Las funciones de OpenCV HoughLines y HoughLinesP son usadas para detectar líneas en una imagen, ambas funciones usan el método HoughTransform usado para buscar líneas, círculos y otras formas básicas.  
  
Para la detección de líneas usamos dos métodos:

* El método estándar HougLines Standard Hough Line Transform.
* El método probabilístico es más eficiente HougLinesP Probabilistic Line Transform.

Antes de usar alguna de estas funciones primero aplicaremos dos funciones a la imagen, Canny para detectar los bordes de la imagen y cvtColor para cambiar la imagen de escala de grises a BGR y poder mostrar los resultados.

*Canny(src, dst, 50, 200, 3);*

*cvtColor(dst, cdst, CV\_GRAY2BGR);*

Aplicar el método HoughLines con los parámetros indicados, de las línea # 3 a 13 se dibujan la líneas encontradas:

* dst: proporcionado por Canny.
* lines: almacena las lineas detectadas.
* rho : usaremos 1 pixel.
* theta: usaremos 1 grado (CV\_PI/180).
* threshold: numero mínimo de líneas a detectar.
* srn and stn: zero por defecto.

*vector<Vec2f> lines;*

*HoughLines(dst, lines, 1, CV\_PI/180, 200, 0, 0 );*

*for ( size\_t i = 0; i < lines.size(); i++ ) {*

*float rho = lines[i][0], theta = lines[i][1];*

*Point pt1, pt2;*

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

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

*pt1.x = cvRound(x0 + 1000\*(-b));*

*pt1.y = cvRound(y0 + 1000\*(a));*

*pt2.x = cvRound(x0 - 1000\*(-b));*

*pt2.y = cvRound(y0 - 1000\*(a));*

*line( cdst, pt1, pt2, Scalar(0,0,255), 3, CV\_AA);*

*}*

Para aplicar el método HoughLinesP se hace  de manera similar solo cambian algunos parámetros y la manera como devuelve las lineas:

* dst: proporcionado por Canny.
* lines: almacena las lineas detectadas.
* rho : usaremos 1 pixel.
* theta: usaremos 1 grado (CV\_PI/180).
* threshold: numero mínimo de líneas a detectar.
* minLinLength: mínimo de puntos que forman una línea.

*vector<Vec4i> lines;*

*HoughLinesP(dst, lines, 1, CV\_PI/180, 50, 50, 10 );*

*for ( size\_t i = 0; i < lines.size(); i++ ) {*

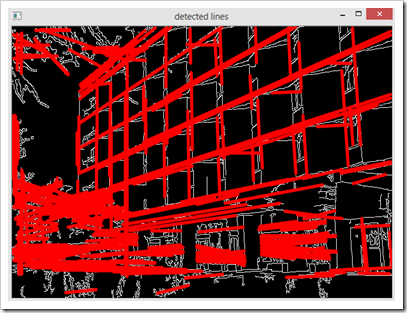
*Vec4i l = lines[i];*

*line( cdst, Point(l[0], l[1]),*

*Point(l[2], l[3]), Scalar(0,0,255),*

*3, CV\_AA);*

*}*

[](http://lh4.ggpht.com/-yTGKPCdGaHA/UqJNRPGV2NI/AAAAAAAAAsQ/_ycfonwvAw4/s1600-h/detector-de-lineas3.png)