CHAPTER 4

Performance of Ant System over other Convolution Masks in Extracting Edge

4.1. Introduction

Computer vision aims to duplicate the effect of human vision by electronically perceiving and understanding an image. Giving computers the ability to see is not an easy task. Towards computer vision the role of edge detection is very crucial as it is the preliminary or fundamental stage in pattern recognition. To establish the boundaries of the objects in an image is to examine each pixel and its immediate neighborhood to determine whether that pixel is, in fact on the boundary of an object.

Pixels exhibiting the required characteristics are labeled edge points. So edges characterize object boundaries and are therefore useful for segmentation and identification of objects in a scene. The idea that the edge detection is the first step in vision processing has fueled a long term search for a good edge detection algorithm.

The main application of swarm intelligence techniques has been applied to solve combinatorial optimization problems. This chapter discusses work in-progress on the application of swarm intelligence ideas to image processing problem, such as extracting boundaries or edges of objects. The proposed algorithm present over here is an Ant Colony Optimization based mechanism to extract the edges in an image. Experimental results indicate that the proposed method is more efficient than the Gradient based edge detection techniques.

4.2. Extracting Edge From Images

An edge [5, 18, 25] is a jump in Intensity or otherwise it can be considered as a typical boundary between two dissimilar regions. An edge is not a physical entity, just like a shadow. It is where the picture ends and the wall starts. It is where the vertical and the horizontal surfaces of an object meet. It’s what happens between a bright window and a dark. Edges in images are areas with strong intensity contrasts.
4.2.1 The Edge Structure

If we look at the concept of a digital edge a little closer, an edge is a set of connected pixels that lie on the boundary between two regions. An ideal edge is a set of connected pixels, in the vertical direction, each of which is located at an orthogonal step transition in gray level. In practice the imperfections in image acquisition yield edges that are blurred, with the degree of blurring being determined by factors such as the quality of the image acquisition system, the sampling rate, and illumination conditions under which the image is acquired. Effects such as refraction or poor focus can result in objects with boundaries defined by a gradual change in intensity.

As a result, if we closely observe the cross section of the edge it is nothing but the shape of the ramp. An ideal edge is a ramp with an infinite slope. The slope of the ramp is inversely proportional to the degree of blurring in the edge. In this model, we no longer have a thin (one pixel thick) path. Instead, an edge point now is any point contained in the ramp, and an edge would then be a set of such points that are connected.

The thickness of the edge is determined by the length of the ramp, as it transitions from an initial to a final gray level. This length is determined by the slope, which, in turn, is determined by the degree of blurring. Blurred edges tend to be thick and sharp edges tend to be thin.
4.2.2 Edge Detection Categories

Though, a variety of edge detection Techniques are available, the most of them may be grouped into two categories [25], Gradient and Laplacian. The gradient method detects edges by looking for a maximum and minimum in the first derivatives of the images [18] ie, it assumes a local maximum at an edge. The laplacian method searches for zero crossing in the second derivatives of the image to find the edges [25].

In gradient method for a continuous image say \( f(x, y) \) we consider the two edge directions, horizontal and vertical represented by \( \partial_x f(x, y) \) and \( \partial_y f(x, y) \). The gradient vector points in the direction of maximum rate of change of ‘f’ at co-ordinates \( f(x,y) \). The important quantities in edge detection are the gradient magnitude denoted by equation 4.1

\[
\nabla f(x,y) = \sqrt{\left(\partial_x f(x,y)\right)^2 + \left(\partial_y f(x,y)\right)^2}
\]

(4.1)

and the gradient orientation (or) the direction of the gradient vector denoted as

\[
\alpha(x, y) = \tan^{-1}\left[\frac{\partial_y f(x,y)}{\partial_x f(x,y)}\right]
\]

(4.2)

where the angle is measured with respect to the x-axis. The direction of an edge at \( x, y \) is perpendicular to the direction of the gradient vector at that point. A pixel location is declared as an edge location if the gradient magnitude exceeds some threshold.
4.2.3 Threshold and edge linking

We are led to the idea that, to be classified as a meaningful edge point, the transition in gray level associated with that point has to be significantly stronger than the background at that point. Since we are dealing with local computations, the method of choice to determine whether a value is significant or not is to use a threshold. Thresholding is particularly a useful region-approach technique for scenes containing solid objects resting upon a contrasting background. Thus, we define a point in an image as being an edge point if its two-dimensional first-order derivative is greater than a specified threshold. A set of such points are connected according to a predefined criterion of connectedness.

It is important to note that these definitions do not guarantee success in finding edges in an image. They simply give us a formalism to look for them. The choice of threshold value determines the resulting segmentation and hence the perceived quality of the edge detector. It is useful to consider the cumulative histogram of the gradient image in selecting the appropriate threshold value. The location of all edge points constructs an edge map. The selection of the threshold value is an important design decision that depends on a number of factors such as image brightness, contrast, noise level etc...

A weak edge positioned between two strong edges is highly probable that this inter positioned weak edge should be a part of a resulting boundary. If, on the other hand, an edge (even a strong one) is positioned by itself with no supporting context, it is probably not a part of any border.
4.3 Edge Detection Techniques

Four frequently used methods are considered here for comparison. Edge detection operators [19, 26, 27] examine each pixel neighborhood and quantify the slope, and often the direction as well of the gray-level transition. The two characteristics of principal interest are the slope and the direction of that transition. There are several ways are available. Most of which are based upon convolution with a set of directional derivative masks.

4.3.1. The Sobel Detection

The Sobel operator performs a 2-D spatial gradient measurement on an image [19, 26, 27] and so emphasizes regions of high spatial gradient that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image. In theory at least, the operator consists of a pair of 3×3 convolution masks as shown in Figure 4.1. One mask is simply the other rotated by 90°. This is very similar to the Roberts Cross operator.

These masks are designed to respond maximally to edges running vertically and
horizontally relative to the pixel grid, one mask for each of the two perpendicular orientations. The masks can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation call these as $G_x$ and $G_y$. These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G| = \sqrt{G_x^2 + G_y^2}$$

(4.3)

Although typically, an approximate magnitude is computed using:

$$|G| = |G_x| + |G_y|$$

(4.4)

which is much faster to compute.

The angle of orientation of the edge relative to the pixel grid giving rise to the spatial gradient is given by:

$$\theta = \arctan(G_y/G_x) \mod \pi/4$$

(4.5)

In this case, orientation $\theta$ is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anticlockwise from this. Often, this absolute magnitude is the only output the user sees the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure 4.2 which is used to quickly compute approximate gradient magnitude.
Using this mask the approximate magnitude is given in equation 4.6

\[
|G| = |(P_2 + 2X P_5 + P_9) - (P_7 + 2X P_8 + P_3)| + |(P_6 + 2X P_9 + P_3) - (P_1 + 2X P_4 + P_7)|
\]

(4.6)

### 4.3.2 The Prewitt Detection

The Prewitt edge detection [26] is an appropriate way to estimate the magnitude and orientation of an edge. The Prewitt kernels (named after Judy Prewitt) are based on the idea of the central difference as in equation 4.7

\[
\frac{df}{dx} \approx \frac{[f(x + 1) - f(x - 1)]}{2}
\]

(4.7)

or for two-dimensional images it is defined as in equation 4.8

\[
\frac{\partial I}{\partial x} \approx \frac{[I(x + 1, y) - I(x - 1, y)]}{2}
\]

(4.8)

This corresponds to the following convolution kernel

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

To get the precise gradient magnitude, we must then divide by two. By rotating this 90 degrees, we get \(\partial I/\partial y\). These kernels are, however, sensitive to noise. The Prewitt convolution mask is as shown in figure 4.3.
4.3.3. The Roberts Detection

The Roberts operator [6, 19, 26, 27] performs a simple, quick to compute, 2-D spatial gradient measurement on an image. It thus highlights regions of high spatial gradient which often correspond to edges. In its most common usage, the input to the operator is a grayscale image, as is the output. It marks edge points only and it does not return any information about the edge orientation. This detection is simplest of all other edge detection operators and it will work best with binary images. Pixel values at each point in the output represent the estimated absolute magnitude of the spatial gradient of the input image at that point. In theory, the operator consists of a pair of 2×2 convolution masks as shown in Figure 4.4. One mask is simply the other rotated by 90°. This is very similar to the Sobel operator.

![Roberts Masks](image)

Fig 4.4: Roberts mask
These masks are designed to respond maximally to edges running at 45° to the pixel grid, one mask for each of the two perpendicular orientations. The masks can be applied separately to the input image, to produce separate measurements of the gradient component in each orientation call these \( G_x \) and \( G_y \). These can then be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is identified by the equation 4.3.

Although typically, an approximate magnitude is computed using the equation 4.4, which is much faster to compute. The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is calculated by equation 4.2. In this case, orientation \( \theta \) is taken to mean that the direction of maximum contrast from black to white runs from left to right on the image, and other angles are measured anticlockwise from this.

Often, the absolute magnitude is the only output the user sees and the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure 4.5 which is used to quickly compute approximate gradient magnitude

![Pseudo-Convolution mask](image)

**Figure 4.5 Pseudo-Convolution mask**

Using this mask the approximate magnitude is given by equation 4.9
4.3.4. The Kirsch Detection

Like other operators, in kirsch edge detection each point in the image is convolved with eight masks and the gradient is estimated in these eight possible directions. Each mask responds maximally to an edge oriented in a particular general direction. The convolution result of greatest magnitude indicates the gradient direction.

Operators approximating the first derivative of an image functions are sometimes called compass operators because of their ability to determine gradient direction. The mask value [19, 26, 27] is as shown in Figure 4.6 and other mask values can be created by doing simple rotation.

Fig 4.6 : Kirsch mask

4.4. The proposed Algorithm to extract edge using Mask

Algorithm Ordimg

Input:

i) An image

ii) Gray value of each pixel in an image stored as a matrix

Output:

i) Pixels that satisfies the above algorithm
ii) Image that contains only the edge formed by connecting all the above pixels

Begin

For each image pixel \((i, j)\)
For \(i = 1\) to \(n\) (pixels)
   For \(j = 1\) to \(n\) (pixels)
      Calculate the weighted average Gradient value
      Perform non maximal suppression
      Connect all the edge points to form the edge map
      Threshold these edges to eliminate insignificant edges
   end for \(j\)
end for \(i\)

end

4.5. The ACO Approach

Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems [7, 16]. In the early 1990's Ant Colony Optimization (ACO) was introduced by M. Dorigo and colleagues.

The inspiring source of ACO is the foraging behaviour of real ants. Initially ants have no idea of where food is in the environment, when searching for food, ants initially explore the area surrounding their nest in a random manner. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it wander back to the nest. During the return trip, the ant deposits a chemical substance called pheromone on the ground.
The pheromone deposited varies in quantity depending upon the quantity and quality of the food. This will guide other ants to the food source. The boundary is identified by considering the gray levels of nearest neighbors of the current position. The neighbors are identified from the current position by considering 8 connectivity as we did in the convolution mask methods. Each ant moves to an adjacent cell and reinforces the pheromone level on that spot. In order to move from state \( i \) to \( j \) the probability [7, 15, 16] is used as given in equation 4.10. The value of \( \tau_{ij} \) is used for moving to adjacent cell which is given in equation 4.11.

Where \( k \) is a constant. Similarly the factor \( \eta_{i,j} \) is given as in equation 4.12.

\[
P_{i,j}(t) = \frac{\tau_{ij}^{\alpha} \eta_{ij}^{\beta}}{\sum_{i \in \Omega} [\tau_{ij}^{\alpha} \eta_{ij}^{\beta}]} \quad \text{if } (i,j) \in \Omega \tag{4.10}
\]

The value of \( \tau_{ij} \) is used for moving to adjacent cell which is defined as follows.

\[
\tau_{ij} = k^+ \frac{\sigma}{k^+ \sigma} \tag{4.11}
\]

where \( k \) is a constant.

Similarly the factor \( \eta_{ij} \) is given as in equation 4.15.

\[
\eta_{ij} = \frac{v_m(I_{ij})}{v_{\text{max}}} \tag{4.12}
\]

where

\( I_{ij} \) is the current intensity value of pixel at \( i,j \).
$V_{\text{max}}$ is the maximum intensity variation between pixels in the whole image. It is calculated based on the 8 direction from the current pixel as shown in equation 4.13 and in figure 4.7.

$$V_{\text{max}}(I_{i,j}) = I_{i-1,j-1} - I_{i+1,j+1} | I_{i+1,j} - I_{i-1,j}$$

(4.13)

![8 directions table]

Fig 4.7 : 8 directions

When the ant moves from one pixel to another if that pixel falls on the edge then it should update the pheromone value of that pixel as given in equation 4.14.

$$P_{\text{update}} = P_{\text{init}} + \rho \frac{\nabla}{255}$$

(4.14)

where $\nabla$ is the difference between the median gray levels of previous cell and its neighbors and current cell and its neighbor.

### 4.6 Features

- The first and most obvious is low error rate. It is important that edges occurring in images should not be missed and there should be no response to pseudo edges.
- The second criterion is localization.
• The third criterion is to have only one response to a single edge.

   The simple threshold technique is used here to partition the image histogram by a single global threshold $T$, segmentation is then accomplished by scanning the image, pixel by pixel and labeling each pixel as edge point or not, depending on whether the gray level of that pixel is greater or less than the value of $T$.

4.7 The Proposed Algorithm to extract Edge using ACO

Algorithm ACOimg

Input:

i) Ordinary image and the Gray value of each pixel in an image stored as matrix

ii) Initial value of the parameters like $\alpha$, $\lambda$ etc and the Initial pheromone value

iii) $K$ – Ants and the Threshold value

Output:

i) Pixels that are selected by Ant

ii) Image that contains only the edge formed by connecting all the above pixels which were selected by Ant

Begin

*Initialize all the parameters*

*Initial pheromone*=Constant

*For each image pixel* $(I,j)$

*For iteration* $k = 1$ to $n $

*Repeat*
Get the pixel at $i, j$

Identify good solution or bad

If Good

Update Pheromone and other attributes

Else

Reduce Pheromone value

End If

Mark the pixel as visited

Until every $i, j$ in the image has been visited

End For

End for

Connect all the edge points to form the edge map

Threshold these edges to eliminate insignificant edges

End

The implementation of our algorithm is done using Visual C++. In VC++, the Microsoft Foundation Classes MFC is used which is an extensive C++ class library designed for creating Windows GUI (Graphical User Interface) programs. Convolution of the image with masks is used here. It is the most often used technique of calculating the derivative of an image in all directions. The idea is to take a 3X3 array of values and multiplying it point by point with a 3X3 section of the image. There are eight possible directions for each of an edge. The directional edge detector detects an edge in only one of the eight directions. If we want to detect only left to right edges we would use only one of eight masks. If however, we want to detect all of the edges, we would need to perform convolution over an image eight times using each of the eight masks.
There are two basic principles for each edge detector mask. The first is that the numbers in the mask sum to zero. If a 3X3 area of an image contains a constant value, then there are no edges in that area. The result of convolving that area with the mask should be zero. If the numbers in the masks sum to zero, then convoluing the mask with a constant area will result in the correct answer of zero. The second basic principle is that the mask should approximate differentiation or amplify the slope of the edge.

Many edge detectors have been designed using convolution mask techniques often using 3X3 mask sizes or even larger. An advantage of using a larger mask size is that local averaging within the neighborhood of the mask reduces errors due to the effects of noise.

4.8. Comparison On Edge Detectors

The relative performance of the gradient based edge detectors namely Sobel, Prewit, Roberts and Kirsch were compared with that of the Ant System. The performances of these methods on ten images were evaluated, of which the results of three sets are presented here.
Fig 4.8 Original Images
Fig 4.9 Sample Screen
Fig 4.10 Edges in Flower Vase
Sobel Detection

Prewitt Detection

Kirsch Detection

Roberts Detection

Fig 4.11 Edges in Tower
Fig 4.12. Edges in Lart
Fig 4.13. Edges using Ant System
4.9. Conclusion

Subjective analysis reveals that the new approach using Ant System of edge detection is effective in all the three categories of the images selected. Edge detecting in an image significantly reduces the amount of data and filters out useless information while presenting the important structural properties in an image. Edge detection is difficult in noisy images since both the noise and the edges contain high frequency content.