# Image Edge Detection Using Quantum Ant Colony Optimization

[1]Jian Zhang, [2]Jiliu Zhou, [2]Kun He, [3]Huanzhou Li
*[*1] School of Electronics and Information Engineering, Sichuan University,*
*E-mail: dctscu07@163.com*
*[2] College of Computer Science, Sichuan University*
*[3] School of Physics and Electronics Engineering, Sichuan Normal University*

### *Abstract*

*Ant colony optimization algorithm (ACO) which performs well in discrete optimization has already been used widely and successfully in digital image processing. Slow convergence, however, is an obvious drawback of the traditional ACO. A quantum ant colony algorithm (QACO), based on the concept and principles of quantum computing can overcome this defect. In this study, a QACO-based edge detection algorithm is proposed. Quantum bit (qubit) and quantum rotation gate are introduced into QACO to represent and update the pheromone respectively. Experiments and comparisons show that QACO is an efficient and effective approach in image edge detection.*

**Keywords**: *Edge Detection, Ant Colony Optimization, Quantum Ant Colony Optimization*

## 1. Introduction

Edge is one of the simplest and the most important features of image[1]. Edge detection is an important content of image processing and low-level computer vision. The purpose of edge detection is not only to extract the edges of the interested objects from an image, but also to lay the foundation for shape extraction, image segmentation, image matching, image tracking and image retrieval, etc.[2]. In shapes-based image retrieval, detected edges should have stable positions. Edge detection has become one of the key research topics in image retrieval technology. Numerous edge detectors, such as differential-based edge detection algorithm, clustering-based edge detection algorithm, multi-scale edge detection algorithm and their improved approaches have been proposed[3,4], and these approaches show their advantages and disadvantages when dealing with different images. In practice, no single edge detection algorithm, has been devised which will successfully determine every different type of edge. So, an approach combing various algorithms may help solving the problem better.

Ant colony optimization (ACO) algorithm is easy to combine with other algorithms for its strong robustness and adaptability. Computing mechanism of parallel and positive feedback has been shown its powerful abilities in solving combination optimization and NP problems which can not be solved by traditional algorithms[5] . And ACO has already been used widely and successfully in digital image processing including edge detection in recent years[6]. However, time consuming is an obvious drawback of the traditional ACO algorithm. Quantum ACO algorithm (QACO) which combines the ACO and quantum computing (QC) can overcome this defect. Quantum computing is a very attractive research area in resent years. The essential feature of quantum computing is the use of quantum superposition, interference and entanglement. And its powerful computational ability is shown in the following areas: parallelism, exponential storage capacity and the feature of exponential acceleration[7,8]. Several researchers have introduced quantum computing into traditional algorithms and proposed a series of quantum computational intelligence theories[9,10].Combing the merits of ACO and QC, several researchers have introduced QACO into some research areas[11,12]. This study introduces the QACO into digital image edge detection. Experiments and comparisons show that QACO is an efficient and effective approach to image edge detection.

## 2. Quantum ant colony optimization algorithm

### 2.1. Ant colony optimization algorithm

ACO algorithm is based on the foraging behavior of real ants that ants communicate by means of a kind of substance called pheromone which can enable them to find the shortest paths (the most preferred ways) between their nest and food sources. While walking from their nest to the food sources and vice versa, ants deposit on the way pheromone, forming in this way a pheromone trail, and with the elapse of time the concentration of the pheromone becomes weaker due to evaporation. Ants can smell the pheromone and when choosing their way, they tend to choose, in probability, the paths with the strongest pheromone concentration. And the more ants choose the path, the stronger the pheromone concentration can be. Thus the pheromone trail can help the ants find quickly the food sources or the nest[13,14]. The first ACO algorithm called ant system is proposed to solve the traveling salesman problem (TSP) by Dorigo[15]. It makes up the main framework of other ACO algorithms and is considered as a prototype. The procedure of ACO algorithm can be summarized as follows:

- Initialize the positions of all ants and the pheromone matrix.
- For the construction-step, move the ant $k$ from city $i$ to city $j$ according to formula (1).

$$s=\begin{cases}\arg\left\{\max\limits_{j\in N_i^k}[(\tau_{ij})^{\alpha}\cdot(\eta_{ij})^{\beta}]\right\}, & \text{while } q\le q_0\\ S, & \text{else}\end{cases} \tag{1}$$

Where $\tau_{ij}$ is the pheromone and $\eta_{ij}$ is the heuristic information, and the parameters $\alpha$ and $\beta$ control the importance of the pheromone versus the heuristic information. $q$ is a random number between [0, 1] and $q_0$ ( $0<q_0<1$ ) is a pre-specified parameter between. If $q>q_0$, the ants can choose the next city to visit according to the probability distribution $p_{ij}$ given by (2).

$$p_{ij}=\frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{j\in N_i^k}(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}\quad,\quad \text{if } j\in N_i^k \tag{2}$$

- At the end of each iteration, the intensity of pheromone $\tau_{ij}$ is updated by a pheromone trail updating rule:

$$\tau_{ij}=(1-\rho)\cdot\tau_{ij}+\sum_{k=1}^{K}\triangle\tau_{ij}^{k} \tag{3}$$

$$\triangle\tau_{ij}^{k}=\begin{cases}Q/L_k, & \text{if ant } k \text{ passed link}(i,j)\\ 0, & \text{else}\end{cases} \tag{4}$$

where $\rho$ ( $0<\rho<1$ ) denotes the pheromone evaporation ratio. $Q$ is a constant, $K$ is the ant number and $L_k$ is the length of tour taken by ant $k$.

- Iteration stopping when met with the terminal condition.

## 2.2. Quantum ant colony optimization algorithm

QACO is a newly quantum computing-based probability optimization theory, it is based on the concepts and theories of quantum computing. And in quantum computing, the smallest information unit is quantum bit (qubit)[16], the characteristic of the qubit is that any linear superposition of solutions can be represented. This superposition can be expressed as follows:

$$\psi=\alpha|0\rangle+\beta|1\rangle\ ,\ (|\alpha|^2+|\beta|^2=1) \tag{5}$$

$\alpha$ and $\beta$ are complex numbers, $|\alpha|^2$ gives the probability that the qubit will be “0” and $|\beta|^2$ presents the probability that the qubit will be “1”. A qubit may be in “1” state, in “0” state, or in a linear superposition of the two. In QACO, the pheromone will be presented as follows:

$$\tau=[\tau_1,\tau_2,...,\tau_m]=\left[\begin{array}{c|c|c|c}\tau_{\alpha}^{1}&\tau_{\alpha}^{2}&\cdots\cdots&\tau_{\alpha}^{m}\\ \tau_{\beta}^{2}&\tau_{\beta}^{2}&\cdots\cdots&\tau_{\beta}^{m}\end{array}\right] \tag{6}$$

When an ant moves from one node to another node, the pheromone will be update. In QACO, the quantum rotation gate is employed to update the pheromone intensity as follows:

$$\begin{bmatrix} \tau'^{i}_{\alpha} \\ \tau'^{i}_{\beta} \end{bmatrix} = \begin{bmatrix} \cos(\triangle\theta_i) & -\sin(\triangle\theta_i) \\ \sin(\triangle\theta_i) & \cos(\triangle\theta_i) \end{bmatrix} \begin{bmatrix} \tau^{i}_{\alpha} \\ \tau^{i}_{\beta} \end{bmatrix} \tag{7}$$

where $\triangle\theta_i$ is the rotation angle, $i = 1, 2, \cdots\cdots, m$. As can be seen from above, $\triangle\theta_i$ is a very important parameter. The quantum rotation gate can be illustrated in Figure 1.

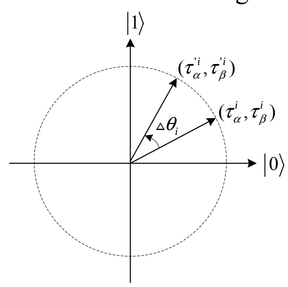



**Figure 1.** Polar Plot of the Quantum Rotation gate for qubit.

## 3. Quantum ant colony optimization for image edge detection

In this section, the QACO algorithm for edge detection will be described in details. The input image is considered as a two dimensional graph whose nodes are image pixels. Ants move from a pixel to another pixel on the graph for constructing a pheromone matrix, each entry of which represents the edge information at each pixel location of the image. The movements of the ants are steered by the pheromone intensity and the heuristic information. The ants will change their position in the image according to the transition rules and leaving a certain amount of pheromone on the visited nodes. The more ants follow a trail, the more pheromone is got, and the more attractive this trail becomes to be followed by other ants. In the end, edge detection results can be gained through analyzing the pheromone distribution in the image. The implementation details of this study are described as follows: 1 -1

### 3.1. Initializing ant distribution and pheromone matrix.

Generally, the number of ants is set as: $K = \sqrt{M \times N}$ ($M$ represents the length and $N$ represents the width of the input image), $K$ ants are placed randomly, with at most one ant on each pixel. The pheromone of every pixel $i$ is presented by a qubit: -1

$$\tau_i = [\tau^{i}_{\alpha}, \tau^{i}_{\beta}]^T = [1/\sqrt{2}, 1/\sqrt{2}]^T, \quad |\tau^{i}_{\alpha}|^2 + |\tau^{i}_{\beta}|^2 = 1 \tag{8}$$

### 3.2. Probability decision

Firstly, a random number $\lambda$ between [0, 1] is generated and a binary solution can be decided by:

$$\tau_i = \begin{cases} 1, & \text{if } |\tau^{i}_{\beta}|^2 > \lambda \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

At each step, an ant $k$ moves from pixel $i$ to $j$ with a probability $p_{ij}$. The probability is given by:

$$p_{ij} = \begin{cases} \dfrac{\tau_i(\eta_j)^{\gamma} w_j(\Delta)}{\sum_{j\in\Omega} \tau_j(\eta_j)^{\gamma} w_j(\Delta)}, & \text{if } j \in \Omega \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $j \in \Omega$ indicates all the pixels that are in the 8-neighborhood of the pixel $i$. $\tau_j$ denotes the pheromone value. $\eta_j$ represents the visibility (heuristic information) and its value is given as below:

$$\eta_j = \triangle I_j \tag{11}$$

The term $\triangle I_j$ represents the gradient of the pixel $j$ and defined as follows:

$$\triangle I_j = \max\left\{\begin{array}{c} \left|I(m,n-1)-I(m,n+1)\right|, \\ \left|I(m-1,n)-I(m+1,n)\right|, \\ \left|I(m-1,n+1)-I(m+1,n-1)\right|, \\ \left|I(m-1,n-1)-I(m+1,n+1)\right| \end{array}\right\} \tag{12}$$

where $m$ denotes the row and $n$ denotes the column of the pixel $j$ in the image. $I(m,n)$ represents the gray value of pixel $j$. In formula (10), the notations of $\gamma$ is the control parameter that determine the relative influence of the visibility. $\Delta$ measures the magnitude of the changes in direction at each step and can take the discrete value: $\Delta = 0, \pi/4, \pi/2, 3\pi/4, \pi$. $w(\Delta)$ is a weighting function and this function ensures that very sharp turns are much less likely than turns through smaller angles, thus each ant in the colony has a probabilistic bias in the forward direction[17,18]. The proposed method defines the value of $w(\Delta)$ as: $w(0)=1$, $w(\pi/4)=1/2$, $w(\pi/2)=1/4$, $w(3\pi/4)=1/12$, $w(\pi)=1/20$.

## 3.3. Transition rules

An ant $k$ located at pixel $i$ will move to pixel $j$ according to

$$s = \begin{cases} \arg\left\{\max\limits_{j\in\Omega}[\tau_j \cdot (\eta_j)^{\gamma} \cdot w_j(\Delta)]\right\}, & \text{while } q \le q_0 \\ S, & \text{else} \end{cases} \tag{13}$$

If $q > q_0$, the ants can choose the next node to visit according to the probability distribution given by formula (10).

## 3.4. Pheromone update

The pheromone matrix needs to be updated by using a quantum rotation gate. The pheromone matrix needs to be updated twice during the QACO procedure. First, after an ant moves from the current pixel $i$ to the next pixel $j$, the pheromone trail of the pixel $j$ is updated as follows:

$$\begin{bmatrix} \tau'^{j}_{\alpha} \\ \tau'^{j}_{\beta} \end{bmatrix} = \begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) \\ \sin(\theta_j) & \cos(\theta_j) \end{bmatrix} \begin{bmatrix} \tau^{j}_{\alpha} \\ \tau^{j}_{\beta} \end{bmatrix} \tag{14}$$

The notation of $\theta_j$ denotes the rotation angle gained by[19]:

$$\theta_j = \triangle\theta \cdot f(\alpha_j, \beta_j) \tag{15}$$

$\triangle\theta$ is a variable which control the rotation angle:

$$\triangle\theta = h \cdot \mathrm{e}^{-t/t_{\max}} \tag{16}$$

where $h$ is a weight factor. $t_{\max}$ is a constant which describe the maximum iterative number and $t$ is the iterative number. The term $f(\alpha_j, \beta_j)$ is defined as follows:

$$f(\alpha_j, \beta_j) = \mathrm{sgn}(\triangle\theta) \cdot \frac{\beta_j}{\alpha_j} \tag{17}$$

And the rotation direction is decided as follows:

$$\mathrm{sgn}(\triangle\theta) = \mathrm{sgn}(A) \tag{18}$$

$$A = \tau^{j}_{\alpha} \cdot \tau^{j}_{\beta} \tag{19}$$

$$\beta_j = \frac{\triangle I_j - \triangle I_{\text{average}}}{\triangle I_{\max}} \cdot 0.5 + 0.5 \tag{20}$$

where $\triangle I_{\text{average}}$ and $\triangle I_{\max}$ represents the average value and the maximum value of the pixel gradient of the image.

$$\alpha_j = \sqrt{1-(\beta_j)^2} \tag{21}$$

The second update is global update on all paths, whether or not the path is passed through. The pheromone is updated after each cycle, and the pheromone matrix is updated as:

$$\begin{bmatrix} \tau'^{j}_{\alpha} \\ \tau'^{j}_{\beta} \end{bmatrix} = \begin{bmatrix} \cos(\theta_0) & -\sin(\theta_0) \\ \sin(\theta_0) & \cos(\theta_0) \end{bmatrix} \begin{bmatrix} \tau^{j}_{\alpha} \\ \tau^{j}_{\beta} \end{bmatrix} \tag{22}$$

$$\theta_0 = -\mathrm{sgn}(A) \cdot \theta_1 \tag{23}$$

where $\theta_1$ is a pre-specified constant and the rotation direction of $\theta_0$ is opposite to $\theta_j$ .

### 3.5. Decision process

The end of the algorithm is set according to a pre-defined number of cycles, each of which contains a fixed number of steps. And in the end, a binary decision is made at each pixel location to determine whether it is on the edge or not, by applying a threshold $T$ on the final pheromone matrix. A flow chart of the proposed algorithm can be illustrated in Figure 2.

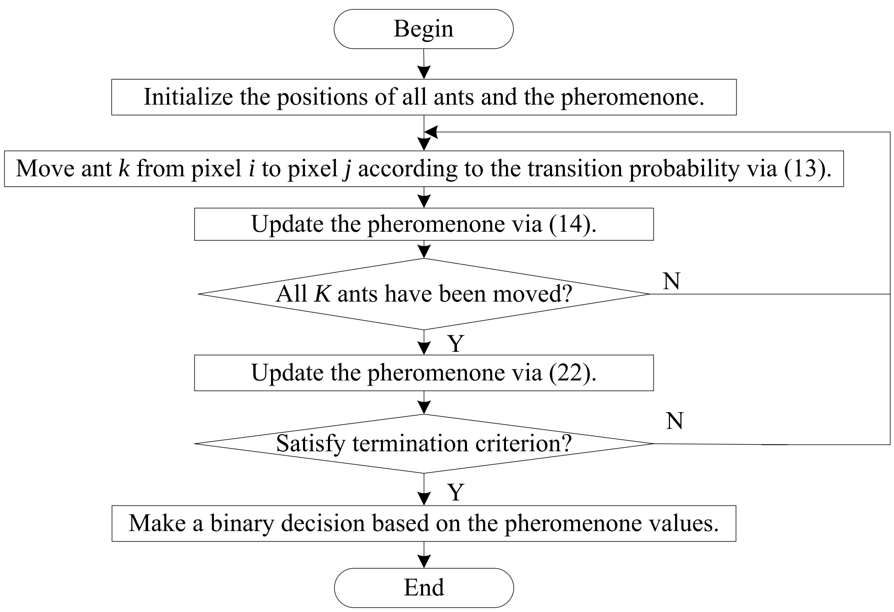



**Figure 2.** Flow Chart of the Proposed Algorithm.

## 4. Experimental results

The experiments and results are presented in this section. In order to verify the effectiveness and efficiency of the proposed method, the experimental results are compared with the algorithm proposed by Nezamabadi-pour's[20]. To provide a fair comparison, binary images got by the two approaches are thinned using the morphological thinning operation [21]. The test is done under the environment of Intel Pentium Dual 1.6GHz CPU, 1GB RAM, WinXP, MATLAB7.1.

### 4.1. Selection of the parameters.

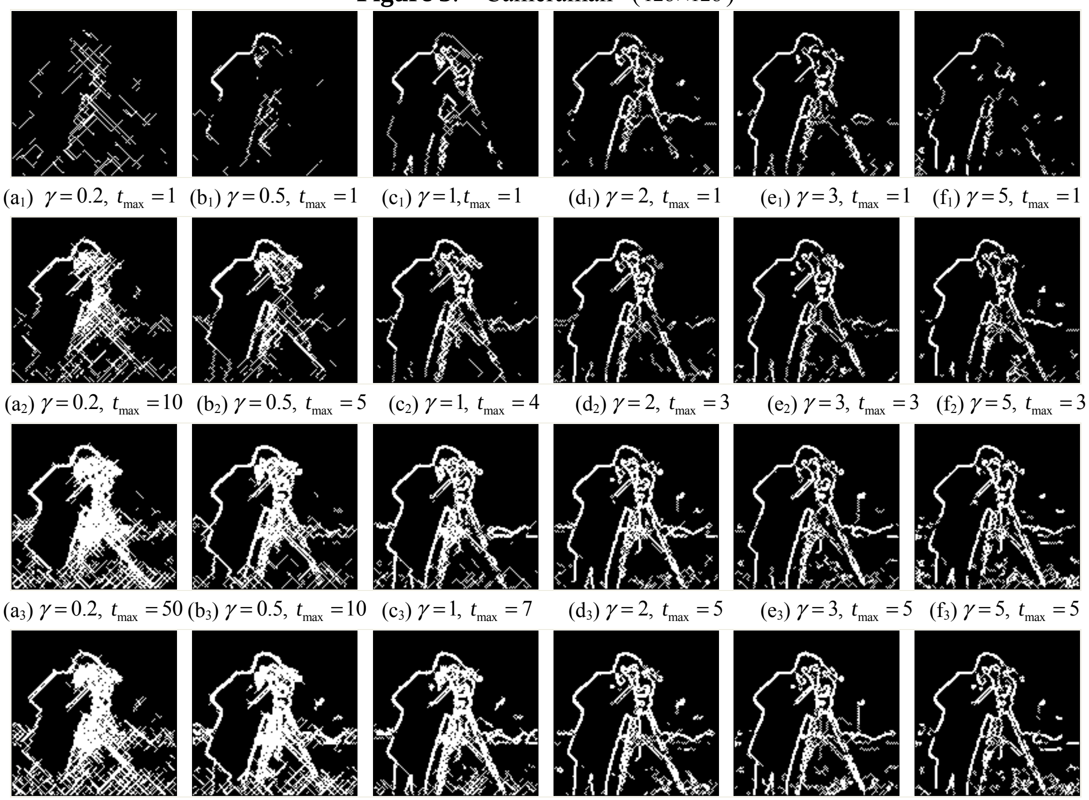
In this study, suitable parameters can be obtained based on a large number of experiments. First, the pheromone is initialized as follows:

$$\tau = \left[\begin{array}{c|c|c|c} \tau^1_\alpha & \tau^2_\alpha & \cdots\cdots & \tau^m_\alpha \\ \tau^2_\beta & \tau^2_\beta & \cdots\cdots & \tau^m_\beta \end{array}\right] = \left[\begin{array}{c|c|c|c} 1/\sqrt{2} & 1/\sqrt{2} & \cdots\cdots & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} & \cdots\cdots & 1/\sqrt{2} \end{array}\right]$$

The parameter $\gamma$ reflects the relative importance during the process of ants' search guided by the heuristic factor. The higher value can accelerate the convergence speed, but the random could decline, and easily falling into local optimum too, and vice versa.

Figure 3 shows the test image “Cameraman” ($128\times128$). The step length of each ant is set as $L=200$. And as can be seen from Figure 4, when $\gamma<1$, the convergence speed is slow, and the desired results haven’t been obtained after 100 times iteration when $\gamma=0.2$, with the increasing of $\gamma$, the faster convergence speed can be got. And the desired result can be obtained after 5 times iteration when $\gamma\geq 2$. Considering the detection performance and the runtime, $\gamma$ is set to be $\gamma=2.5$, and the number of iteration is set as $t_{\max}=5$.

**Figure 3**. “Cameraman” ($128\times128$)



(a1) $\gamma=0.2,\ t_{\max}=1$ (b1) $\gamma=0.5,\ t_{\max}=1$ (c1) $\gamma=1, t_{\max}=1$ (d1) $\gamma=2,\ t_{\max}=1$ (e1) $\gamma=3,\ t_{\max}=1$ (f1) $\gamma=5,\ t_{\max}=1$

(a2) $\gamma=0.2,\ t_{\max}=10$ (b2) $\gamma=0.5,\ t_{\max}=5$ (c2) $\gamma=1,\ t_{\max}=4$ (d2) $\gamma=2,\ t_{\max}=3$ (e2) $\gamma=3,\ t_{\max}=3$ (f2) $\gamma=5,\ t_{\max}=3$

(a3) $\gamma=0.2,\ t_{\max}=50$ (b3) $\gamma=0.5,\ t_{\max}=10$ (c3) $\gamma=1,\ t_{\max}=7$ (d3) $\gamma=2,\ t_{\max}=5$ (e3) $\gamma=3,\ t_{\max}=5$ (f3) $\gamma=5,\ t_{\max}=5$

(a4) $\gamma=0.2,\ t_{\max}=100$ (b4) $\gamma=0.5,\ t_{\max}=30$ (c4) $\gamma=1,\ t_{\max}=10$ (d4) $\gamma=2,\ t_{\max}=7$ (e4) $\gamma=3, t_{\max}=7$ (f4) $\gamma=5, t_{\max}=7$

**Figure 4.** Edge Images (not be thinned) with $\gamma=0.2,\ 0.5,\ 1,\ 2,\ 3,\ 5$ Respectively and Different Iteration Times $t_{\max}$.

Figure 5 shows the original images and edge images with different values of the step length $L$. As can be seen from Figure 5, images from the top row to the bottom row are original image and the edge images with different size of $128\times128$, $256\times256$, $512\times512$, respectively. The desired results can be got when $L\geq120$ ($128\times128$), $L\geq150$ ($256\times256$), $L\geq200$ ($512\times512$), respectively. The smaller size of the image is, the smaller value of the step length can be got. The step length $L$ is set as:

$$L=\begin{cases}150, & \text{if size} = 128\times128\\ 200, & \text{if size} = 256\times256\\ 300, & \text{if size} = 512\times512\end{cases}$$

The other parameters of the proposed algorithm are set as: $q_0=0.9$, $h=0.5$, $\theta_1=0.01\pi$.

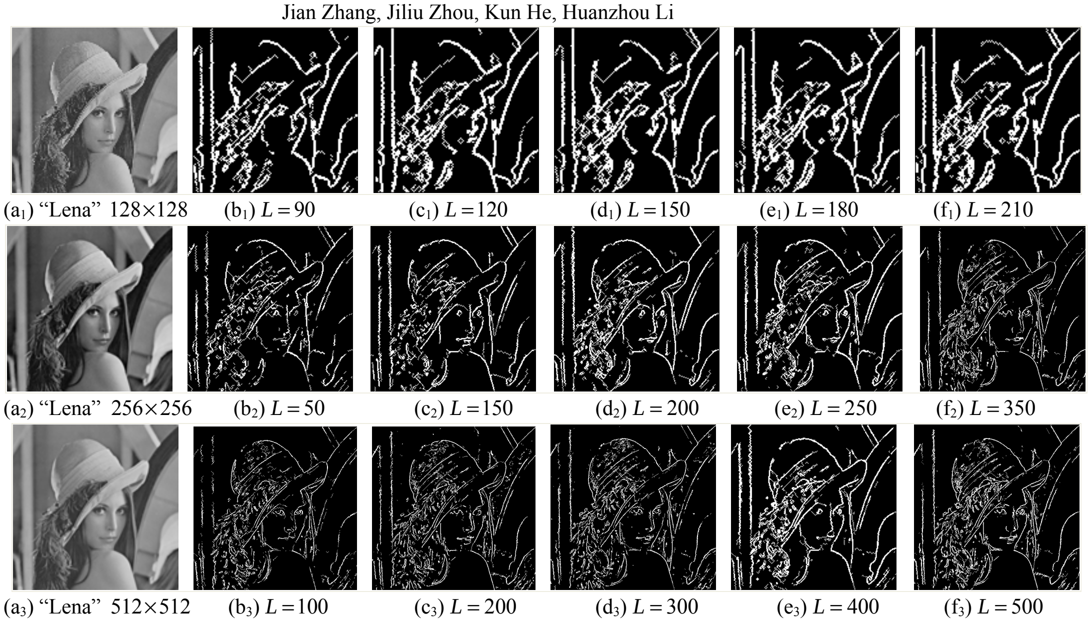



**Figure 5.** Original Images and Edge Images (not be thinned) with Different Values of the Step Length $L$.

## 4.2. Comparison of the different ACO-based edge detection algorithms.

In this section, the proposed algorithm is compared with the Nezamabadi-pour's method, one of the early best papers on edge detection using the ACO approaches. Three images in different sizes ($128\times128$, $256\times256$, $512\times512$) are used. The runtimes are presented as the Table 1. Every result is the average value got by 10 times computation. As can be seen from Table 1, the runtime of the proposed method is far less than the Nezamabadi-pour's.

Figure 6 shows the edge detection results got by Canny operator, Nezamabadi-pour's method and the proposed algorithm. The images from the top row to the bottom row are original images, the edge images got by the Canny detector, the Nezamabadi-pour's approach and the proposed algorithm. And all of the edge images are thinned by the morphology method.

In addition to the visual comparison, the evaluation function can used to be a quantitative indicator to evaluate the performance of the edge detection algorithm. Generally, the methods based on priori knowledge of the image, on human vision and on purely theory are three main ones used to evaluate the performance [22]. The priori knowledge-based method is adopted by this study. Compared with the traditional differential operator, Canny operator has the advantage of higher signal noise ratio and high detection precision [23,24] the edge extracted by Canny operator from clear image is used as the priori knowledge, and the evaluation function $f$ can be described by

$$f = \frac{2\times p(A/B)\times p(B/A)}{p^2(A/B)+p^2(B/A)} \tag{25}$$

In the formula (25), $A$ stands for the edge information detected. $B$ represents the priori knowledge. The term $p(A/B)$ represents the probability that the edge points of priori knowledge are correctly detected. The greater the value is, the smaller the false dismissal probability could be. The term $p(B/A)$ stands for the probability by which the consistency between edge points detected by the above method and the priori knowledge is reflected. The greater the value is, the smaller the false detection probability could be. Evaluation function can well reflect the difference between the edge points extracted by the method and the priori knowledge. The greater the value of the evaluation function $f$ is, the closer the edge extracted by the method is to the priori knowledge. The evaluation function values of the Nezamabadi-pour's method and the proposed algorithm are presented in Table 2. And seen from Table 2, the values of the proposed algorithm are higher than the Nezamabadi-pour's method.

**Table 1.** Comparison of the Runtimes.

| size | Nezamabadi-pour's method (s) | The proposed method (s) |
|---|---|---|
| $128\times128$ | 126.924591 | 13.107387 |
| $256\times256$ | 620.260533 | 39.375912 |
| $512\times512$ | 7658.610764 | 606.600907 |

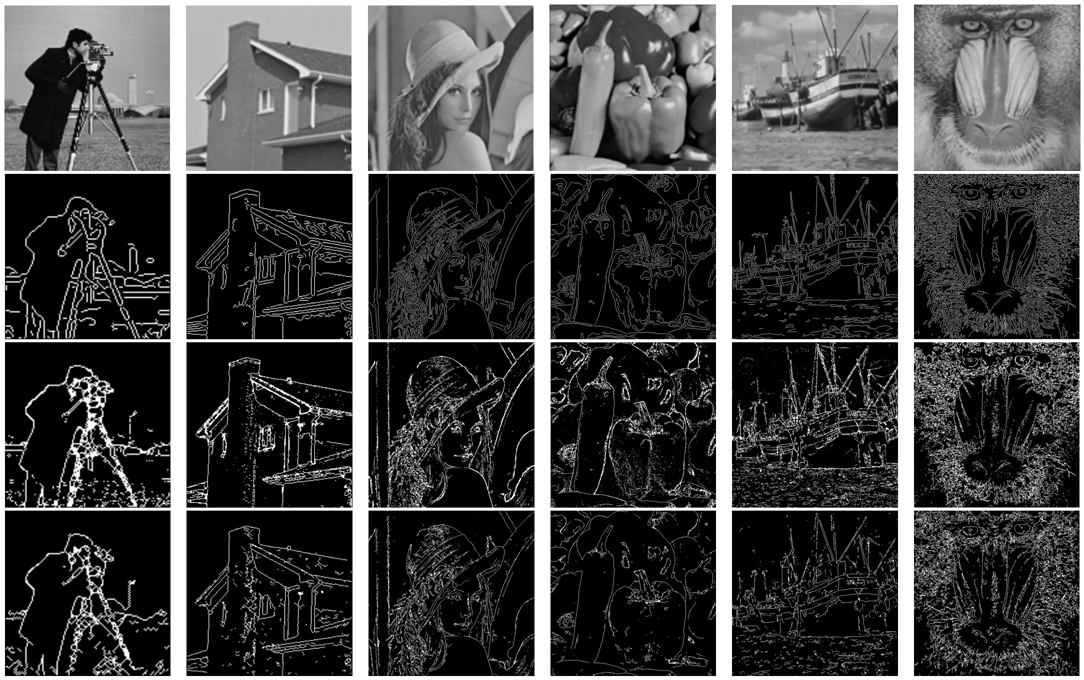


**Figure 6.** Original Images and Edge Images.

**Table 2**. Comparison of the Evaluation Function.

| Image | Nezamabadi-pour's method | The proposed method |
|---|---|---|
| Cameraman ( $128\times128$ ) | 0.9758 | 0.9832 |
| House ( $256\times256$ ) | 0.9794 | 0.9857 |
| Lena ( $512\times512$ ) | 0.9857 | 0.9976 |
| Peppers( $512\times512$ ) | 0.9912 | 0.9314 |
| Boat( $512\times512$ ) | 0.9902 | 0.9959 |
| Baboon( $512\times512$ ) | 0.9846 | 0.9988 |

## 5. Conclusions

In this paper, the QACO-based edge detection algorithm is presented. Quantum theory is without any doubt one of the greatest scientific achievements of the 20th century. Quantum ant colony optimization algorithm is a new type of bionic intelligent optimization algorithm, although the history of this study is not long, and the rigorous theoretical basis has not yet been established. Experimental results have shown that the proposed method is effective and feasible, and the runtime of the proposed algorithm is significantly reduced compared with the traditional methods. This study is attempt to verify the feasibility of the QACO-edge detection approach, so the pixel gradient is chose as the edge feature which is simplest. The future work is to investigate how to improve the performance of QACO and apply to other research areas.

## Acknowledgments

This work is supported by the CALIS construction project of the Ministry of Education of China (03-3204). The authors would like to thank the reviewers for valuable comments.

## References

[1] S. Mallat, S. Zhong, "Characterizations of signals from multiscale edges", IEEE Trans. on Pattern Anal. and Machine Intelligence, vol. 14, no. 7, pp. 710-732, 1992.
[2] D. Marr, E. Hildreth, "Theory of edge detection", Proc. R. Soc. Lond. B, vol. 207, no. 1667, pp. 187-217, 1980.
[3] Jizhao Hua, , Jianguo Wang, Heqiang Peng, Jingyu Yang, "A novel edge detection method based on PCA1", IJACT, vol. 3, no. 3, pp. 228-238, 2011.
[4] S. Berlemont, J.-C. Olivo-Marin, "Combining local filtering and multiscale analysis for edge, ridge, and curvilinear objects detection", IEEE Trans. on Image Processing, vol.19, no.1, pp.74-84, 2010.
[5] S. Janakiraman, V. Vasudevan, "ACO based distributed intrusion detection system", JDCTA, vol. 3, no. 1, pp. 66-72, 2009.
[6] De-Sian Lu, Chien-Chang Chen, "Edge detection improvement by ant colony optimization", Pattern Recognition Letters, vol. 29, no. 4, pp.416-425,2008.
[7] Mingsheng Ying, "Quantum computation, quantum theory and AI", Artificial intelligence, vol. 174, no. 2, pp.162-176, 2010.
[8] Licheng Jiao, et al., "Advances in natural computation", Machine learning and image understanding, Xidian University Press, CHN, 2008.
[9] K. Meng, H. G. Wang, Z.Y. Dong, K. P. Wong, "Quantum particle swarm optimization for valve-point economic load dispatch", IEEE Trans. on Power Systems, vol. 25, no. 1, pp. 215-222,2010.
[10] S.-S. Kim, K.-C. Kwak, "Development of quantum-based adaptive neuro-fuzzy networks", IEEE Trans. Systems, Man, and Cybernetics, vol. 40, no. 1, pp. 91-100, 2010.
[11] Xiaoming You, Xingwai Miao, Sheng Liu, "Quantum computing-based ant colony optimization algorithm for TSP", 2nd Int. Conf. on Power Elec. and Intelligent Transportation Sys., pp.359-362, 2009.
[12] Lei Yang, Jue Wu, "Quantum ant colony optimization algorithm on collision detection", Int. Conf. on computational and information sciences, pp. 341-344, 2011.
[13] R.J. Mullen, D. Monekosso, S. Barman, and P. Remagnino, "A review of ant algorithms", Expert systems with Applications, vol. 36, no. 6, pp. 9608-9617, 2009.
[14] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents", IEEE Trans. on Systems, Man and Cybernetics, Part B, vol. 26, no. 1, pp. 29-41,1996.
[15] C. Blum, "Ant colony optimization: introduction and recent trends", Physics of Life Reviews, vol. 2, no. 4, pp. 353–373, 2005.
[16] Jianhua Xiao, Jin Xu, Zhihua Chen, Kai Zhang and Linqiang Pan, "A hybrid quantum chaotic swarm evolutionary algorithm for DNA encoding", Computers and Mathematics with Applications, vol. 57, no. 11-12, pp. 1949-1958, 2009.
[17] V. Ramos and F. Almeida, "Artificial ant colonies in digital image habitats – a mass behaviour effect study on pattern recognition", Arxiv: cs/0412086, 2004.
[18] D. R. Chialvo and M. M. Millonas, "How swarms build cognitive maps", In Luc Steels, Ed., The Biology and Technology of Intelligent Autonomous Agents, pp. 439-450, 1995.
[19] Yuguang Li, Junsheng Zhao, Yuanping Zhang, "Improved quantum ant colony algorithm for TSP", Computer engineering and design, vol. 30, no. 16, pp. 3843-3845,2009.
[20] H. Nezamabadi-pour, S. Saryazdi and E. Rashedi, "Edge detection using ant algorithms", Soft Computing, vol. 10, no. 7, pp. 623–628, 2006.
[21] R. C. Gonzalez, R. E. Woods, "Digital image processing", Publishing house of electronics industry, 2003.
[22] Yitzhak Yitzhaky, Eli Peli, "A method for objective edge detection evaluation and detector parameter selection", vol. 25, no. 8, IEEE Trans. on Pattern Analysis and Machine Intelligence, pp. 1027-1033, 2003
[23] Zhang Yang, P. I. Rockett, "he Bayesian operating point of the canny edge detector", IEEE Trans. on Image Processing., vol. 15, no. 11, pp. 3409-3416, 2006.
[24] Lijun Ding, Ardeshir Goshtasby, "On the canny edge detector", Pattern Recognition, vol. 34, no. 3, pp.721-725, 2001.