A Survey on Line Detection Techniques using Different Types of Digital Images

International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) ijarcet

International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)

International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 8, Issue 5, May 2019, ISSN: 2278 – 1323

A Survey on Line Detection Techniques using Different Types of Digital Images

Greeshma T K
M. Tech. Computer Science Image processing

Dr.Priya S
Professor
M. Tech. Computer Science Image Processing

Ms. Chaithanya C
Asst. Professor
M. Tech. Computer Science Image Processing

Abstract— **Line detection is a very basic, yet important problem in image processing. It is an approach based on whether sets of pixels lie on curves of a specified shape. Once detected, these curves form the edges of interest. Most of the shape information of an image is enclosed in edges. So first detect these edges in an image and by using these filters and then by enhancing those areas of image. Which contains edges, sharpness of the image will increase and image will become clearer. However the output from an edge detector is still an image described by its pixels. If lines, ellipses and so forth could be defined by their characteristic equations, the amount of data would be reduced even more. Line detection is an algorithm that takes a collection of an edge points and finds all the lines on which these edge points lie. The most popular line detectors are the Hough transform and convolution based techniques. While edges (i.e. boundaries between regions with relatively distinct gray levels) are by far the most common type of discontinuity in an image, instances of thin lines in an image occur frequently enough that it is useful to have a separate mechanism for detecting them. Here present a convolution based technique which produces an image description of the thin lines in an input image. Note that the Hough transform can be used to detect lines; however, in that case, the output is a parametric description of the lines in an image**.

Index Terms— **Hough transform, Image segmentation, Line detection, Probabilistic Hough transform, Randomized Hough transform**.

I. INTRODUCTION

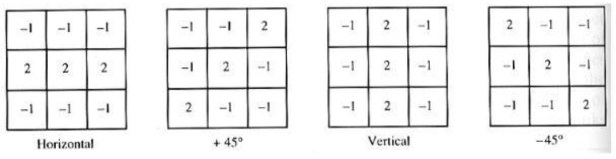
Image processing is a method of analyzing and manipulating the digital images with the computer using mathematical operators. In image processing, the input is an image and outcome may be either set of characteristics or set of the parameter of image or an image. An image comprises various information like contour of the object, its orientation, size and color. Most of the shape information of an image is enclosed in edges. So first detect these edges in an image and by using these filters and then by enhancing those areas of image which contains edges, sharpness of the image will increase and image will become clearer[3][4] by it's pixels. If lines, ellipses and so forth could be defined by their characteristic equations and the amount of data would be reduced even more. To manually extract the line information from an image can be very tiring and time- consuming especially if there are many lines in the image. An automatic method is preferable, but is not as trivial as edge detection since one has to determine which edge point belongs to which line, if any. Line detection is an algorithm that takes a collection of n edge points and finds all the lines on which these edge points lie. The most familiar line detectors are the Hough transform and convolution based techniques.[2]

II.CONVOLUTION BASE TECHNIQUE

Convolution is a simple mathematical operation which is fundamental to many common image processing operators. Convolution provides a way of 'multiplying together' two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality. This can be used in image processing to implement operators whose output pixel values are simple linear combinations of certain input pixel values.

In an image processing context, one of the input arrays is normally just a gray level image. The second array is usually much smaller, and is also two-dimensional (although it may be just a single pixel thick), and is known as the kernel. The convolution is performed by sliding the kernel over the image, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries of the image. (Note that implementations differ in what they do at the edges of images, as explained below.) Each kernel position corresponds to a single output pixel, the value of which is calculated by multiplying together the kernel value and the underlying image pixel value for each of the cells in the kernel, and then adding all these numbers together.

The line detection operator consists of a convolution kernel tuned to detect the presence of lines of a particular width n, at a particular orientation θ. Figure 1 shows a collection of four such kernels, which each respond to lines of single pixel width at the particular orientation shown.



–1	–1	–1
2	2	2
–1	–1	–1

Horizontal

–1	–1	2
–1	2	–1
2	–1	–1

+ 45°

–1	2	–1
–1	2	–1
–1	2	–1

Vertical

2	–1	–1
–1	2	–1
–1	–1	2

–45°

Figure 1. Four line detection kernels which respond maximally to horizontal, vertical and oblique (+45 and - 45 degree) single pixel wide lines.

These masks above are tuned for light lines against a dark background, and would give a big negative response to dark lines against a light background. If you are only
interested in detecting dark lines against a light background, then you should negate the mask values. Alternatively, you might be interested in either kind of line, in which case, you could take the absolute value of the convolution output. In the discussion and examples below, we will use the kernels above without an absolute value. If$_i$ denotes the response of kernel i, we can apply each of these kernels across an image, and for any particular point, if $R_i > R_j$ for all j = i that point is more likely to contain a line whose orientation (and width) corresponds to that of kernel i. One usually thresholds R_i to eliminate weak lines corresponding to edges and other features with intensity gradients which have a different scale than the desired line width. In order to find complete lines, one must join together line fragments, e.g., with an edge tracking operator.

A. Fitting a Straight Line

Lawrence Roberts has proposed the Roberts edge detection technique for detecting the edges within an image in 1965. It is a simple and computationally efficient approach. It measures the spatial gradient of an image. The pixel value at that point in the resultant image characterizes estimated absolute magnitude value of the spatial gradient of the inputted image at that point. It takes input image as gray scale image and produces edges involving in that image. The main disadvantages of this technique are that it can't detect that type of edges which are multiplies of 45 degrees and it is not symmetric. The Robert operator contains the pair of 2x2 convolution masks which are illustrated in Figure1. One mask is just to other rotated by 90 degrees. This is true because there are few single pixel width lines in this image, and therefore the detector is responding to the other high spatial frequency image features (i.e. edges, thick lines and noise). (Note that in the previous example, the image contained the feature that the kernel was tuned for and therefore we were able to threshold away the weaker kernel response to edges.) We could improve this result by increasing the width of the kernel or geometrically scaling the image.

To fit a single straight line to data, we must fit

$$y = mx + h \qquad (1)$$

where m is the gradient and h is the intercept with the y - axis. This is a very common fitting problem and the simplest is least squares fit. If we have n points y_i, x_i, being point on the line, then if we define the square error as

$$e^2 = \sum_{i=1}^{n} (y_i - (mx_i + h))^2 \qquad (2)$$

and we get the standard solution by minimizing e^2 by setting

$$\frac{\partial e^2}{\partial m} = 0 \text{and} \ \frac{\partial e^2}{\partial n} = 0 \qquad (3)$$

which has the effect of minimizing the vertical distance Between the points and the line as shown in figure 2

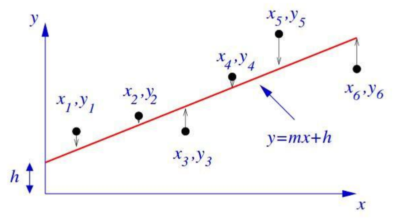


Figure 2. Least square fit to a single line of data points

The works very well for a single line, but if there is more than one line, things get rather more complicated and, as shown in figure 3, the simple least squares simply gives the best average line single line which is usually wrong. Least-Square only works if you have a single line, or are able to segment out a segment of the image that contains a single line. We need to look for something a lot more general than this.

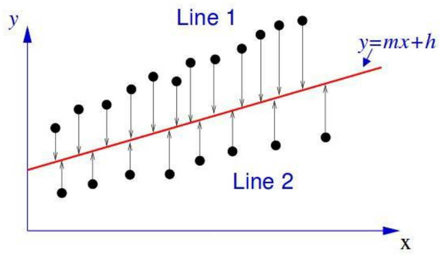


Figure 3. Least square fit two lines of data point by a single line

III. Hough Transform

Consider the idea of a line-to-point transform, as shown in figure 4 where the image data in x ,y is transformed to a m, h space, so that each line is transformed to a point. Then as points are easy to detect, then a sharp peak in the m ,h space would correspond to a line of the form n the original image space[5].

$$y = mx + h \tag{4}$$

If was have such a transform and we have multiple lines,

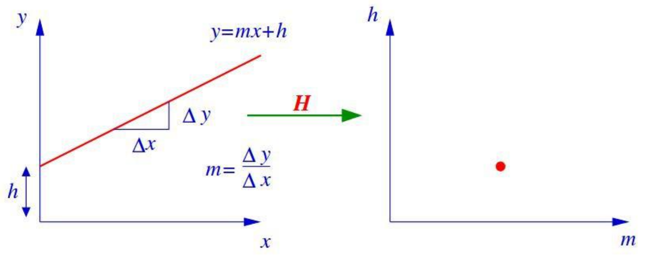


Figure 4. Concept of a line-to-point which a line is real space becomes point

then as shown in figure 5 we then simple get multiple points, one for each line. So if we can derive such a transform we have solves the general line extraction problem being able to extract any number of lines from an image. However there is a problem with the scheme as it stands since neither m or h are bounded, so that

Line ||to x axis ⇒ h not defined

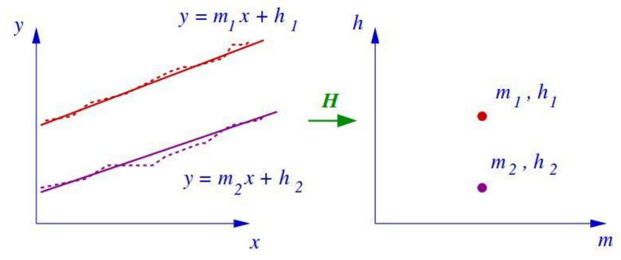


Figure 5. Line-to-point transform of multiple lines results is multiple points.

Line ||to y axis ⇒ $m \to \alpha$

Which make the simple scheme computationally impractical, however this can be modified

Hough transforms, HT, in image processing and computer vision. It has long been recognized as a technique of almost unique promise for shape and motion analysis in images containing noisy, missing, and extraneous data but its adoption has been slow due to its computational and storage complexity and the lack of a detailed understanding of its properties. However, in recent years much progress has been made in these areas. In this review we discuss ideas for the efficient implementation of the HT and present results on the analytic and empirical performance of various methods

A. Probabilistic Hough Transform

Kiryati et al[9] described an algorithm which is perhaps the easiest of the probabilistic methods to understand due to its similarity to standard Hough Transform. As with standard Hough Transform, a one-to-many mapping from image to parameter space is used. Then difference is that rather than using all M edge points, only a subset m is used. As m<M ,the complexity of the voting stage is reduced from O $(M.N_\theta)$ O $(m.N_\theta)$ Intuitively, this works because a random subset of M will fairly represent all features and noise based on the area they occupy in the image. Choosing a smaller value for m will lead to a faster algorithm but clearly it must not be so small that features can no longer be detected. Kiryati et al performed an analysis which suggested the presence of a thresholding effect for the value of m. Values of m below the threshold gave poor results whilst values above gave very good results. This threshold effect was confirmed experimentally with good results being obtained with as few as 2 % of edge points being sampled. However, the value of m must be determined on a per problem basis.

B. Randomized Hough Transform

In contrast to standard Hough Transform, the Randomized Hough Transform (RHT) uses a many-to-one mapping from image space to parameter space[4.To understand how it works, imagine an input image with a set of M active edge points. Two points are selected at random and removed from this set . The equation for the line joining these two points is obtained by solving the simultaneous equations:

$$y_1 = ax_1 + by_1 = ax_2 + b \tag{5}$$

This gives the point (a,b)in parameter space. Rather than Maintaining a 2D array of accumulators to represent parameter space, a linked list is used. The list is searched for an element representing the point (a, b) and if a match is found, its count is incremented. If none exists, a new one is created with a count of one. The selection and removal process is then repeated. Elements in the list that represent lines in the original image will be incremented multiple times. A threshold, which can be as small as 2 or 3, is used to determine which elements represent true lines.

One complication that arises with this method is that of dealing with tolerances. It is unlikely that two points in parameter space will match exactly even if they correspond to a single line. It is, therefore, necessary to use a tolerance. If a match within the tolerance is found, the old and new values of (a,b) are averaged and the count incremented by one.

Randomized Hough Transform [10] has several advantages over standard Hough Transform. The storage requirements are greatly reduced and, as only one accumulator needs to be incremented at each iteration, it is significantly faster than standard Hough Transform. These improvements become more pronounced as the dimensionality of parameter space increases. Furthermore, standard Hough Transform splits the parameter space into discrete accumulators limiting its resolution. The resolution can be increased but at the cost of performance. In contrast, Randomized Hough

Transform can support arbitrarily high resolution without affecting performance.

IV. Hough Transform Applications

Due to the merits of the Hough Transform mentioned above e.g. Noise immunity, it has been widely used in many applications. For example, 3D applications, detection of objects and shapes, lane and road sign recognition, industrial and medical applications, pipe and cable inspection, and un-dewater tracking. Some of these applications are illustrated below

A, Traffic and Transport Applications.

A system for vehicle license plate (VLP) detection is shown in [12]. The system combines the HT and a contour detecting algorithm to improve the speed. Although the input images can be taken from various distances with inclined angles, the sys- tem is able to detect multiple VLPs.In [12], a perspective centroid-based HT for Skew correction of license plate recognition is shown. It gets first the centroid of each char- actor on the plate, and then the HT is used to estimate a line passing by the centroids. Knowing the skew angle, the plate position can be corrected for a more accurate recognition. A project for rail road defect inspection is proposed in. It finds the defects of the railroad tracks with a notification to the inspector of the type and location of the defect. For example, the CHT is used to detect missing bolts, defects, and cracks in ties.

B. Biometrics and Man-Machine Interaction.

A real- time human-robot interaction system based on hand gestures and tracking is proposed in [12]. It combines the connected component label ling (CCL), and the HT. Detecting the skin color, it has the ability to extract the center of the hand, the directions and the fingertip positions of all outstretched fingers. On the other hand, the HT is applied to finger print matching in [94]. The algorithm uses a robust alignment method (descriptor -based HT). Additionally, it measures the similarity between fingerprints by considering both minutiae and orientation field information. The CHT is used to locate the eyes and the heads for the people detection in [95]. Additionally, the pupil is located in [96] using a combination of the Canny edge detector and the CHT. Moreover an iris localization algorithm is suggested in [97] where the CHT and the image statistics are used to estimate a coarse iris location in the images of the eye. The pupil is located using a bi-valued adaptive threshold and the two -dimensional shape properties. The limbic boundary is located by using the Hough accumulator and the image statistics

C. *3D Applications.*

A detector is devised for 3D lines, boxes, and blocks [12]. Using 2D Hough space for parallel line detection, this method reduces the computational cost. The 3D HT is used for the extraction of planar faces from the irregularly distributed point clouds. The work also suggests two reconstruction strategies. The first tries to detect the intersection lines and the height of the rising edges. The second assumes that all detected planar faces should model some part of the building. The experiments show that the latter is able to reconstruct more buildings with more details, but sometimes generates artificial parts which do not exist in reality. To recognize objects in 3D, proposes a 3D Hough space voting approach for determining object categories learned from artificial 3D models. The proposed method uses the ray voting for object reference points. This allows a cluster of votes to be shown in similar directions. The model can be trained with large amounts of data taken from different sources at varied scales. A modified GHT for the 3D image processing is presented in [52]. The principal modification in the GHT appears in the lookup- table indexing.

D. Object Recognition.

An approach is presented in [12] for object detection using the GHT with the color similarity between homogeneous segments of the object. The input of the algorithm is previously segmented regions with homogeneous color. According to this work, the approach is robust to illumination changes, occlusion and distortion of the segmentation output. It is able to recognize objects in spite of being translated, rotated, scaled and even located in a complex environment. The work shown in [102] proposes a robust method for recognizing objects among clutter and in the presence of occlusion. A close to real-time performance is achieved. The recognition of the objects depends mainly on matching individual features stored in a database of known objects using a fast nearest neighbor algorithm. The HT is then used to identify clusters belonging to a single object. In addition to this, recognition of multiple instances of an object is proposed.

E. Object Tracking.

A shape based tracking is per- formed using a mixture of uniform and Gaussian Hough (MOUGH)[12]. This approach is able to locate and track objects even against complex backgrounds such as dense foliage while the camera is moving. An object tracking in a sequence of sparse range images is suggested using the bounded Hough transform (BHT). This is a variation of the GHT which exploits the coherence across image frames. A method for rectangular object tracking is suggested in [109]. First, it finds the edges using the canny edge detector, and then applies the HT to find all lines in a predefined window surrounding the tracked object. A spatial color histogram is proposed in [58] for segmentation and object tracking. The spatial color histogram model encodes the color distribution and the spatial information. The GHT is used as a location estimator, which estimates the object location from a frame to another using its voting capabilities.

F. Underwater Applications.

A method for geometrical shape recognition for an underwater robot images is proposed [12]. The recognition problem is transformed into a bounded error estimation problem compared to the classical GHT. An underwater system is developed to carry out visually guided tasks on an autonomous underwater vehicle (AUV). Underwater pipelines are detected using the HT.

Then, the binary morphology is employed for determining their orientations. The HT and an AUV are used for cable tracking. Additionally. a micro remotely operated vehicle (micro-ROV) is used to test the net integrity in fish cages by analyzing the frames of a video camera exploiting the PPHT.The PHT is used to locate underwater pipelines. It searches for line segments among the edges and returns many line segments per pipe outline. The Hough pruning algorithm merges the semi-collinear lines. a model is suggested to detect and track underwater pipelines in complex marine environments. The images are filtered to reduce noise and to be enhanced. Then, the HT is used to get the boundary of the pip lines parameterized as curves. The curves with extracted features are used for estimating the locations and orientations of the pipelines. A marine tracking system is proposed in where the automatic detection and tracking of man-made objects in subs ea environments are pursued under the circumstances of poor visibility and marine snow. The proposed system involves minimizing the noise and video artifacts, estimating the camera motion, detecting the line segments and tracking the targets.

V. CONCLUSION

Line detection is an important task in the field of image processing. Line detection has lot of applications in the field of military and other important areas. So fast and accurate line detection is required. This paper evaluates the detailed study of different techniques of Line Detection. Here present a convolution based technique which produces an image description of the thin lines in an input image. Note that the Hough transform can be used to detect lines; however, in that case, the output is a parametric description of the lines in an image. The Hough transform (HT) is named after Paul Hough who patented the method in 1962. It is a powerful global method for detecting parameterized boundaries or curves. It transforms between the Cartesian space and a parameter space in which a straight line (or any parameterized curve) can be defined. The advantage of the Hough transform is that pixels lying on one line need not all be contiguous. This can be very useful when trying to detect lines with short breaks in them due to noise, or when objects are partially occluded.

References

[1] Sheng Liu,Charles F. Babbs and Edward J. Delp. "Line Detection Using A Spatial Characteristic Model". National Institutes of Health, Grant NumberCA62243, and a fellowship from the Purdue Cancer Center.

[2] J. Canny, "Line Detection by Hough transformation " IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, pp. 679-698, 1986

[3] G.T. Shrivakshan, Dr.C. Chandrasekar, "A Comparison of various Edge Detection Techniques used in Image Processing" IJCSI Inter- national Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, September 2012

[4] Er. Komal Sharma, Er. Navneet Kaur "Comparative Analysis of
Various Edge Detection Techniques" IJARCSSE Volume 3, Issue
12, December 2013

[5] Richard O. Duda and Peter E. Hart, “Use of the hough transfor- mation to detect lines and curves in pictures” Commun. ACM,
15(1):11–15, January 1972.

[6] C. Galambos, J. Kittler, and J. Matas , “Progressive probabilistic hough transform for line detection”Computer Vision and Pattern Recognition, IEEE Computer Society Conference on, 1:1554, 1999.

[7] R. O. Duda, and P. E. Hart, "Use of the Hough Transformation to
Detect Lines and Curves in Pictures”. Comm. ACM, Vol. 15, No.1, 1972, pp. 11-15

[8] J. Illingworth , and J. Kittler, "A Survey of Efficient Hough Trans- form Methods", in Alvey Vision Conference (AVC), 1987, pp. 319 – 326, doi:10.5244/C.1.43

[9] W. Niblack, and D. Petkovic, "On Improving the Accuracy of the Hough Transform: Theory, Simulations, and Experiments", in IEEE Conference on Computer Vision and Pattern Recognition, 1988, pp.
574- 519

[10] R.J. Christopher, and P.H. Gregson, "Detecting Corners Using the
‘Patchy’ Hough Transform", in Conference on Electrical and Com- puter Engineering, 1994, pp. 576- 579.

[11] Y. Zhang, and R. Webber, "A Windowing Approach to Detecting
Line Segments Using Hough Transform", Pattern Recognition, Vol. 29, No. 2, 1996, pp. 255-265

[12] Allam Shehata Hassanein, Sherien Mohammad, Mohamed Sameer, and Mohammad Ehab Ragab "A Survey on Hough Transform, Theory, Techniques and Applications " IJCSI International Journal of Computer Science Issues, Volume 12, Issue 1, No 2, January 201