Hough Transform User Manual

Introduction

The Hough transform is a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the *classical* Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, *etc*. A *generalized* Hough transform can be employed in applications where a simple analytic description of a feature(s) is not possible. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise. Due to the computational complexity of the generalized Hough algorithm, we have focused on the classical Hough transform in our project.

How It Works

The Hough technique is particularly useful for computing a global description of a feature(s) (where the number of solution classes need not be known *a priori*), given (possibly noisy) local measurements. The motivating idea behind the Hough technique for line detection is that each input measurement (*e.g.* coordinate point) indicates its contribution to a globally consistent solution (*e.g.* the physical line which gave rise to that image point).

As a simple example, consider the common problem of fitting a set of line segments to a set of discrete image points (*e.g.* pixel locations output from an edge detector). Figure 1 shows some possible solutions to this problem. Here the lack of *a priori* knowledge about the number of desired line segments (and the ambiguity about what constitutes a line segment) render this problem under-constrained.

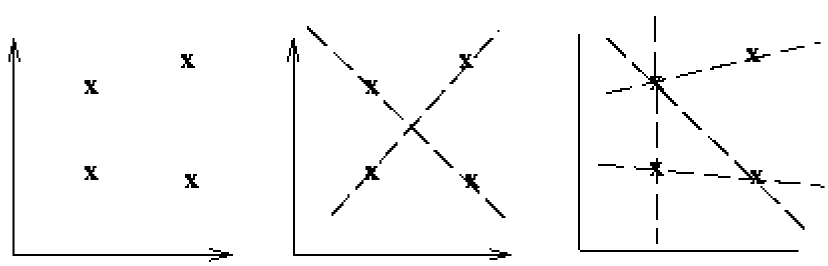


Figure 1 a) Coordinate points b) and c) Possible straight-line fittings.

We can analytically describe a line segment in a number of forms. However, a convenient equation for describing a set of lines uses *parametric* or *normal* notion:

$$x\cos\theta + y\sin\theta = r$$

where r is the length of a normal from the origin to this line and θ is the orientation of r with respect to the X-axis (See Figure 2.) For any point (x,y) on this line, r and θ are constant.

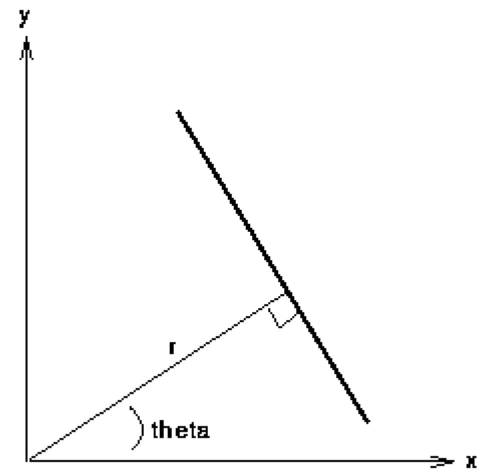


Figure 2 Parametric description of a straight line.

In context of image analysis, the coordinates of the point(s) of edge segments (i.e. (x,y)) in the image are known and therefore serve as constants in the parametric line equation, while r and θ are the unknown variables we seek. If we plot the possible (r,θ) values defined by each (x_i,y_i), points in cartesian image space map to curves (*i.e.* sinusoids) in the polar Hough parameter space. This *point-to-curve* transformation is the *Hough transformation for straight lines*. When viewed in Hough parameter space, points which are collinear in the cartesian image space become readily apparent as they yield curves which intersect at a common (r,θ) point.

The transform is implemented by quantizing the Hough parameter space into finite intervals or *accumulator cells*. As the algorithm runs, each (x_i,y_i) is transformed into a discretized (r,θ) curve and the accumulator cells which lie along this curve are incremented. Resulting peaks in the accumulator array represent a corresponding straight line that exists in the image. We can use this same procedure to detect other features with analytical descriptions. For instance, in the case of *circles*, the parametric equation is

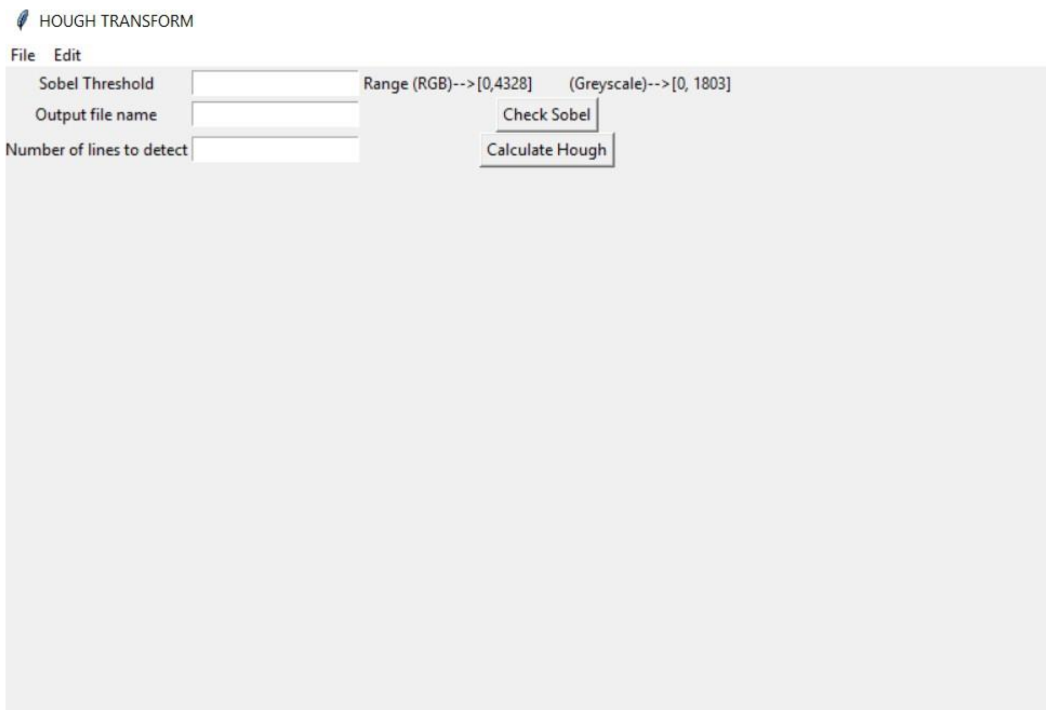
$$(x - a)^2 + (y - b)^2 = r^2$$

where a and b are the coordinates of the center of the circle and r is the radius. In this case, the computational complexity of the algorithm begins to increase as we now have three coordinates in the parameter space and a 3-D accumulator. In general, the computation and the size of the accumulator array increase with the number of parameters.

Guidelines for Use

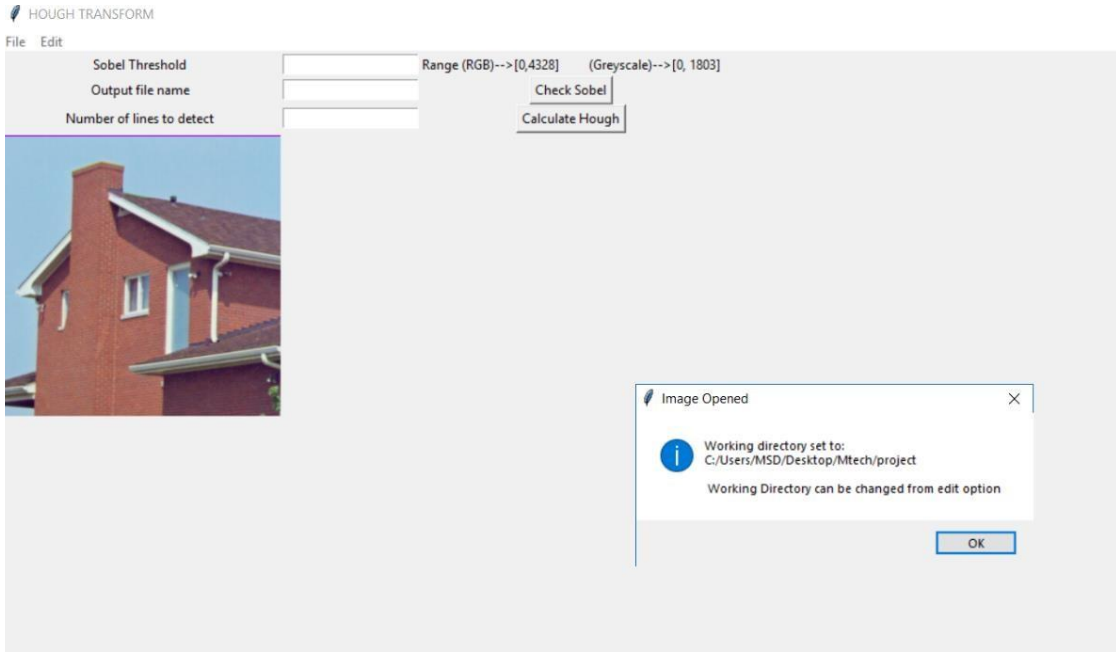
The Hough transform can be used to identify the parameter(s) of a curve which best fits a set of given edge points. This edge description is commonly obtained from a feature detecting operator such as the Roberts Cross, Sobel or Canny edge detector and may be noisy, *i.e.* it may contain multiple edge fragments corresponding to a single whole feature. Furthermore, as the output of an edge detector defines only *where* features are in an image, the work of the Hough transform is to determine both *what* the features are (*i.e.* to detect the feature(s) for which it has a parametric (or other) description) and *how many* of them exist in the image.

In our project we have used Sobel edge detector to get edges which are used by Hough transform. The GUI used is as shown:

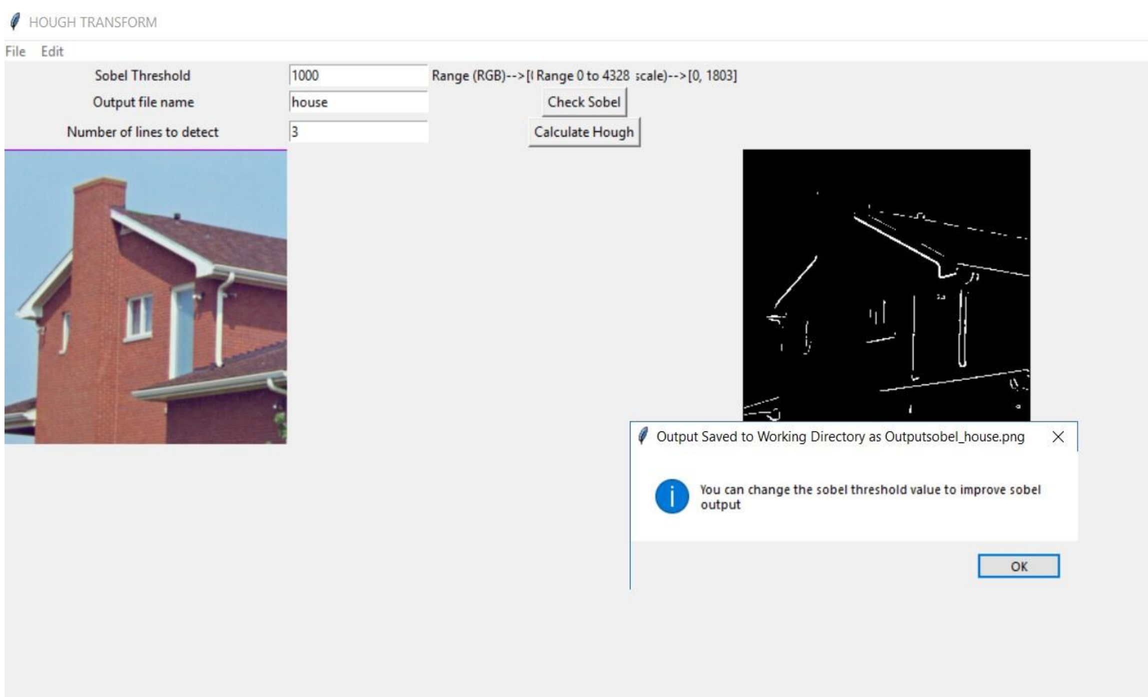


The steps to be followed to use this application are as follows:

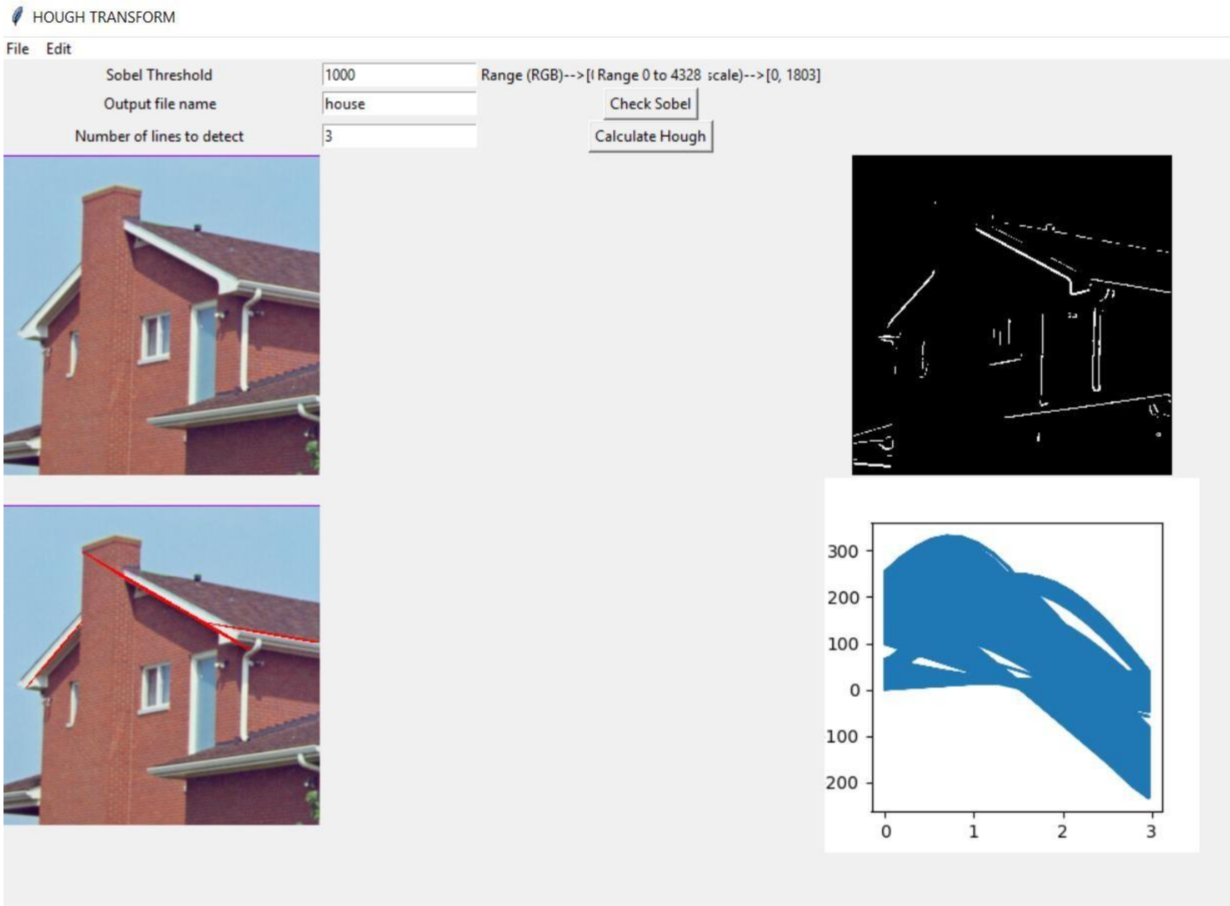
1. **Run** the **Hough_Transform.exe** executable file.
2. Select the image which you would like to work on by clicking on **File→Open**. An Image Dialog box will open in which you can browse and select the image in your computer.
3. On selecting the image, it will be displayed along with current working directory.



4. Click **OK** and if you want to change directory else choose **Edit**→ **Change directory**. Choose directory where you want to save your output file.
5. Now choose **Sobel Threshold** as per limits given correspondingly for RGB and grayscale image.
6. Write your output file name and click on **Check Sobel.** The file will be saved in working directory as *Outputsobel_filename.png* which you have chosen earlier.

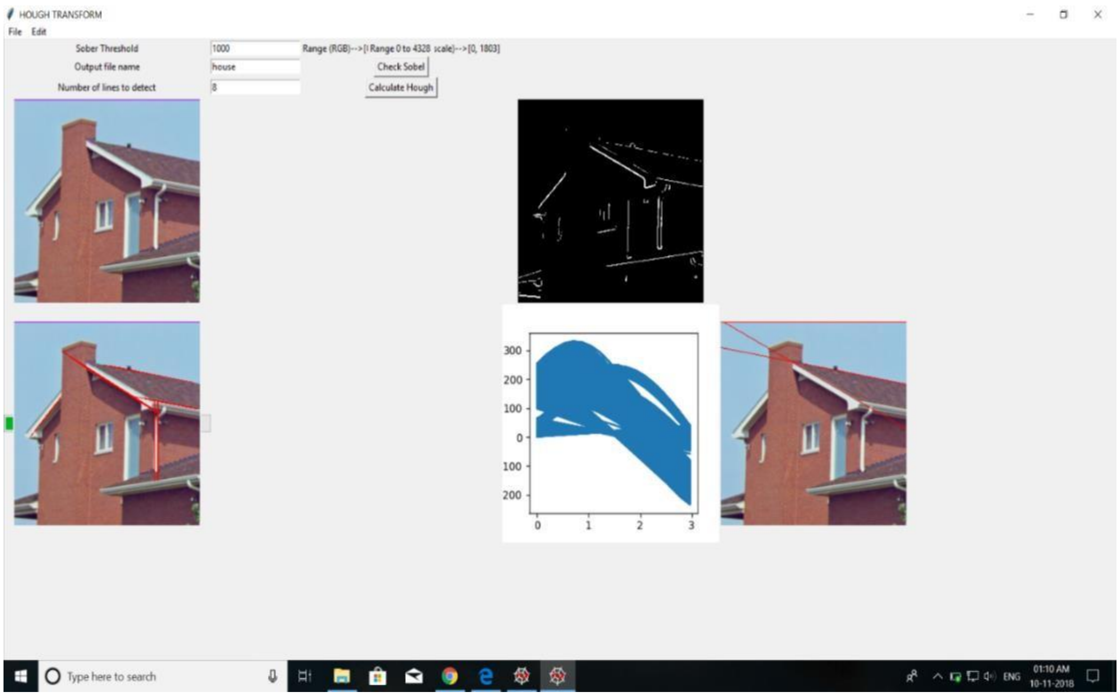


7. Choose **Number of lines to detect** and click on **Calculate Hough** to get the Hough transformed image in which lines are detected. The output file is saved as *Houghout_filename.png*.



The Sobel output along with Hough output of three lines detected is shown. Red lines show the detected lines in house. Also (r,θ) plane is displayed. r (integer values in accumulator array) is on vertical axis and θ (degrees) is on horizontal axis. This parametric equation of line eliminates the drawback of coordinate equation in which the size of accumulator array becomes infinite as line becomes vertical.

8. Users can use **Default Hough** in **Edit** to get Hough Transformed image using OpenCV2 libraries in python. The output is saved as *DHoughout_filename.png*.



- **Explanation of Range of RGB and grayscale:**

The number of gray levels in a single band of RGB image is 255. For three bands the number of bands is 765. The maximum output of Sobel operator (in both x or y direction) in a single band is 1020. For the 3 RGB bands the output is $3 * 4 * 255 = 3060$.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

$$|G| = \sqrt{Gx^2 + Gy^2} \quad (|G| = |Gx| + |Gy|)$$

Figure 3 Sobel Edge Detector

Sobel output is $x = \sqrt{3600^2 + 3600^2} = 4328$, which is given in GUI.

For grayscale image we have one band of 255 DN hence maximum Sobel output in x or y is $255 * 4 = 1020$. The Sobel output is $x = \sqrt{1020^2 + 1020^2} = 1803$ as given in GUI.