A CASE STUDY: EDGE DETECTION TECHNIQUES USING HOUGH TRANSFORM AND CANNY EDGE ALGORITHM

Deepika Rani Sona, Prayline Rajabai. C, Debopam Dey, Akanksha Jain
School of Electronics Engineering, VIT University, Vellore, Tamil Nadu, India

Rashmi Ranjan Das
School of Electrical Engineering, VIT University, Vellore, Tamil Nadu, India

Stephen Olatunde Olabiyisi
Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria

ABSTRACT

Edge Detection is an ever-expanding frontier in the field of Digital Image Processing. Several research papers have been published in this domain in the past decade specifying many applications of edge detection methods using platforms like MATLAB or OpenCV like Hough, Canny, Sobel, Prewitt Transforms among many others. In this case study, various elaborations on basic methods of edge detection have been made and their applications have been discussed in real-time. Furthermore, the performance analysis is figured out by comparison between the results of Canny Edge Detection method and Hough Transforms.

Keywords: Edge Detection, Canny Edge Detection Algorithm, Computation Time, Hough Transform, Memory Mapping.


http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=8&IType=10

1. INTRODUCTION

Paul Hough suggested an efficient method for detecting lines and edges in binary images in 1962 [1]. Hough Transform (hereinafter referred as HT) converts a global detection problem in the image space into an easier local peak detection problem in the parameter space. The working algorithm of HT is explained in fig 1 which uses the slope-intercept parameterization and the voting scheme.

http://www.iaeme.com/IJMET/index.asp
1) Building the parameter space with an appropriate quantization level for any given line of slope $m$ and intercept $c$.
2) Create an accumulator array denoted as $A(m, c)$.
3) Set $A(m, c) = 0 \forall (m, c)$
4) Extract the edge of the images using Canny Detector.
5) For each pixel on the image edges:
   
   $$(x_i, y_i) \forall (m_k, c_l)$$

   Verifying equation: $c_l = -x_im_k + y_i$;
   Increment: $A(m_k, c_l) = A(m_k, c_l) + 1$
6) Find the local maxima $A(m, c)$ and in that indicate the lines in the parameter space.

For the above algorithm, all edge pixels lying on a line plays a role in increasing the content of the cell in the accumulator array which corresponds to the real slope and intercept of that line. This suggests that every corresponding straight line does really exist in the image. Thus, we can say that the HT can be considered as a vote-counting-procedure where each edge point votes for all the parameter combinations which could have possibly resulted in its formation. Fig. 1 represents a straight line with slope 1 and intercepts 3 and the corresponding pixel values of the image formed by it.

**Figure 1** A line in image space is represented as a cell: $(m=1, c=3)$ in parameter (Hough) space.

Figure 2 shows various images and cases for point-to-curve transformation. Here, we obtain degenerate solutions when the input is a vertical line or nearly vertical line. Thus, the slope approaches to infinity. This is the stimulus which suggests the $(\rho, \theta)$ parameterization in [2]. The parameter $\rho$ is the perpendicular distance from the origin to the line, and the parameter $\theta$ is the angle between the perpendicular and the horizontal axis. In fact, this is a point-to-curve transformation using the normal representation of the line:

$$xcos\theta + ysin\theta = \rho$$

All the edge points are independently treated in HT; which makes the parallel processing of all points possible which leads to its real-time applications. Also, all the partially deformed and noisy shapes can be handled very easily due to its voting scheme. Moreover, the multiple occurrences of lines can be detected very easily since each occurrence has its unique cell address in the parameter space. HT can be extended to detect shapes other than lines. All the above-mentioned points give HT an edge over any other edge detection methods.

**Figure 2** A point-to-curve transformation in $(\rho, \theta)$ parameterization
Moreover, the HT has certain drawbacks. It requires a high computational demand and a large storage area. However, it can be increased by reducing the size of the accumulator array. Another edge detection technique is Canny Edge Detection Algorithm (hereinafter referred as CEDA) which was developed by John F. Canny in 1986. According to Canny, an optimal edge detection filter should maximize the product of localization criteria and detection with the restriction in low spurious inputs.

The main advantage of canny edge detection method is that the algorithm is adaptable to various environments. Its parameters allow it to be modified to the edges of differing characteristics and figures depending on the requirements from the end user in any given implementation or application.

The remaining paper is arranged and organized as follows: Section II gives us the methodology followed by both the edge detection techniques separately. Section III contains the shape detection aspects of any image. Section IV presents the speedup and memory saving techniques in both the algorithm. Section V deals with the experimental results obtained in MATLAB. Section VI focuses on all possible real-time applications. Section VII has the future work related to this article and Section VIII concludes the article.

2. METHODOLOGY

Hough Transform

Figure 3 shows the architecture of the proposed circle detection algorithm. The algorithm consists of two modules; an edge detector and a voting processor. First, in the edge detector module, we apply the Ball Detection Algorithm [4] to obtain edges from grayscale images. To efficiently handle edges, we utilize double-buffered internal memory blocks to store the edge list (hereinafter referred as EL). Next, for each edge pixel, the voting processor constructs a circular cone, as described in [5], in Hough space with adaptive circular resolution using pre-calculated trigonometric values based on the standard circle Hough transform. Finally, relatively high voter circles are chosen from the circular cone. To minimize the high-cost numerical resources, the proposed edge detector module is mainly implemented based on the ball detection algorithm [4] for background subtraction.

The trigonometric function values are indexed by $n \times dt_\theta$ ($\theta$ interval in the trigonometric function table), as depicted in Fig4. Since we use eight sample points in a pixel, more than eight $V Es$ for a pixel may not have a good effect on the weight calculation. Thus, $dt\theta$ can be adaptively decided by the following equation:

![Figure 3 Block Diagram of the Edge Detection System](http://www.iaeme.com/IJMET/index.asp)
\[ dt\theta = \begin{cases} \frac{128}{2^{k+1}}, & k \geq 1 \\ 1, & k < 1 \end{cases} \]

\(k\) can be calculated by simple shift operators without complex dividers and multiplexers since the dividing and multiplexing operations in the equation are binary operations.

Figure 4 shows the architecture of the proposed voting processor. First, the Edge fetch unit reads the edge information from EL and the quad-vectored edge generator modules in the Rasterizer Unit produce vectored edges in parallel using pre-calculated values obtained from the trigonometric table. Thus, each module processes only 128 divided by 4 times with the range of 0° ∼ 21.8°, 22.5° ∼ 44.3°, 45.0° ∼ 66.8°, and 67.5° ∼ 89.3°.

The coordinate values between 90° and 359.3° can be simply calculated with addition/subtraction operators from the results of the modules because they are all symmetric by the x-axis or y-axis, or both. Next, render modules set the flag bits based on the edge flag algorithm [5]. They then accumulate and calculate the weight value of each pixel as described in [6].

Canny Edge Detection Algorithm

Canny Edge Detection is one of the simplest methods used in shape recognition. This method uses 4 stages for edge detection which are mentioned as follows [3]:

- Image smoothing
- Calculating edge strength and edge direction.
- Directional non-maximum suppression.
- Invoking threshold with Hysteresis.

The output is an image containing only the edges detected through these processes.
Figure 5 Steps involved in Canny Edge Detection Algorithm

Image smoothing is performed by convolving the input image with a Gaussian filter. This process is used to reduce the noise within an image or to produce a less pixilated image. The output obtained is a blurred intermediate image. This blurred image is input to the next block to calculate the edge strength and direction of the edges.

The final output of this stage is a blurred image obtained from the image smoothing stage which is convolved with a 3x3 Sobel operator. The Sobel operators are used to obtain a gradient image.

\[ G_x = (I * g_x) \]
\[ G_y = (I * g_y) \]

The edge strength G is given by:

\[ G = \sqrt{G_x^2 + G_y^2} \]

Edge direction is defined as the direction of the tangent to the contour that the edge defines in 2-dimensions.

\[ A = \arctan \left( \frac{G_x}{G_y} \right) \]

It is a process in which all pixels whose edge strength is not maximal are marked as zero within a certain local neighborhood. The non-maximum suppression is used to suppress all the image information which is not the local maxima. For each pixel in the linear window, edge strength of the neighbors is compared, and if the pixels are not part of the local maxima then they are set to zero. Results of this stage give us two threshold values, T1 (high) and T2 (low).

Edge strength which is greater than T1 is considered as a definite edge. Edge strength which is less than T2 is set to zero. The pixel with edge strength between the thresholds T1 and T2 is considered only if there is a path from this pixel to a pixel with edge strength above T1.

3. SHAPE DETECTION

Standard Hough Transform (hereinafter referred as SHT) is used for identification of lines. However, our daily based objects are irregular in shape. Therefore, over the years, several variations have been suggested to identify and analyze irregular shapes. In this section, we will demonstrate HT in various irregular shapes.

Analytical Shapes

The original use of the HT is detecting lines, however; it took years of research to identify other analytical shapes such as circles and ellipses. For an ellipse detection, as shown in fig. 6, five parameters are defined, its center \((x_c,y_c)\), the major axis \(a\), the minor axis \(b\), and the slope \(\theta\).
Therefore, an ellipse is detected in 5-D parameter space. A method is presented in [7] for detecting ellipses by finding symmetric lines with the help of the linear Hough transform (LHT). The image is pre-processed by performing edge detection process in the case of concentric circles, and then the circle centers are found using the gradient HT. Finally, the radii are found using the one-dimensional (1D)-HT. The detection efficiency is enhanced by the image discretization, and by reducing the resolution ratio in the process of circle center detection. A sphere detection method based on the hierarchical HT (HHT) is presented in [7]. The HHT has its added advantage of reducing storage space and being robust with respect to noise and being applicable to 3D images.

Irregular Shape Detection

Various HT such as SHT and CHT are used for detecting analytical curves using suitable parameterization. However, if the shape does not have an analytical form, we need to resort the parameterization using the generalized Hough transform (GHT) presented in [8]. This transform uses a look-up table defining the relation between the positions and orientations, and the Hough parameters for a parametric equation. Fig.7 shows the parameters of a GHT. The advantages of the GHT include being tolerant to noise and being able to find multiple occurrences of a shape during one pass. The main problems of it are the high storage and computation demands.

The generalized algorithm for edge detection techniques is given as below:

**Constructing Look-up Table of Model (usually once and off-line):**

1) Extract edges of the model image e.g. using Canny detector.
2) Choose a point inside edges as a reference, \((x_{ref}, y_{ref})\).
3) For each edge point, \((x_i, y_i)\):
   1.1. Compute angle between the x-axis and slope direction of contour, \(\phi_i\), and displacement vector to reference, \(r_i\).
   1.2. Quantize \(\phi_i\) according to required precision.
   1.3. Computing distance to reference, \(\|r_i\|\), and angle \(\alpha_i\) between x-axis and \(r_i\).
   1.4. Storing \(\|r_i\|\) and \(\alpha_i\) in table in the row of quantized \(\phi_i\).

**Searching for Model in New Images (assuming same model scale and orientation, equivalent to searching for the reference point):**

1) Extract the edges of the images using any method like Canny detector or Sobel Transform.
2) Create the Accumulator Array $A[(x, y)]$, for all the possible reference points

3) For each edge point, $(x_i, y_i)$:
   1.1. Compute $\phi_i$, quantize $(\alpha_i, l_i)$, and store all the data in a row-wise manner in the look-up table.
   1.2. For each pair $(\alpha_i, l_i)$, compute reference candidates:
      
      $x_{cand} = x_i + l_i \cos(\alpha_i)$
      $y_{cand} = y_i + l_i \sin(\alpha_i)$
   1.3. Increase candidate votes:
      $A[(x_{cand}, y_{cand})] = A[(x_{cand}, y_{cand})] + 1$

4) For local maxima in $A[(x, y)]$, $(x_{ref} = x, y_{ref} = y)$

![Figure 7 Parameters of GHT](image)

**4. SPEEDUP AND MEMORY MANAGEMENT TECHNIQUES**

As stated earlier, the main drawback of HT is long computation times and large data storage requirement. A lot of research is going on and have been done to solve the mentioned drawbacks. These efforts range from software algorithmic changes like voting acceleration to hardware implementations like passing by parallel computing, and memory reduction. The Field Programmable Gate Array (hereinafter referred as FPGA) is among the most common hardware accelerators which is used in real-time implementations of the HT which an alternative solution for hardware acceleration to give tremendous computing performance.

As a method for straight line detection, the edge feature image is converted into a new domain by the classical HT and this new domain is called the Hough Parameter Space. By mapping the parameter points to image space the computational result in Hough space can be determined. Each point in Hough space represents a line in the initial image. The performance of accuracy and robustness of HT are limited because of the need to store a lot of calculation parameters.

Moreover, the FPGA is used to store the Hough parameters in its memory block. This approach is not only memory efficient but also can also be used for both line and circle detection. The hardware resources are minimized in the implementaion of the HT due to exploiting both the angle level and pixel-level parallelism [9]. Various other algorithms have been used to accelerate the HT by skipping the unnecessary computations and memory accesses.

There are two modules which result in the formation of FPGA:

1) The first one is an optimized Canny edge detector with enhanced non-maximum suppression which is used to extract the candidate edge pixels for subsequent accelerated computations.
2) The other one is a multi-level Pipelined Parallel Hough transform (PP-HT) [10] for line detection.
The block diagram of a simple multi-leveled PP-HT architecture is shown in fig. 8.

![Block Diagram of a multi-leveled Pipelined Parallel HT](image)

**Figure 8** Block Diagram of a multi-leveled Pipelined Parallel HT

5. EXPERIMENTAL RESULTS

Extracting the edges of objects in an image is of great importance in industrial applications of image processing. Many edge detectors have been proposed such as Canny, Robert, Prewitt, and Driche algorithms. Among these, the Canny edge detector and Hough transform have been widely successful in extracting the edge feature from an image due to their simplicity and accuracy. Here we compare Canny edge detection method and Hough transform. Fig. 9(a) and fig. 9(b) are the input images taken by us which shows the pattern and fringes formed by a LASER source and any irregular shape as shown below:

![Input Images](image)

**Figure 9** Input Images

When applying Hough Transform and Canny Edge Detection algorithm to detect the LASER fringes and the irregular shape separately, the following parameters were observed which are tabulated in Table 1 and Table 2 respectively. Fig. 10(a) and fig. 10(b) shows the Hough Transform and Canny Edge Detection algorithm used in Fig. 9(a) respectively and its parametric analysis are summarized in Table 1:

![Hough Transform and Canny Edge Detection Algorithm applied on Figure 9(a)](image)

**Figure 10** Hough Transform and Canny Edge Detection Algorithm applied on Figure 9(a)
Table 1 Parametric Analysis and Comparison of Figure 10

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Parameters</th>
<th>HT</th>
<th>CEDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Euclidean Distance</td>
<td>0.0648</td>
<td>0.0324</td>
</tr>
<tr>
<td>2.</td>
<td>Peak SNR (PSNR)</td>
<td>30.42</td>
<td>28.27</td>
</tr>
<tr>
<td>3.</td>
<td>Average Error</td>
<td>656.9433</td>
<td>645.2516</td>
</tr>
</tbody>
</table>

Similarly, Fig. 11(a) and Fig. 11(b) shows the Hough Transform and Canny Edge Detection algorithm used in Fig. 9(b) respectively and its parametric analysis are summarized in Table 2.

Table 2 Parametric Analysis and Comparison of Figure 11

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Parameters</th>
<th>HT</th>
<th>CEDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Euclidean Distance</td>
<td>0.0665</td>
<td>0.0134</td>
</tr>
<tr>
<td>2.</td>
<td>Peak SNR (PSNR)</td>
<td>29.22</td>
<td>28.26</td>
</tr>
<tr>
<td>3.</td>
<td>Average Error</td>
<td>644.9711</td>
<td>632.1324</td>
</tr>
</tbody>
</table>

After comparing various parameters like Average Error, Peak Signal to Noise Ratio (PSNR) and Euclidian Distance for the laser output fringes and the irregular shape, we observed that Average Error, PSNR, and Euclidian Distances are less in the case of Canny Edge Detection Algorithm for both the cases. Thus, we can conclude that Canny Edge Detection Algorithm is more efficient as compared to Hough Transform.

6. REAL-TIME APPLICATIONS
The real-time applications of various edge detection methods are given as follows:

Medical Applications
Low back pain is a major concern in people nowadays and diagnosis of this problem can be very difficult. For motion studies, Digital Video Fluoroscopy (DVF) technique was commonly used to obtain images, but the images obtained would often be blur due to noise. Therefore, determining the position of the vertebrae was a major challenge. In this new approach presented in [11], they initially use phase congruency to remove threshold Selection difficulty in the edge detection. Then to determine the moving vertebrae, our new Hough transform approach is applied. In laparoscopic Operations, a tracking system of surgical instruments is used which is clearly presented in [12].
**Industrial and Commercial Applications**

There are a countless number of industrial and commercial applications of the HT. Some of them include optical remote sensors onboard unmanned aerial vehicles (UAVs) which capture the spatial information which has great potential in the automatic surveillance of electrical power infrastructure. A challenging task is the detection of power lines from the cluttered background and as a remedy, an inspection system and a knowledge-based power line detection method for a vision based UAV surveillance is presented in [13]. A PCNN filter is used to remove background noise from the images and Hough transform to detect straight lines.

Steel is the material used for numerous industrial applications and its Surface properties are the most important parameters. Traditional inspection procedures are inadequate to ensure guaranteed quality-free surface. To ensure precise requirements of customers, automated vision-based steel surface inspection techniques have been found that uses the HT technique which is used to check the clamps, holes, and welds. The use of CHT and the LHT have been shown in [14] for the detection of the botanical and mineralogical alterations that results from the natural seepage of carbon dioxide and light hydrocarbons. To differentiate the holes from the background CHT is used and then LHT is used to found the collinear centers of the detected circles.

**Underwater Applications**

An automatic method to detect buoy of geometric shapes in underwater robot images is proposed in [15]. The identified problem is changed into a bounded-error estimation problem and the results are then compared with the classical GHT. Underwater pipelines can also be detected using the HT.

**Bio-metric and Man-Machine Interaction**

A generic hand detection method is presented in [16]. In this, the lines are detected using PHT and then they are matched with the registered model. A real-time human-robot interaction system is proposed in [17] which is based on the hand gestures and tracking. It combines the Connected Component Labeling (CCL) and the HT technique. It can extract the center of the hand, its directions and the fingertip positions of all outstretched fingers through the detection of the skin color.

An eye detection approach is presented in [18] using Circular Hough transforms technique for detection of a person. The search of eye pair depends on the circular shape of the eye in the 2D image and detection accuracy of this proposed method is about 86%.

**3-D Applications**

The identification of 3D objects is of great use and yet a challenging task in computer vision. A method that uses rays instead of points to vote for object reference points is proposed in [19]. The usage of Ray voting allows a clustering of votes showing in the same direction, to a single vote with a suitable vote weight.

**Object Recognition**

A method for extracting invariant features from images is presented in [20]. The features are invariant to image scale and rotation. The paper also describes an approach of using these features for object recognition. The recognition takes place by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform and finally performing verification through the least-squares solution.
7. FUTURE SCOPE
We have seen several applications of edge detection methods. The applications of edge detection methods can also be extended to security domains as well. An effective technique for personal authentication on noisy iris recognition can be done using Hough Transform. The iris segmentation technique consist of six modules, namely determine the expected region of the iris; Apply the Canny Edge Detection method; Then to find the Cartesian parameters apply Circular Hough Transform (CHT) on the binary edge image; Localization of Upper eyelid; Localization of Lower eyelid; Isolate the specular reflections and remove the pupil region to make the IRIS recognition more accurate and in future advancing this technique a security analysis of the same hybrid mechanism on noisy irises can be found when the lower or upper eyelids and eyelashes cover the pupil of the iris.

8. CONCLUSION
Detection of circles was a very important and a challenging task for industrial applications before the discovery of various edge detection methods but after various edge detection methods have discovered this task was simplified to a great extent. Of all the detection methods Hough and Canny Edge Detection Algorithm are considered the best. If we want to find the true edges of the buildings, a canny edge detector cannot recover information very well, however, the Hough transform can detect some of the straight lines representing building edges even within obstructed region. As compared to Circular Hough Transform (CHT) Canny Edge Detection Algorithm is a very fast algorithm to detect circles, So Canny Edge Detection Algorithm is better as compared to Hough Transform in terms of speed but in terms of detection, CHT is better.

REFERENCES


