Comparative Analysis Of Image Segmentation Using Hough Transform

Pramod Kumar Pandey¹, Nitin Khurana², and Ashish Aggarwal³, Amit Manocha⁴

¹ Department of E&I, ITM University, Gwalior (M.P.), ²Department of ICE, N.I.T, Jalandhar (Pb.), ³ Department of EE, K.I.T.M., Kkr (Hr.), Department of EE, GIMT, Haryana

Abstract: Image segmentation, a way of extracting and representing information from an image is to group pixels together into regions of similarity. We would group pixels together according to the rate of change of their intensity over a region or the rate of change of depth in the image, corresponding to pixels lying on the same surface such as a plane, cylinder, sphere etc.

Hough has proposed an interesting and computationally efficient procedure for detecting lines in pictures. In this paper we point out that the use of angle- radius rather than slope-intercept parameters simplifies the computation further. We also show how the method can be used for circle detection.

The objective of this application is the recognition of different shapes in an image. This task can be subdivided into following procedures. First, an image is converted into gray scale. For detecting lines in images, we used histogram for the intensity related information and threshold. Then, an edge recognition procedure is implemented. We use the first-derivative Sobel detector to determine the edges and edge directions. Then, a Hough transform is accomplished on the threshold edge map for linking the edges. This is the procedure to detect straight line. For circle detection we wrote a function called “houghcircle”.

1.1 SEGMENTATION

In computer vision, segmentation refers to the process of partitioning a digital image into multiple regions (sets of pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images.

The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region is similar with respect to some characteristic or computed property, such as colour, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).

Segmentation algorithms generally are based on one of 2 basis properties of intensity values.

Discontinuity: to partition an image based on abrupt changes in intensity (such as edges)
Similarity: to partition an image into regions that is similar according to a set of predefined criteria.

For intensity images (i.e. those represented by point-wise intensity levels) four popular approaches are: threshold techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation methods.

Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. Because spatial information is ignored, however, blurred region boundaries can create havoc. Edge-based methods center around contour detection: their weakness in connecting together broken contour lines make them, too, prone to failure in the presence of blurring.

A region-based method usually proceeds as follows: the image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving perhaps homogeneity or sharpness of region boundaries. Over stringent criteria create fragmentation; lenient ones overlook blurred boundaries and over merge. Hybrid techniques using a mix of the methods above are also popular. A connectivity-preserving relaxation-based segmentation method, usually referred to as the active contour model, was proposed recently. The main idea is to start with some initial boundary shape represented in the form of spline curves, and iteratively modifies it by applying various shrink/expansion operations according to some energy function. Although the energy minimizing model is not new, coupling it with the maintenance of an elastic contour model gives it an interesting new twist. As usual with such methods, getting trapped into a local minimum is a risk against which one must guard; this is no easy task.

1.2 EDGE

An edge is seen at a place where an image has a strong intensity contrast. Edges could also be represented by a difference in color, without any difference in intensity. Detecting such edges goes beyond the scope of this introduction. Of course there are exceptions where a strong intensity contrast does not embody an edge. Therefore a zero crossing detector is also thought of as a feature detector rather than a specific edge detector.

1.2.1 Edge Detector:

Edge detection is a very important area in the field of Computer Vision. Edges define the boundaries between regions in an image, which helps with segmentation and object recognition.

They can show where shadows fall in an image or any other distinct change in the intensity of an image. Edge detection is a fundamental of low-level image processing and good edges are necessary for higher level processing. The problem is that in general edge detectors behave very poorly. While their behavior may fall within tolerances in specific situations, in general edge detectors have difficulty adapting to different situations. The quality of edge detection is highly dependent on lighting conditions, the presence of objects of similar
intensities, density of edges in the scene, and noise. While each of these problems can be handled by adjusting certain values in the edge detector and changing the threshold value for what is considered an edge, no good method has been determined for automatically setting these values, so they must be manually changed by an operator each time the detector is run with a different set of data. Since different edge detectors work better under different conditions, it would be ideal to have an algorithm that makes use of multiple edge detectors, applying each one when the scene conditions are most ideal for its method of detection. In order to create this system, you must first know which edge detectors perform better under which conditions. That is the goal of our project. We tested four edge detectors that use different methods for detecting edges and compared their results under a variety of situations to determine which detector was preferable under different sets of conditions. This data could then be used to create a multi-edge-detector system, which analyzes the scene and runs the edge detector best suited for the current set of data. For one of the edge detectors we considered two different ways of implementation, one using intensity only and the other using color information.

2.2 EDGE DETECTION TECHNIQUES
There are many ways to perform edge detection. However, the majority of different methods may be grouped into two categories:

✓ **Gradient**: The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.

✓ **Laplacian**: The Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location.

Suppose we have the following signal, with an edge shown by the jump in intensity below:

![Fig 1.1 Intensity graph of a signal](image)

If we take the gradient of this signal (which, in one dimension, is just the first derivative with respect to \( t \)) we get the following:

![Fig 1.2 First derivative of the signal](image)

Clearly, the derivative shows a maximum located at the center of the edge in the original signal.

This method of locating an edge is characteristic of the "gradient filter" family of edge detection filters and includes the Sobel method. A pixel location is declared an edge when the value of the gradient exceeds some threshold. As mentioned before, edges will have higher pixel intensity values than those surrounding it. So once a threshold is set, you can compare the gradient value to the threshold value and detect an edge whenever the threshold is exceeded. Furthermore, when the first derivative is at a maximum, the second derivative is zero. As a result, another alternative to finding the location of an edge is to locate the zeros in the second derivative. This method is known as the Laplacian and the second derivative of the signal is shown below:

![Fig 1.3 Second derivative of the signal](image)

2.2.1 SOBEL OPERATOR:
The Sobel operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image.

In simple terms, the operator calculates the gradient of the image intensity at each point, giving the direction of the largest possible increase from light to dark and the rate of change in that direction. The result therefore shows how "abruptly" or "smoothly" the image changes at that point and therefore how likely it is that that part of the image represents an edge, as well as how that edge is likely to be oriented. In practice, the magnitude (likelihood of an edge) calculation is more reliable and easier to interpret than the direction calculation.

2.2.3 PREWITT'S OPERATOR:
Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images. The mask is:

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 0 \\
-1 & -1 & -1
\end{bmatrix}
\]

h1 = 0 0 0  h2 = −1 0 1
−1 −1 1  0 1  0

![Fig 1.4 Prewitt edge detector mask](image)
2.3 HOUGH TRANSFORM:

In Hough transform, the points are linked by determining first if they lie on the curve of specified shape. Unlike the local analysis method, where given n points of an image are taken into consideration. Suppose we want to find the subset of these points that lie on the straight lines.

One possible solution is to find all the lines determine by every pair of points and then find all subsets of points that are close to particular lines. The problem with this procedure is that it involves finding n (n-1)/2 ~ n^2 lines and then performing (n) (n (n-1)) ~ n^3 comparisons of every point to all lines. This approach is computationally prohibitive in all but in most the trivial applications.

The Hough transform is a method that, in theory, can be used to find features of any shape in an image. In practice it is only generally used for finding straight lines or circles. The computational complexity of the method grows rapidly with more complex shapes.

Assume we have some data points in an image which are perhaps the result of an edge detection process, or boundary points of a binary blob. We wish to recognize the points that form a straight line.

Consider a point \( (x_i, y_i) \) in the image. The general equation of a line is

\[
y = ax + b.\tag{1}
\]

There are infinitely many lines that pass through this point, but they all satisfy the condition

\[
y_i = ax_i + b\tag{2}
\]

For varying \( a \) and \( b \).

We can rewrite this equation as

\[
b = -x_i a + y_i\tag{3}
\]

And plot the variation of \( a \) and \( b \).

If we divide parameter space into a number of discrete accumulator cells we can collect ‘votes’ in a b space from each data point in x y space. Peaks in a b space will mark the equations of lines of co-linear points in x y space.

However, we have a problem with using \( y = ax + b \) to represent lines when the line is vertical. In that case \( a = \infty \) and our parameter space is unbounded (we would need a very large computer to store our parameter accumulator array!)

An alternative representation of a line is given by

\[
x \sin(\theta) - y \cos(\theta) = r
\]

Where \( r \) is the distance of the line from the origin and \( \theta \) is the angle between this perpendicular and the x-axis.

Our parameter space is now in \( \theta \) and \( r \), where \( 0 \leq \theta \leq 2\pi \) and \( r \) is limited by the size of the image. As before, peaks in the accumulator array mark the equations of significant lines.

ACCUMULATOR

Finding the dominant lines in the image can now be reformulated as finding all the locations in parameter space where significant number of lines intersect. This is basically the goal of Hough Transform.

In order to compute the Hough Transform, we must decide on a discrete representation of the continuous parameter space by selecting an appropriate step size for the \( k \) and \( d \) axes. Once we have selected step sizes for the coordinates, we can represent the space naturally using an array. Since the lines intersect, it is called an accumulator array. Each parameter space line is painted into the accumulator array and the cells through which it passes through are incremented so that ultimately each cell accumulates the total number of lines that intersect at each cell (Fig. 1.5).

2.3.2 BASIC ALGORITHMS

- Compute the gradient of an image and threshold it to obtain a binary image.
- Specify subdivisions in the \( \rho \theta \)-plane.
- Examine the counts of the accumulator cells for high pixel concentrations.
- Examine the relationship (principally for continuity) between pixels in a chosen cell.
- Based on computing the distance between disconnected pixels identified during traversal of the set of pixels corresponding to a given accumulator cell.
- A gap at any point is significant if the distance between that point and its closest neighbor exceeds a certain threshold.
CIRCULAR HOUGH TRANSFORM
The general Hough transform can be used on any kind of shape although the complexity of the transformation increase with the number of parameters needed to describe the shape. In the following we will look at the Circular Hough transform (CHT).

2.4.1 PARAMETEER REPRESENTATION
The general Hough transform can be described as a transformation of a point in the x-y plane to the parameter space. The parameter space is defined according to the space of the object of interest.

The circle is simpler to represent in parameter space, compared to the line, since the parameter of the circle can be directly transfer to the parameter space.

The equation of the circle is:
\[ r^2 = (x-a)^2 + (y-b)^2 \]

As it can be seen the circle to get three parameter \( r, a \) & \( b \), where \( a \) & \( b \) are the centre of the circle in the direction \( x \) & \( y \) respectively and \( r \) is the radius.

The parameter representation of the circle is:
\[
\begin{align*}
    x &= a + r \cos \theta \\
    y &= b + r \sin \theta
\end{align*}
\]

Thus the parameter space for a circle will belong to \( R^3 \) whereas the line only belonged to \( R^2 \).As the number of parameter needed to describe the shape increase as well as the dimension of the parameter space \( R \) increase so do the complexity of the Hough transform. There for simple shape with parameter belonging to \( R^2 \) or at most \( R^3 \). In order to simplicity the parametric representation of the circle, the radius can be held as a constant or limited number of known radii.

ACCUMULATOR
At each edge point we draw a circle with centre in the point with the desired radius. This circle is drawn in the parameter space, such that our x-axis is the a-value and y-axis in the b-value and z-axis is the radii. At the coordinates which belongs to the parameter of the drawn circle. We increment the value in our accumulator matrix which essentially has same size as parameter space. In this way we sweep over energy edge point in the input image drawing circle with the desired circle with desired radii and incrementing the value in our accumulator. When every edge point and every desired radius is used, we can turn our attention to accumulator will now contain numbers corresponding to the number of circles passing through the individual coordinate. Thus the highest number corresponds to the circle of the circle in the image.

Figure 1.7: The parameter space used for CHT

Figure 1.8: A Circular Hough transform from the x,y–space (left) to the parameter space (right), this example is for a constant radius

ADVANTAGES
One important difference between the Hugh Transform and other approaches is resistance of the former to noise in the image and its tolerance towards holes in the boundary line. Figures 1 and 2 compare the Hugh transform of a plain straight line with a dotted one. As can be seen there is almost no differences in the results.

Figure 1.9 Straight line and its Hough transform.

Figure 1.10 Straight dashed line and its Hough transform.

COMPARISON OF DIFFERENT EDGE DETECTOR

SOBEL EDGE DETECTOR

Figure 1.11 Sobel image
DETECTION PROCESS

First, we made an object in wood which contained triangle, square, parallelogram & circle. The inner side of this shape is painted in black colour so that the shadow will not come in the picture. For better quality and to reduce the other effect, the image has been taken by a digital camera with white background. The image is shown in fig 1.16.

To know the intensity level of the image, the histogram in bar form of the image has been found (fig 4.8). It gives the information about the background and object and how pixels have gray levels grouped into two dominant modes.

For recognizing edge and edge direction, 'sobel' function is used (fig 4.9). It gives the binary image with discrete point at the edges and where the intensity level changes. Points are present on every where including the edges. To clear the edges, other points should be minimizing. To minimize the point the threshold value is kept [0.9 0.9].
In binary image the points are discrete. So the points may or may not be in the same line. To detect the co-linearity of the points the ‘Hough’ function in MATLAB is made and implemented. It gives the information that how many points are co-linear to each other.

CONCLUSION
Image segmentation for a sample image is presented based on the Hough transform. This method detects both straight line and circle. When we are detecting circle, it is necessary to give the minimum and maximum value of the radius of the circle i.e. range of the circle. So it is required to have some knowledge about the radius. This method cannot be applied with unknown radius. One more problem arises when it detects the corner portion of an image if the corner is semi-circle. It detects semi-circle as circle. Though is circle are small, this can be avoided by taking the radius value large.

FUTURE SCOPE
From the above analysis, it can be suggested this method is not efficient for circle detection where there is no information about the radius of the circle. The future work will be to develop a algorithm using hough transform for circle detection with unknown radius and detect only the full circle.

REFERENCES

