Comparison of Edge Detection Techniques Using Traditional and Soft Computing Approach

[1]Ms.S.Tamilselvi, [2]Mrs.S.Saritha

[1]Assistant Professor/ECE, [2]Assistant Professor/ECE Nandha College of Technology, Erode-52.

Abstract: **One important application of image segmentation is edge detection. Traditional methods are the most common method that are being used in edge detection. Soft computing is an emerging field that consists of elements of Genetic algorithm. A main technique used in Genetic algorithm is Ant Colony Optimization (ACO).The ant colony optimization algorithm is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. In this paper the main aim is to find out the edge of the image using both traditional and soft computing approach and also to compare the output to find out which one is the best.**

Keywords—**Edge detection, Edges, Image segmentation, Genetic Algorithm, Ant Colony Optimization (ACO).**

I. INTRODUCTION:

Segmentation subdivides an image into its constituent regions or objects. Normally the partitions of an image are the different objects in that image which is having the same color or texture. The result of image segmentation is a set of regions that collectively cover the entire image. The boundary of a surface is called as an Edge. Marking of a point in an image at which the intensity changes sharply is known as Edge Detection. The most important and common technique used in digital image processing is Edge Detection. The Edge detection techniques which are used in this paper are based on traditional methods and soft computing approach.

In traditional methods four type of detection techniques are used. They are Robert Edge detection, precut edge detection, Sobel edge detection and Canny edge detection. The three steps involved in traditional method are filtering, enhancement and detection. Filtering is used to reduce the noise. Enhancement is used to improve the detection of edges and detection is used to determine which the edge points.

The soft computing approach used here is Genetic Algorithm.

Genetic Algorithm are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. Algorithm is started with a set of solutions called population, solutions from one population are taken and used to form a new population, and this is motivated by a hope, that the new population will be better than the old one. Solutions which re selected to form new solutions are selected according to their fitness. The more suitable they are the chances they have to reproduce. This is repeated until some condition is satisfied. The three major operations in Genetic Algorithm are selection, crossover and mutation.

This paper is organized as follows in section II, traditional methods for edge detection is explained. Soft computing approach is explained in section III and also section II & section III output are focused. Section IV presents the conclusion.

II. TRADITIONAL METHODS FOR EDGE DETECTION

The most frequently used edge detection methods are

1. The Robert's Detection :

The Roberts cross operator performs 2-D spatial gradient measurement. It highlights the region of high spatial frequency which often corresponds to edges. Pixel values at each point in the output represent the estimated absolute magnitude of spatial gradient.

2. The Prewitt detection :

The Prewitt edge detector is an way to estimate the magnitude and orientation of an edge. The Prewitt operator is limited to possible orientations. This gradient based edge detection is estimated in the 3x3 neighborhood for eight directions all the eight convolution masks are calculated. One convolution mask is then selected, namely that with the largest module.

3. The Sobel detection :

The Sobel operator performs a 2-D spatial gradient measurement on an image. It is used to find the approximate absolute gradient magnitude at each point in an input gray scale image. The operator consists of a pair of 3x3 convolution kernels.

4. The Canny Detection:

The Canny edge detector first smoothes the image to eliminate noise. It then finds the image gradient to highlight regions with spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum. The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a nonedge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

III. SOFT COMPUTING APPROACH

Genetic algorithm is inspired by Darwin's theory of evolution.

One of the most important types of Genetic Algorithm is Ant Colony Optimization.

Generally in the real world, ants wander randomly and leave pheromone trails while find food and return to their colony and the other ants will follow the trail and will find out the shortest path. The same concept is followed in this method for finding out the edges.

The steps to be performed in ant colony optimization are Initialization process, Construction process, Update Process and Decision process.

A. *Initialization process:*

In the initialization process, each of the ants is assigned a random position in the image. The initial value of each element in the pheromone matrix is set to a constant, which is small but non-zero. Also, the heuristic information matrix is constructed based on the local variation of the intensity values. The heuristic information is determined during initialization since it is dependent only on the pixel values of the image, thus, constant.

The heuristic information at pixel (i,j) is determined by the local statistics at that position:

$$\eta_{i,j} = \frac{V_c(I_{i,j})}{\sum_{i=1}^{M_1}\sum_{j=1}^{M_2} V_c(I_{i,j})} \qquad (1)$$

Where Ii,j s the intensity value of pixel at (i,j).

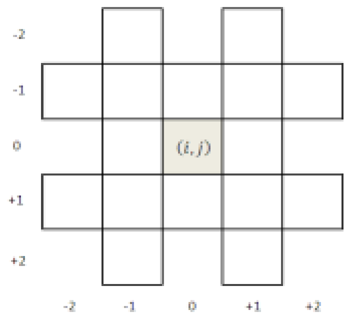


Fig. 1. A local configuration for computing the intensity

Variation at (i,j) .

B. *Construction process:*

In a construction process, a set of artificial ants construct solutions from a finite set solution components from a fully connected graph that represents the problem to be solved. A construction process contains a certain number of construction steps. Ants traverse the graph until each has made the target number of construction steps. The choice of solution components is made probabilistically. The exact decision rule for choosing the solution components varies across different ACO components. On the nth construction process, the kth ants move from node I to node j according to the transition probability, the probability that an ant will move from node I to node j. The AS decision rule is based on the transition probability given by

$$p_{i,j}^{(n)} = \frac{\left(\tau_{i,j}^{(n-1)}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}}{\sum_{j\in\Omega_i}\left(\tau_{i,j}^{(n-1)}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}}, \quad \text{if } j \in \Omega_i \qquad (2)$$

where $\tau_{i,j}^{(n-1)}$ is the quantity of pheromone on the edge from node i to node j; $\eta_{i,j}$ is the heuristic information of the edge from node I to node j;Ω_i is the neighbourhood nodes for the ant given that it is at node i;α and β are constants that control the influence of the pheromone and heuristic information respectively,to the transition probability. $\sum_{j\in\Omega_i}\left(\tau_{i,j}^{(n-1)}\right)^{\alpha}\left(\eta_{i,j}\right)^{\beta}$ is a normalization factor to limit the values of $P_{i,j}^{(n)}$ within [0,1].

C. *Update process:*

Each time an ant visits a pixel, it immediately performs a local update on the associated pheromone. The amount of the pheromone on the pixel (i,j) on the nth iteration, $\tau_{i,j}^{(n)}$, is updated based on the equation for ACS local pheromone update:

$$\tau_{i,j}^{(n)} = (1-\varphi)\cdot\tau_{i,j}^{(n)} + \varphi\cdot\tau_{init} \qquad (3)$$

where ψ ε (0,1] is the pheromone decay coefficient and τ_0 is the initial pheromone value.

The global pheromone update is performed only on the best-so-far solution according to the equation

$$\tau_{i,j}^{(n)} = (1-\rho)\cdot\tau_{i,j}^{(n-1)} + \rho\cdot\Delta\tau_{i,j}^{(k_{bs})} \qquad (4)$$

where

$$\Delta\tau_{i,j}^{(k_{bs})} = \begin{cases}\frac{1}{L_{k_{bs}}}, & \text{if ant } k_{bs} \text{ used edge } (i,j)\\ 0, & \text{otherwise}\end{cases} \qquad (5)$$

where L_k is the tour length associated with the best-so-far solution.

Local pheromone updates are interleaved with the solution construction process; the pheromone values change within the iteration.

The permissible range of movement of the ants is obtained from the 8-connectivity neighborhood (Fig-2). An ant can move to any adjacent pixel. But, this is restricted by the condition that an ant moves only to a node that it has not recently visited. This is to prevent the ants from visiting the same set of nodes repeatedly. In order to keep track of recently visited nodes, each ant has a memory.

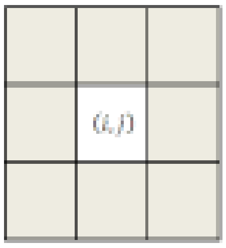


Fig. 2. Neighborhood pixels (shaded) of the pixel (i,j): 8-connectivity neighborhood

After all the ants finish the construction process, global pheromone update is performed on pixels that have been visited by atleast one ant:

$$\tau_{i,j}^{(n)} = (1-\rho)\cdot\tau_{i,j}^{(n-1)} + \rho\cdot\sum_{k=1}^{K}\Delta\tau_{i,j}^{(k)} \qquad (6)$$

Where $\Delta\tau_{i,j}$ is the amount of pheromone deposited by the kth ant on pixel (i,j). $\Delta\tau_{i,j}$ is assigned to be equal to $\eta_{i,j}$. Its reciprocal $1/\eta_{i,j}$, can be interpreted as the tour length.

There are two main differences in the global pheromone update. First, there is no selection of a best-so-far tour; all visited pixels are updated. In ACS, only the solution components belonging to the best-so-far solution is updated. Having a best-so-far solution makes sense for the ACO- based edge detection approach; however, an individual ant does not aim to produce a complete possible solution to the problem. Instead, the global of each ant is to produce only a partial edge trace in the image.

The second difference is in the function for the amount of pheromone deposited. In ACS, the amount of pheromone deposited on edges within a single tour is equal for all the edges and is a function of the cost of the tour as a whole.

D. Decision process:

The final pheromone matrix is used to classify each pixel either as an edge or a non-edge. The decision is made by applying a threshold on the final pheromone matrix $\tau^{(N)}$. The threshold value is computed based on the method known as Otsu thresholding technique.

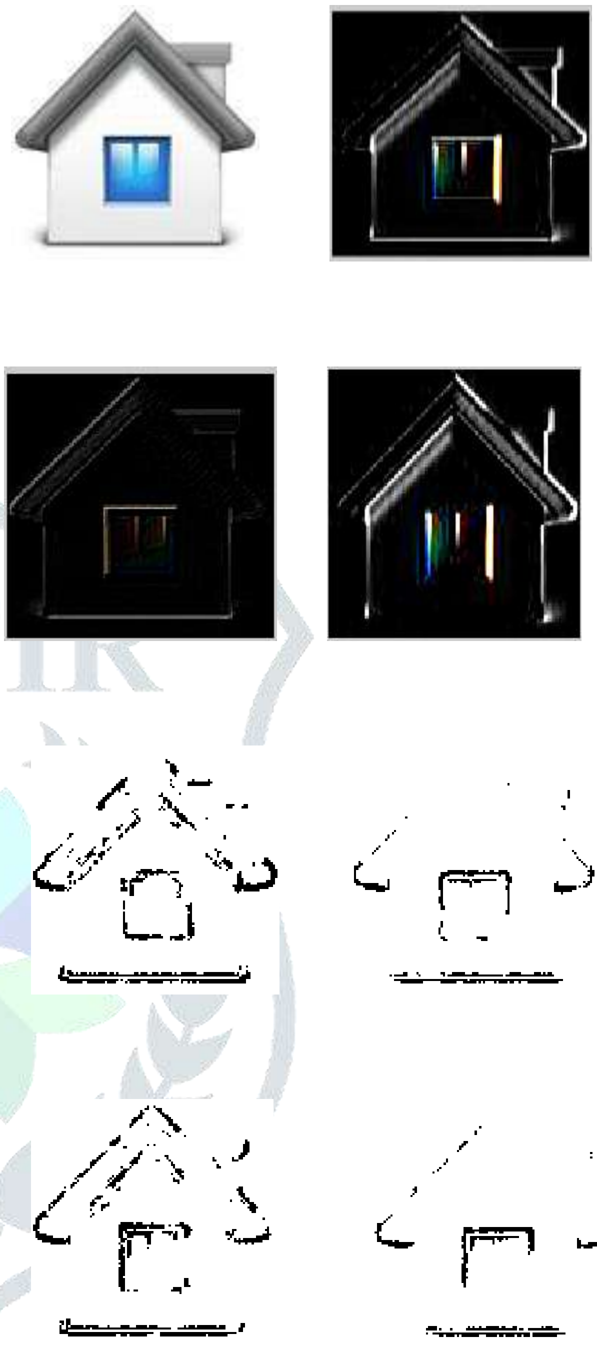
The initial threshold $T^{(0)}$ is selected as the mean value of the pheromone matrix. Next, the entries of the pheromone matrix is classified into two categories according to the criterion that its value is lower than $T^{(0)}$ or larger than $T^{(0)}$.

where

$$T^{(0)} = \frac{\sum_{i=1:M_1}\sum_{j=1:M_2}\tau_{i,j}^{(N)}}{M_1 M_2}, \qquad (7)$$

Then the new threshold is computed as the average of two mean values of each of above two categories. The above process is repeated until the threshold value does not change any more.

Experimental results:



IV. Conclusion:

Thus both the traditional method and the soft computing approach are implemented successfully for a database containing more number of images within a reasonable time. By comparing the outputs of both the traditional and soft computing approach we conclude that the output from soft computing approach is better than that of the output produced by the traditional approach.

V. References:

[1] M. Dorigo and S. Thomas, *Ant Colony Optimization.* Cambridge MIT Press, 2004.

[2] Jing Tian, Weiyu Yu, and Shengali Xie, *An Ant Colony Optimization for Image Segmentation.* Congress of Evolutionary Computation,2008

[3] Anna Veronica Baterina and Carlos Oppus, *Ant Colony Optimization for Image Edge Detection.* Recent advances in signal processing, Robotics and Automation, 2004.

[4] H. Nezamabadi-pour, S. Saryazdi, and E. Rashedi, *Edge detection using Ant Algorithms*, *Soft Computing*, Vol.10,2006,pp.623-628.