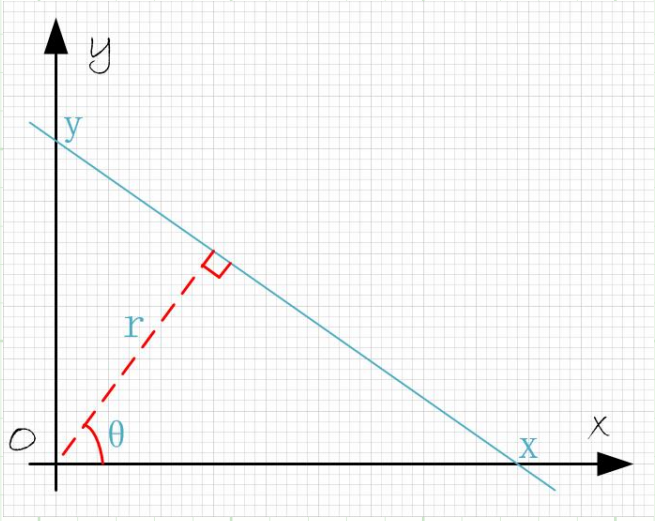

Figure 2: r-θ

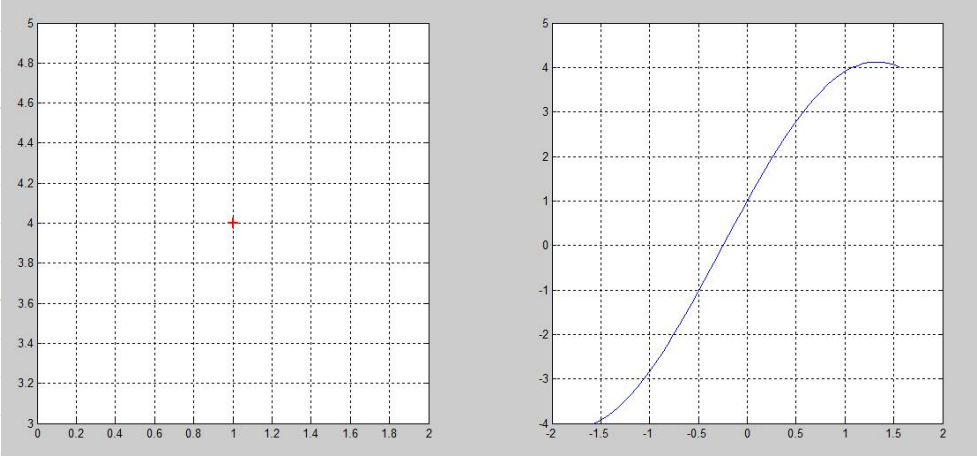

Figure 3: point vs curve

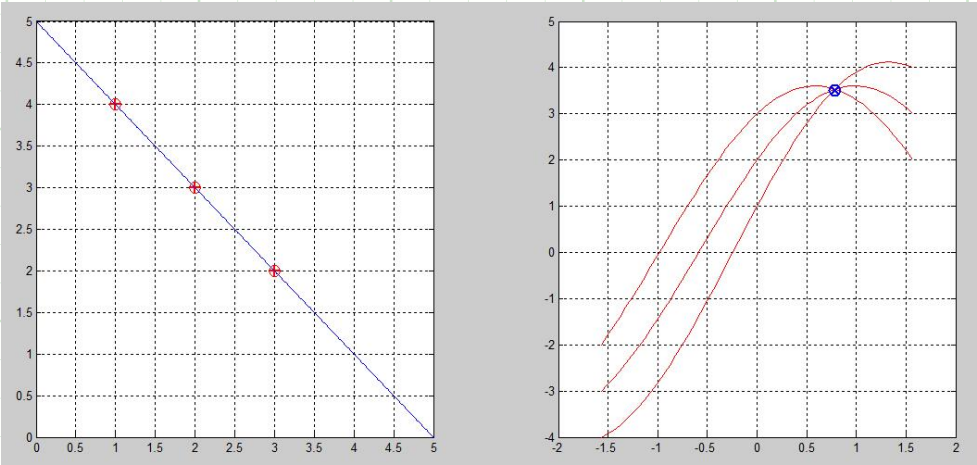

Figure 4: three points vs three curves, but one point of intersection

⋆ Hough变换代码分析

以下是使用Matlab进行直线检测的代码。

Hough Transform

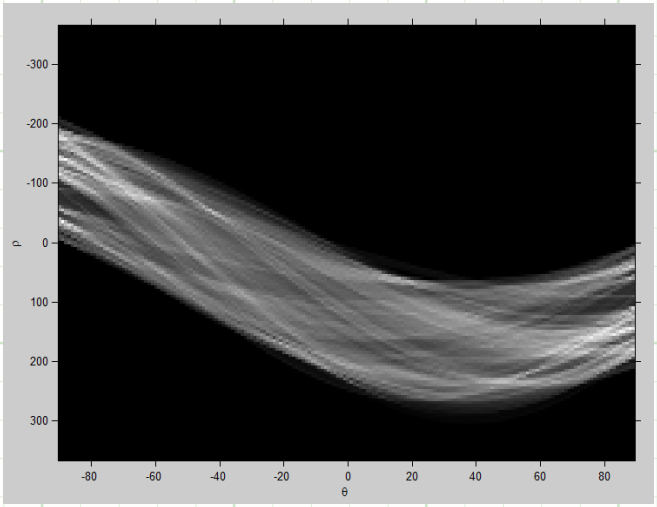

Figure 5: discretization of hough space

首先预处理，转为二值图像：

```
I = imread('road.jpg');
I = rgb2gray(I);
BW = edge(I,'canny');
```

然后进行霍夫变换：

```
[H,T,R] = hough(BW,'RhoResolution',0.5,'Theta',-90:0.5:89.5);
imshow(H,[],'XData',T,'YData',R,'InitialMagnification','fit');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;
```

检测Hough域极值点

```
P  = houghpeaks(H,50,'threshold',ceil(0.3*max(H(:))));
x = T(P(:,2));
y = R(P(:,1));
plot(x,y,'s','color','white');
```

检测直线

```
% Find lines and plot them
lines = houghlines(BW,T,R,P,'FillGap',7,'MinLength',100);
figure, imshow(I), hold on
max_len = 0;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

   % plot beginnings and ends of lines
   plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
   plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');

   % determine the endpoints of the longest line segment
   len = norm(lines(k).point1 - lines(k).point2);
   if ( len > max_len)
```

```
16          max_len = len;
17          xy_long = xy;
18      end
19  end
20
21  % highlight the longest line segment
22  plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','cyan');
```

实验结果

$r-\theta$空间及前50个极值点：

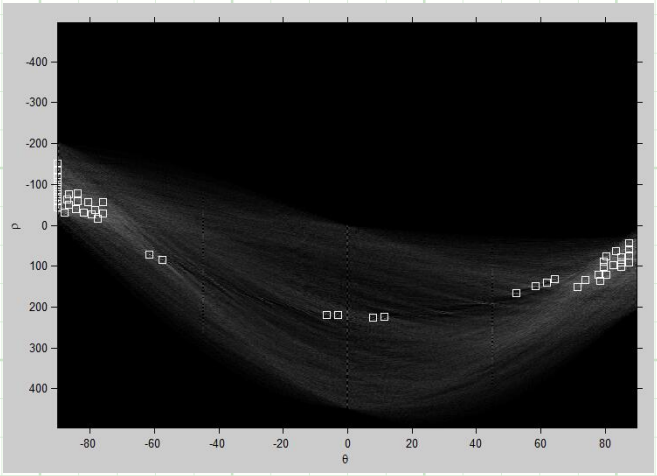

Figure 6: $r = \theta$ space

最终车道直线检测结果：

Figure 7: Results

[注]所有的代码可以在此处免费下载：http://download.csdn.net/detail/ws_20100/9492054