Development Hough transform to detect straight lines using pre-processing filter

Younes Parsa1,4, Hasan Hosseinzadeh2,3,*, Mehdi Effatparvar1,4
1Department of Computer, Ardabil Science and Research Branch, Islamic Azad University, Ardabil, Iran
2Department of Mathematics, Ardabil Branch, Islamic Azad University, Ardabil, Iran
3Department of Mathematics, Ardabil Science and Research Branch, Islamic Azad University, Ardabil, Iran
4Department of Computer, Ardabil Branch, Islamic Azad University, Ardabil, Iran

ABSTRACT

Image recognition is one of the most important field in image processing that in recent decades had much attention. Due to expansion of related fields with image processing and various application of this science in machine vision, military science, geography, aerospace and artificial intelligence and lots of other aspects, outstand the importance of this subject. One of the most important aspects of this case is line detection that it could be base of image recognition. Because all pictures, shapes and even whole written in picture has been formed from lines. Accordingly, exact line detection can be basis to exact image recognition. One of the methods that are used in this aspect is to identifying lines, using Hough transform. This technique is one of useful and practical method in the field of line detection. Problem with this approach is the errors in detected lines that with improving this method can lead thy way for improvement of lots of other application in this topic. In this thesis a method for improving the Hough transform in line detection is proposed. According to the achieved results it can be seen the effect of proposed method in improving Hough transform.

Keywords
Hough transform, Images text, Morphological operators, Wiener filter, Edge detection canny

1. Introduction

The Hough transform is a technique in which features are extracted that is used in image analysis and digital image processing. Previously the classical Hough Transform worked on the identification of lines in the image but later it has been extended to identifying positions of shapes like circles and ellipses [1,15,18]. In automated analysis of digital images, there was a problem of detecting simple geometric shapes such as straight lines, circle, etc. So in the pre-processing stage edge detector has been used to obtain points on the image that lie on the desired curve in image space. But due to some imperfections in image data or in the edge detector, some pixels were missing on the desired curve as well as special deviation between the geometric shape used and the noisy edge pixels obtained by the edge detector. So to refine this problem Hough transform is used. In this the grouping of edge pixels into an object class is performed by choosing appropriate pixels from the set of parametric image objects. (Fig1)

The simplest case of Hough transform is finding straight lines that are hidden in large amounts of image data. For detecting lines in images, the image is first converted into binary image using some form of thresholding and then the positive or suitable instances are added into the dataset. The main part of Hough transform is the Hough space. Each point (d, T) in Hough space matches to a line at angle T and distance d from the origin in the data space. The value of a function in Hough space gives the point density along a line in the data space. (Fig2)

For each point in the Hough space, consider all the lines which go through that point at an actual discrete set of angles that are chosen on the priority basis. For each angle T, calculate the distance to the line through the point at that angle and discretize that distance using a priori chosen discretization, giving value d. Now make a corresponding discretization of the Hough space. This will result in a set of boxes in Hough space. These boxes are called the Hough accumulators. For each line it considers above, we increment a count (initialized at zero) in the Hough accumulator at point (d, T). After considering all the lines through all the points, a Hough accumulator with a high value will probably correspond to a line of points [2,3]
2. Hough transform

The HT is a global operation that maps a line to a point. Consider object points in image space (the xy-plane) as shown in Fig. 3(a). Each such object point, say object point A, in image space has an infinite number of straight lines that could pass through it according to its slope-intercept representation, \( y_i = x_i \beta + \gamma \). Rewriting this representation as \( \beta = -\frac{x_i}{y_i} + \frac{\gamma}{y_i} \), and upon realizing that \( x \) and \( \beta \) can take many possible values; one can construct a \( x\beta \) space, which is called Hough space. In Hough space, the equation for all possible lines going through point A(\( x_i, y_i \)) is represented as \( \beta = -\frac{x_i}{y_i} + \frac{\gamma}{y_i} \). Thus, a single point in an image space corresponds to a straight line in the Hough space shown in Fig. 3(b). The Hough space is usually searched for local maxima where the parameters of lines that connect the points in the input image are determined. Once a Hough space is developed, the line detection problem in an image space becomes considerably simpler local maxima searching problem in Hough space. With a line slope of infinity, the y-intercept of a line becomes negative infinity. The most popular way to overcome this problem is by using an alternate analytic representation of lines [15,18]. A straight line can be expressed in polar coordinates as \( \rho = x \cos \theta + y \sin \theta \), where the pair (\( \rho, \theta \)) defines a vector from the origin to the nearest point on the line. The main utility of the \( \rho \theta \) Hough space is that it can easily handle vertical lines arising in image space. Regardless of which representation is used, the main disadvantage of the HT is that the extent of the line is sometimes difficult to determine. Further processing must be used to determine the extent of a line.

3. Edge Detection

Edge detection works on the idea of the identification of points in the digital image at which the image brightness changes sharply. The points at which image brightness changes sharply are organized into a set of curved line segments termed as edges [4,5,6]. Edge detection is a fundamental tool in image processing particularly in the areas of feature detection and feature extraction. Applying an edge detection algorithm to an image may significantly reduce the amount of data to be processed and may therefore filter out information that may be regarded as less relevant, while preserving the important structural properties of an image. If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified. However, it is not always possible to obtain such ideal edges from real life images of moderate complexity [7,15,16].

Is usually the first step in the analysis of image segmentation. The action part of the image is divided into parts of individual constituents. The segmentation of the issue depends on the intended use when objects were separated must be stopped segmentation. The boundary between an object and an image of the border between overlapping objects, defined edges. Ideally, if we assume that the intensity of the same image with different levels of intensity surrounding objects, then any significant change in the amount of edge intensity can be taken into account. By this definition, if the edges of an image to be determined, place all the objects in the specified image and its basic properties such as the environment, etc. can easily be measured [8,9,15].

Edge detection based on the first derivative and second derivative operators Sobel operator [3,15,17], operator Prewitt.
Roberts operator [5,15,17] Laplacian operators. In these methods, stereotypes around the image gradient and Laplacian convolution operator adapt. The amount of the brightness level changes in several directions and reveals the range by applying a threshold value on the resulting image, edge extraction are associated with stereotypes. Although the classical methods of smoothing methods did not use them this problem by calculating average picture was solved. The reason for this is that derived estimates were calculated using a relatively stronger than the noise of neighboring pixels of two pixels, so the idea operators have been proposed [9]. Some of them is an extended version of the previously mentioned detector. For example, a mask made by prewitt operator $7 \times 7$

Based on the assessment of horizontal columns the first derivative of the image is presented as a matrix (1) is provided:

$$
H_r = \frac{1}{21}
\begin{bmatrix}
-1 & -1 & -1 & 0 & 1 & 1 \\
-1 & -1 & -1 & 0 & 1 & 1 \\
-1 & -1 & -1 & 0 & 1 & 1 \\
-1 & -1 & -1 & 0 & 1 & 1 \\
-1 & -1 & -1 & 0 & 1 & 1 \\
-1 & -1 & -1 & 0 & 1 & 1 \\
\end{bmatrix}
$$

3. Non-maximum suppression: Only local maxima should be marked as edges.

4. Double thresholding: Potential edges are determined by thresholding, edge pixels stronger than the high threshold are marked as strong; edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked.

5. Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge [17,19].

In this definition, an optimal edge detector includes the following things:

1. **Good detection** – low probability of not marking real edge points, and falsely marking non-edge points is the filter, $G$ is the signal edge, denominator is the root-mean-square response to noise $H(f)$ only.

2. **Good localization** – edges marked through this algorithm should approach as close as possible to the edge in the real image. Use reciprocal of the rms distance of the marked edge from the center of the true edge.

3. **Minimal response** – a given edge in the image should only be marked once so as to reduce false edges. [8,11,14,17,19]

![Fig3: masks made of (a)sobel, (b)prewitt, (c)zerocross (d) roberts, (e) log , (f) original Image](image)

![Fig4: Input Image](image)

### 3.1. Canny edge detector

The Canny edge detector [9,18,19] is an edge detection algorithm that uses a multiple stage algorithm so as to detect edges in images. Its aim is to discover the optimal edge detection. The algorithm canny runs in 5 separate steps:

1. **Smoothing**: Blurring of the image to remove noise.
2. **Finding gradients**: The edges should be marked where the gradients of the image has large magnitudes.

![Fig5: Edge detection using Canny algorithm](image)
morphology of a wide range of image processing operations that process the images based on the shapes. The value of each pixel in the output image morphology operation by the corresponding pixel in the input image and its neighbors. The most basic morphological operators Dilation expansion and Erosion wear Dilates pixels to the image borders. While Erosion removes pixels from the image borders. Open areas usually smooth operator environment, break down and remove the protrusions of the light beam. Depending on the size and shape of the structure element is used for functional morphology.[11]

In grayscale morphology, images are functions mapping a Euclidean space or grid E into \( \mathbb{R} \cup \{\infty, -\infty\} \), where \( \mathbb{R} \) is the set of reals, \( \infty \) is an element larger than any real number, and \( -\infty \) is an element smaller than any real number.

Grayscale structuring elements are also functions of the same format, called "structuring functions". Denoting an image by \( f(x) \) and the structuring function by \( b(x) \), the grayscale dilation of \( f \) by \( b \) is given by
\[
(f \oplus b)(x) = \sup_{y \in E} f(y) + b(x - y),
\]
where "sup" denotes the supremum.

Similarly, the erosion of \( f \) by \( b \) is given by
\[
(f \ominus b)(x) = \inf_{y \in E} f(y) - b(y - x),
\]
where "inf" denotes the infimum.

Just like in binary morphology, the opening and closing are given respectively by
\[
f \ast b = (f \ominus b) \oplus b, \quad f \bullet b = (f \oplus b) \ominus b.
\]

5. The proposed Improved Hough transform algorithm

Eight step algorithm is proposed that the first step is entering the image pre-processing step and seven-step. Finally, the Hough transform to identify lines used in the first stage Wiener filter and operators Morphology and the Hough transform apply the proposed method to improve the identification of lines by transforming Hough seven step preprocessing step used in the process, is proposed steps to fully process the proposed algorithm can be seen later.[15,19]

Step One: Sign in picture
Step Two: Convert the image to gray
Step Three: Apply filters motion
Step Four: Apply filters Gaussian
Step Five: Apply filters Weiner
Step Six: Apply operator morphology
Step Seven: Applying an edge detector canny
Step Eight: Apply Hough algorithm

\[\text{Step One: Sign in picture}\]
\[\text{Convert the image to gray}\]
\[\text{Apply Filters Motion}\]
\[\text{Apply filters Gaussian}\]
\[\text{Apply filters Weiner}\]
\[\text{Apply operator morphology}\]
\[\text{Applying an edge detector Canny}\]
\[\text{Apply Hough Algorithm}\]
At this point, the image can be brought in for processing and identification of lines. The best image to image recognition accuracy with the lowest noise in the picture.

**Second step: convert the image to gray**

The model image, gray image is important because in most applications there is no need for a color image having only a gray image will be enough. In this step of the method the input image is converted into a gray image and then processing the image of gray will be used also if the image is a color entry can convert the image to gray.

**Step Three: Apply the Motion filter Motion blur**

is the apparent streaking of rapidly moving objects in a still image or a sequence of images such as a movie or animation. It results when the image being recorded changes during therecording of a single exposure, either due to rapid movement or long exposure. Suppose you’re taking pictures with a camera in your hands, if in the time taken tomovethe camera have the image will be drawn. Filter algorithm of the same operation on Image. Len is the determinant of motion. \( \theta \) also determines the angle he does. For example 90 to \( \theta \) as shock and the image up and down or a value of 0 to \( \theta \) as you move the image to the left and right. This filter is one of the inputs for the Wiener filter.[13,18]

![Image](image1.png)

**Step Four: Apply Gaussian**

Low frequency pixel intensity difference with neighboring pixels is small. The image is smoothed by convolving with the Gaussian filter. Smoothing is to reduce the noise within an image. Gaussian smoothing function is given by

\[
G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)
\]

Where \( \sigma \) is standard deviation. 1. Gaussian smoothing is very effective for removing Gaussian noise.[12,15]
2. The weights give higher significance to pixels near the edge (reduces edge blurring).
3. They are low linear pass filters
4. Computationally efficient (large filters are implemented using small 1D filters).
5. Rotationally symmetric (perform the same in all directions).

**Step Five: Apply filters Weiner**

The weiner filtering is optimal in terms of the mean square error. In other words, it minimizes the overall mean square error in the process of inverse filtering and noise smoothing. The weiner filtering is a linear estimation of the original image. The approach is based on a stochastic framework. The orthogonality principle implies that the weiner filter in Fourier domain can be expressed as follows:

\[
W(f_1, f_2) = \frac{H^*(f_1, f_2)S_{xx}(f_1, f_2)}{|H(f_1, f_2)|^2S_{xx}(f_1, f_2) + S_{yy}(f_1, f_2)}
\]

Where are \( S_{xx}(f_1, f_2) \) and \( S_{yy}(f_1, f_2) \) respectively power spectra of the original image and the additive noise, and

![Image](image2.png)
\( H(f_1, f_2) \) is the blurring filter. It is easy to see that the weiner filter has two separate parts, an inverse filtering part and a noise smoothing part. It not only performs the deconvolution by inverse filtering (high pass filtering) but also removes the noise with a compression operation (low pass filtering). To implement the weiner filter in practice we have to estimate the power spectra of the original image and the additive noise. For white additive noise the power spectrum is equal to the variance of the noise. To estimate the power spectrum of the original image many methods can be used. A direct estimate is the period gram estimate of the power spectrum computed from the observation.

\[ S_{yy}^p = \frac{1}{N^2} [ Y(k, l) Y(k, l)^* ] \]

Where \( y(k, l) \) is the dft of the observation. The advantage of the estimate is that it can be implemented very easily without worrying about the singularity of the inverse filtering[5]. Another estimate which leads to a cascade implementation of the inverse filtering and the noise smoothing is

\[ S_{xx} = S_{yy}^p - S_{yy}^n \]

Which is a straightforward result of the fact: If the power spectrum \( S_{yy}^p \) can be estimated directly from the observation using the period gram estimate. This estimate results in a cascade implementation of inverse filtering and noise smoothing[6]

\[ W = \frac{1}{H} \frac{S_{yy}^p - S_{yy}^n}{S_{yy}^p} \]

The disadvantage of this implementation is that when the inverse filter is singular, we have to use the generalized inverse filtering. People also suggest the power spectrum of the original image can be estimated based on a model

---

**Step Seven: Identify the edges of the image using the canny algorithm**

At this point you're using the edges detected. Usually edge detection algorithms like you. A threshold level of the threshold by subtracting the intensity level where the intensity is low, edge detection. Where high brightness is too weak to be strong edge detection. When you need to be strong edge detection with the use strong slope of you. Edge detection algorithm, do you have three different threshold brightness level, if the difference between the intensity level of the first threshold level, she is more known as edge, even if it is smaller than the second threshold value, an edge, and if the two weak edge, the edge of the preserve continuity.

---

**Fig Eighth step: Apply algorithm Hough**

At this stage, after applying the input image pre-processing algorithms to identify the lines enter Hough will become. Infact, this is the part that identifies lines. But according to the Hough transform is highly sensitive to noise, so it does not produce significant results in identifying lines the end result is to produce shortcomings. The proposed method was explained that the
main problem, as far as significantly reduced and also the changes that occur in the image, improve outcome in recognition of the lines.[15,18]

5. Comparison Results

To be able to identify the lines of the proposed algorithm Review, we have the Hough transform method compared with conventional methods. The benchmark comparison of the number of lines detected correctly. We will use as a benchmark. In this section we try to compare the use of multiple images with different characteristics, to improve the proposed algorithm can be seen in all kinds of images. Also at the end to compare the number of lines per picture will be compared properly detected and the number of lines Methods to identify the wrong line, also will be evaluated. Below are some pictures that will be used to compare the two methods, have been proposed, and the purpose of this algorithm is also marked on them that we produce the desired output for the specific purposeofAnd optimal image output is output that tries to fully reach the desired goal.

5.1. Terms of performance

To run and compare the results for both approaches, namely conventional Hough improved application environment MATLAB is used for the comparison of images of the same size and same quality with the same suffix used And to complete the comparison conditions for both methods were equal to conclude there was an error No. In the form of an aerial view of a city to find lines the streets of the city. In a specific direction the lines identified by the green line are used. As can be seen, in Haugh Typical of the left two lanes have been identified, none of the lines detected line is not correct and Error occurred one hundred percent, but the figure on the left is generated using the proposed method Two completely correct, is detected. That is to say, one hundred percent recovery.

5.2. comparison of two Hough normal output and improved Hough

In figure (5-2) you can see an aerial view of the road is the goal in this picture Find Road ways has been given. As you can see in the image on the right to use the conventional Hough lines have been identified, only a specifically identified, but left image using Hough developed specified identify the correct line number five, which is seen a five-fold growth.

5.3. comparison of two Hough normal output and improved Hough

In figure (5-3) which is shaped like a chessboard, to find the horizontal and vertical lines between houses chess. As in the right figure lines have been identified using conventional Hough, see the only horizontal lines and vertical lines have been identified, but the proposed method four vertical line more than half of the ordinary is detected.
5.4. comparison of two Hough normal output and improved Hough

Figure 5-4 that form the facade of a building facing is to find lines in the image. The results produced by conventional half right and left images produced using Hough algorithm is proposed.

A: Hough improved  B: Hough normal

5.5. comparison of two Hough normal output and improved Hough

And more form:

image original

A: Hough improved  B: Hough normal

7. Results and Discussion

Given the importance of identifying lines and Hough transform, given that one of the methods in the field lines is detected, the better this method is also very important. Accordingly, in this article was to provide a way to improve the conversion and according to the results produced in this section see results resulting from the proposed method is a significant improvement in the detection lines. Also according to the accuracy of the results the proposed method produced accurate results also greatly increased and it can be said that the method proposed the improved Hough transform using Wiener filter and morphological operators a substantial improvement in results and the results thus created could be related to the identification process lines used. The proposed approach is likely to find straight lines and right to a 70% increase.

References
1. Hough Transform, [online available]: wikipedia.org
11. Hsiao Ying-Tung T A mathematical morphological method to thin edge detection in dark region A mathematical morphological method to thin edge detection in dark region Signal Processing and Information Technology 2004.