International Journal of Scientific Research in Computer Science and Engineering

Research Paper

Volume-1, Issue-2, March-April-2013

Available online at www.isroset.org

ISSN 2320-7639

Edge Detection in blurred images using Ant Colony Optimization Techniques

ShwetaAgarwal

Computer Science and Engineering, MVN University India, gud22aga@gmail.com

***Abstract*—This paper defines edge detection and prioritization in blurred images. Edge Detection has been center of concern in image processing, as this help to distinguish object from the system and draw boundaries. A large number of powerful and effective techniques exist for edge detection in digital images. But sometimes our image is not clear, it is blurred. All edge detection algorithms are not able to process such blurred images. So by using ant colony optimization technique it is possible to detect and prioritize edges in blurred images.**

***Keywords*—** ACO Ant Colony Optimization, ED Edge Detection, IP Image Processingt,SM-Ant bot system,SAant bot system.

I. INTRODUCTION

Edge detection is a fundamental tool in image processing and computer vision in the areas of feature detection and feature extraction. Image edge detection refers to the extraction of the edges in a digital image. It is a process whose aim is to identify points in an image where discontinuities or sharp changes in intensity occur. The main aim of edge detection is to identifying points in a digital image at which the image brightness changes sharply or, more formally. Image edge detection deals with extracting edges in an image by identifying pixels where the intensity variation is high. It is a fundamental tool used in most image processing applications to obtain information from the frames as a precursor step to feature extraction and object segmentation. This process detects outlines of an object and boundaries between objects and the background in the image. The edge is the set of the pixel, whose surrounding gray is rapidly changing. The internal characteristics of the edge-dividing area are the same, while different areas have different characteristics. The edge is the basic characteristics of the image. There is a lot of information of the image in the edge. Edge detection is to extract the characteristics of discrete parts by the difference in the image characteristics of the object, and then to determine the image area according to the closed edge.

II. EDGE DETECTION METHODS

Edge information for a particular pixel is obtained by exploring the brightness of pixels in the neighborhood of that pixel. If all of the pixels in the neighborhood have almost the same brightness, then there is probably no edge at that point. However, if some of the neighbors are much brighter than the others, then there is a probably an edge at that point. Concept on neighboring pixels is very important in image processing. The use of neighboring pixels in image processing algorithm provides a localized window of spatial information as suggested by Perkowski. Traditional edge detectors were based on a rather small 3x3 neighborhood, which only examined each pixel's nearest neighbor. This may work well but due to the size of the neighborhood that is being examined, there are some limitations to define the accuracy of the final edges. These local neighborhoods will only detect local discontinuities, and it is possible that this may cause false edges to be extracted.

A. Roberts edge detection method –

The Roberts method performs a 2-D spatial gradient measurement on an image and emphasizes regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image.

0	0	0
0	-1	0
0	0	0

Gx

0	0	0
0	0	-1
0	1	0

Gy

Figure 1. Masks use for gradient operation on Roberts operation

Diagonal edge gradients can be obtained by forming running difference of diagonal pairs of pixels. This is the basis of the Robert cross difference method which is defined in magnitude form and it is shown as:

$$G(x, y) = |G1 (x, y) + G2 (x, y)|$$

again the spatial gradient amplitude in square root form can be defined as:

$$G(x, y) = |G1 (x, y)|2 + |G2 (x, y)| 2$$

Where

$$G1 (x, y) = F (x+1, y+1) - F(x, y)$$

And

$$G2 (x, y) = F (x, y+1) - F(x+1, y).$$

The angle of orientation of the edge giving rise to the spatial gradient (relative to the pixel grid orientation) is given by: $\theta = \arctan (Gy / Gx) - 3\pi / 4$.

B. Prewitt edge detection method –

The Prewitt edge detector is an appropriate way to estimate the magnitude and orientation of an edge. It is based on first order derivative of intensity function. In this method, similar weights are assigned to all the neighbors of candidate pixels whose edge strength is being calculated. Although differential gradient edge

Corresponding Author: *Shweta Agarwal*

detection needs a rather time consuming calculation to estimate the orientation from the magnitudes in the x and y-directions, the compass edge detection obtains the orientation directly from the kernel with the maximum response.

-1	0	+1
-1	0	+1
-1	0	+1

+1	+1	+1
0	0	0
-1	-1	-1

GxGy

Figure 2. Masks use for gradient operation on Prewitt operation

The pewit square root edge gradient is given by
Where

GR (x, y)=1/ (K+2) [(A2+ KA3+A4) - A0+ KA7+ A6]

&

GC (x, y)=1/ (K+2) [(A2+ KA1+A2) - A0+ KA5+ A4]

Where K = 1.

C. Sobel edge detection method–

In this method, higher weights are assigned to pixels close to candidate pixels. The Sobel method performs a 2-D spatial

$$G(x, y) = \sqrt{|G_R(x, y)|^2 + |G_C(x, y)|^2}$$

gradient measurement on an image and emphasizes on regions of high spatial frequency that correspond to edges. Sobel edge detectors masks detect vertical and horizontal edges separately and these directional edges are combined finally. 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.

-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

GxGy

Figure 3. Masks use for gradient operation on Sobel operation

The gradient magnitude is given by:

G = (G x2 + Gy2)½ ,

Typically, an approximate magnitude is computed using:

|G|=|Gx|+|Gy|

The angle of orientation of edges giving rise to the spatial gradient is given by:

Arg (G) = tan−1 (| Gy | / | Gx |).

D. Canny edge detection method–

The Canny edge detection algorithm is known to many as the optimal edge detector. Canny edge detection technique, the procedure is as follows:

Step1. Convolve the image with a two dimensional Gaussian filter to smooth it.

Step2. Differentiate the image in two orthogonal directions.

Step3. Calculate the gradient amplitude and direction.

Step4. Perform non-maximal suppression.

Step5. Any gradient value that is not a local peak is set to zero. The gradient direction is used in this process.

Step 6. Threshold these edges to eliminate `insignificant' edges.

III. ANT COLONY OPTIMIZATION TECHNIQUE

Ant colony optimization (ACO) is a class of optimizationalgorithms modeled based on the actions of an ant colony. ACO methods are useful in problems that need to find paths to goals. Artificial 'ants'—simulation agents—locate optimal solutions by moving through a parameter space representing all possible solutions. Real ants lay down pheromones directing each other to resources while exploring their environment. The simulated 'ants' similarly record their positions and the quality of their solutions, so that in later simulation iterations more ants locate better solutions.

Ant colony optimization (ACO) is paradigm inspired by the intelligence of real ants, for finding solutions to combinatorial optimization problems. More specifically, ACO is inspired by food foraging behaviors exhibited by ant societies. Ants as individuals are unsophisticated living beings. This algorithm is a member of ant colony algorithms family, in swarm intelligence methods, and it constitutes some met heuristic optimizations. The first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food.

The ants communicate using a chemical substance called pheromone. As an ant travels, it deposits a constant amount of pheromone that other ants can follow. Each ant initially moves in a somewhat random fashion, but when an ant encounters a pheromone trail, it must decide whether to follow it. If it follows the trail, the ant's own pheromone reinforces the existing trail, and the increase in pheromone increases the probability of the next ant selecting the path. Therefore, the more the ants travel on a path, the more attractive the path becomes for subsequent ants. Additionally, an ant using a short route to a food source will return to the nest sooner and, therefore, mark its path twice, before other ants return. This directly influences the selection probability for the next ant leaving the nest. Over time, as more ants are able to complete the shorter route, pheromone accumulates faster on shorter paths and the longer paths are less reinforced and ultimately abandoned. When looking for food, ants tend to follow trails of pheromones whose concentration is higher. These trails are created by individuals looking for food, to guide others toward the same sources of food. The concentration of pheromone is stronger in highly visited places. Because of the distance travelled by ants to reach food sources and return to the nest, trails near the nest develop a higher concentration of pheromone. The phenomenon give rise to one of the most important type of optimization algorithms named as ACO [13].

The ACO mechanism has the following desirable properties:

1. ACO is versatile in that it can be applied to similar versions of the same problem. An example of a similar version might be the traveling salesman problem, which can be directly extended to the asymmetric traveling salesman problem.
2. ACO is robust, and can be applied with only minimal changes to other combinational optimization problems, such as the quadratic assignment problem and the job shop scheduling problem.
3. ACO is a population-based approach, and allows positive feedback exploitation to be adopted as a search mechanism.

A. ACO based algorithm –

An image matrix model is used in computer science as a discrete model. The use of ACO in image processing and graphical applications has received some attention over the past few years [Popovici, A. and Popovici, D., 2002]. As stated previously, ACO techniques appear as a natural tool for image processing due to their local nature and simple parallel computing implementation. In this section, one main algorithm has been presented and investigates its variation and effectiveness for processing of gray level images. The algorithm will correspond to edge detection for gray scale images. The application of the ACO techniques to real images will be presented, which together with the results will show the performance characteristics and comparison. The ACO techniques algorithm for k gray levels of digital images is on the basis of a two-dimensional ACO techniques (I, N, V, f) with V = {0, 1, 2, …, k-1}, where k is a number of states, N is the type of neighborhood (e.g. n is total number of neighbors varies from 0 to 8), while the local transition function defined as f: Vn into V.

Given a configuration c of the pixels in the ACO at a certain time t, the configuration c′ at time t + 1 for each pixel x can be calculated as c′(x) = f(c (x1. . . xn)).

In such a two dimensional ACO, specific neighborhoods can be defined. For example, the so-called Von Neumann neighborhood (4-neighborhood) and Moore neighborhood (8-neighborhood) for a pixel x i, j is defined as:

x i−1, j, x i, j−1, x i, j+1, x i+1,j,

andx i−1, j , x i, j−1, x i, j+1, x i+1, j, x i-1, j+1, x i+1,j -1, x i+1,j+1, x i-1,j-1, respectively. Gray Level Edge Detection.

In digital image processing, each image is quantized into pixels. With gray-scale images, each pixel indicates the level of brightness of the image in a particular spot: 0 represents black and 255 represent white. An edge is an abrupt change in the brightness (gray scale level) of the pixels. Edge information for a particular pixel is obtained by exploring the brightness of pixels in the neighborhood of that pixel. If all of the pixels in the neighborhood have almost the same brightness, then there is probably no edge at that point. However, if some of the neighbors are much brighter than the others, then there is probably an edge at that point. Measuring the relative brightness of pixels in a neighborhood is mathematically similar to calculating the derivative of brightness. Brightness values are discrete, not continuous, so the derivative function has been estimated. The objective of the edge detection techniques using ACO techniques is to enhance the magnitude of the local differences in gray level values between regions of the images. Over regions which are different, changes must be made to enhance the edges. Here it has been considered grid lattices of 256 × 256 pixels where each pixel has 256 states values ranging 0-255 using the grayscale method. Each pixel is scanned and checked for being a part of active edge using Moore Neighborhood techniques. A memory matrix representing image is maintained. If a pixel is a part of edge then corresponding matrix pixel value is marked else 0. Finally the edged image is constructed using the original input image and image-edge value matrix.

IV. PROPOSED WORK

Image processing has been under focus since long, lot of study and work has been done in this field, a variety of techniques and algorithms have emerged in the process.

Edge Detection has been center of concern in image processing, as this help to distinguish object from the system and draw boundaries.

A large number of powerful and effective techniques exist for edge detection in digital images. But sometimes our image is not clear, it's blurred. All edge detection algorithms are not able to process such blurred images. For such images De-blurring Edge Detection algorithms come into picture. These algorithms use Gaussian de-blurring and then implement edge detection. These usually results in loss of temporal data in blurred images, which may lead to loss of necessary information, and hence degraded results.

I propose a new approach towards edge detection in blurred images. Our algorithm does not consider image de-blurring hence eliminating any chances of data loss. We know that a blur image will produce multiple edges in an area of concern, few among these edges will be non-prominent and less useful, and few others will be more prominent, strong and useful. We simply do not remove these non-prominent edges, we detect each and every possible edge in our area of concern, calculate its strength, importance and assign different strength values of edge pixels.

We will be converting input images to Gray scale images using generalized formula:-

Grayscale Value= .299 * red + .587 * green + .114 * blue

Strength value of a pixel is calculated as maximum variation in Grayscale values between his own value and neighborhood pixel values. Below Tables show how this is done

Empirical Probability:

TABLE I. STRENGTH CALCULATION FOR RADIUS 1

50	50	100
50	120	100
50	120	100

TABLE II. STRENGTH CALCULATION FOR RADIUS 2

0	50	50	100	100
0	50	50	100	100
0	50	120	100	100
0	50	120	100	100
0	50	120	100	100

Table I. shows strength calculation for radius of 1, out of two possible differences of 120-50=70, 120-100=20; we consider maximum difference of 70.

Table II.Shows strength calculation for radius of 2, out of three possible differences of 120-0=120, 120-50=70, 120-100=20,140-120=20 we consider maximum difference of 120.

Values indicated gray scale values on a scale of 0-255, Center pixel is current pixel for which strength is being calculated.

V. RESULT AND DISCUSSION

This section presents and analyses the results produced by above mentioned traditional method and proposed method. It is quite difficult to develop standard performance criteria and methods to evaluate the effectiveness of each edge detector. Locating a real edge pixel becomes extremely crucial. Edge slope angle and its spatial orientation are also important criteria in the evaluation.

There are five different criteria those are typically used for testing the quality of an edge detector:

- The probability of a false positive (marking something as an edge which isn't an edge)
- The probability of a false negative (failing to mark an edge which actually exists)
- The error in estimating the edge angle.
- The mean square distance of the edge estimate from the true edge.
- The algorithm's tolerance to distorted edges and features such as corners and junctions.

Experiments were carried out over several 256×256 sizes of standard test images. All above said edge detection method like Roberts's operator, Sobel operator, Prewitt operator, canny operator and the proposed method have been implemented on same standard test images using latest software viz, .Net technology.

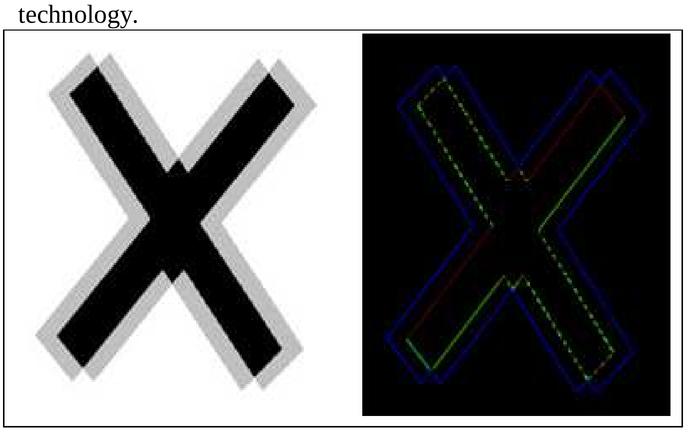


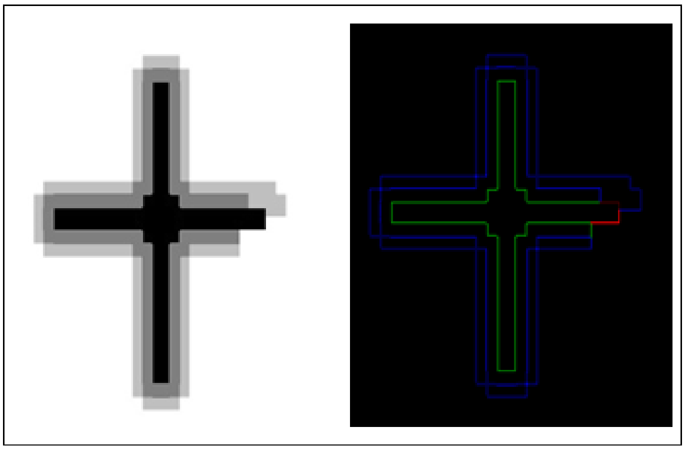
Figure 4. Edge detection of a blurred images

It can be seen that ACO gives clean and continuous edges and marks edges by strength with different colours. After viewing all the results it can be said that new proposed technique has provided better results. Quantitative analysis plays a good role in experimental performance for satisfactory results. Here, various edge detection methods have been measured in terms of computing time, keeping considering the quantitative analysis as a major factor of performance evaluation.

VI. CONCLUSION

The proposed ACO-based Image Edge detection approach has been successfully implemented. With the evidence of shown images and results, I can state that I have successfully detected edges in normal as well as blurred digital images and prioritized them using different color values, and this prioritization can be visually correlated with the help of differently color marked edges according to their strength and importance.

Results appear to be above satisfactory for different sets of inputs, some of them can be visually correlated and some can't. Thus I can say that I have been successful in presenting and implementing a new approach for detecting and prioritizing edges in blurred digital images, without using any de-blurring and loss of data.



VII. ACKNOWLEDGMENT

We would like to acknowledge and extend my heartfelt gratitude to the GOD, parents and husband.

REFERENCES

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

[2]. Jing Tian, Weiyu Yu, and Shengli Xie, "An Ant Colony Optimization Algorithm For Image Edge Detection", in Proc. of the IEEE International, pp.**751-756,2008**.

[3]. Hossein Nezamabadi-pour · Saeid Saryazdi Esmat Rashedi, "Edge detection using ant algorithms", in proc. of Springer-Verlag, pp**.623-628,2005**.

[4]. Aleksandar Jevti´c, Joel Quintanilla-Dominguez, M.G. Cortina-Januchs and Diego Andina, "Edge Detection Using Ant Colony Search Algorithm and Multiscale Contrast Enhancement", in the Proceedings of the 2009 IEEE International Conference on systems, man, and cybernetics, pp.**2193-2198,2009.**

[5]. X. Zhuang, "Edge Feature Extraction in Digital Images with the Ant Colony System", in proc. of the IEEE international Conference an computational intelligence for Measurement Systems and Applications, pp. **133-136,2004.**

[6]. Alirezae Rezaee, "Extracting edge of images with ant colony", in the Journal of Electrical Engineering,pp. **57–59,2008.**

Shweta Agarwal was born in Jodhpur,Rajasthan, on 22nd November 1983. Sheis an Assistance Professor of MVN University,Palwal,Haryana. She received a B.E., in 2005, MBA(Finance), in 2009, and a M.Tech., in 2011. Her research interests are in intelligence informaticsand image processing. Ms. Shweta Agarwal is an author of Edge Detection ans Prioritization in blurred images using Ant Colony Optimization techniquesin International journal.She has published 4 papers in International journal and 4 in national or international conferences.