An Ant Colony Optimization Algorithm For Image Edge Detection

Jing Tian, Weiyu Yu, and Shengli Xie

***Abstract*— *Ant colony optimization* (ACO) is an optimization algorithm inspired by the natural behavior of ant species that ants deposit pheromone on the ground for foraging. In this paper, ACO is introduced to tackle the image edge detection problem. The proposed ACO-based edge detection approach is able to establish a pheromone matrix that represents the edge information presented at each pixel position of the image, according to the movements of a number of ants which are dispatched to move on the image. Furthermore, the movements of these ants are driven by the local variation of the image's intensity values. Experimental results are provided to demonstrate the superior performance of the proposed approach.**

I. Introduction

ANT colony optimization (ACO) is a nature-inspired optimization algorithm [1], [2], motivated by the natural phenomenon that ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony. The first ACO algorithm, called the *ant system*, was proposed by Dorigo *et al.* [3]. Since then, a number of ACO algorithms have been developed [4], such as the *Max-Min ant system* [5] and the *ant colony system* [6]. ACO has been widely applied in various problems [7]–[16].

In this paper, ACO is introduced to tackle the image edge detection problem, where the aim is to extract the edge information presented in the image, since it is crucial to understand the image's content [17]. The proposed approach exploits a number of ants, which move on the image driven by the local variation of the image's intensity values, to establish a pheromone matrix, which represents the edge information at each pixel location of the image.

To the best of our knowledge, there has been very little research work on the problem addressed in this paper except [18], [19]. However, there are fundamental differences between our proposed approach and theirs. First, our proposed approach exploits the *ant colony system* [6]; on the contrary, Nezamabadi-Pour *et al.*'s method [18] exploits the *ant system* [3]. It has been shown that the above fundamental difference is crucial to the respective designed ACO-based algorithms [4]. Second, ACO is exploited to 'directly' extract the edge information in our proposed method, in contrast to that ACO serves as a 'post-processing' in [19] to enhance the edge information that has already been extracted by conventional edge detection algorithms.

The paper is organized as follows. In Section II, a brief introduction is provided to present the fundamental concepts of ACO. Then, an ACO-based image edge detection approach is proposed in Section III. Experimental results are presented in Section IV. Finally, Section V concludes this paper.

Jing Tian (email: eejtian@gmail.com). Weiyu Yu, and Shengli Xie are with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, P. R. China 510641 (email: yuweiyu@scut.edu.cn).

II. Ant Colony Optimization

ACO aims to iteratively find the optimal solution of the target problem through a guided search (i.e., the movements of a number of ants) over the solution space, by constructing the *pheromone* information. To be more specific, suppose totally K ants are applied to find the optimal solution in a space χ that consists of $M_1 \times M_2$ nodes, the procedure of ACO can be summarized as follows [4].

- Initialize the positions of totally K ants, as well as the pheromone matrix $\boldsymbol{\tau}^{(0)}$.
- For the construction-step index $n = 1 : N$,
  - For the ant index $k = 1 : K$,
    - Consecutively move the k-th ant for L steps, according to a probabilistic transition matrix $\boldsymbol{p}^{(n)}$ (with a size of $M_1M_2 \times M_1M_2$).
  - Update the pheromone matrix $\boldsymbol{\tau}^{(n)}$.
- Make the solution decision according to the final pheromone matrix $\boldsymbol{\tau}^{(N)}$.

There are two fundamental issues in the above ACO process; that is, the establishment of the probabilistic transition matrix $\boldsymbol{p}^{(n)}$ and the update of the pheromone matrix $\boldsymbol{\tau}^{(n)}$, each of which is presented in detail as follow, respectively.

First, at the n-th construction-step of ACO, the k-th ant moves from the node i to the node j according to a probabilistic action rule, which is determined by [4]

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

where $\tau_{i,j}^{(n-1)}$ is the pheromone information value of the arc linking the node i to the node j; Ω_i is the neighborhood nodes for the ant a_k given that it is on the node i; the constants α and β represent the influence of pheromone information and heuristic information, respectively; $\eta_{i,j}$ represents the heuristic information for going from node i to node j, which is fixed to be same for each construction-step.

Second, the pheromone matrix needs to be updated twice during the ACO procedure. The first update is performed after the movement of *each* ant within each construction-step. To be more specific, after the move of the k-th ant within the n-th construction-step, the pheromone matrix is

978-1-4244-1823-7/08/$25.00©2008 IEEE

updated as [4]

$$\tau_{i,j}^{(n-1)} = \begin{cases} (1-\rho)\cdot\tau_{i,j}^{(n-1)} + \rho\cdot\Delta_{i,j}^{(k)}, & \text{if } (i,j) \text{ belongs to the } \textit{best tour}; \\ \tau_{i,j}^{(n-1)}, & \text{otherwise.} \end{cases} \quad (2)$$

where ρ is the *evaporation rate*. Furthermore, the determination of *best tour* is subject to the user-defined criterion, it could be either the best tour found in the current construction-step, or the best solution found since the start of the algorithm, or a combination of both of the above two [4]. The second update is performed after the move of *all* K ants within each construction-step; and the pheromone matrix is updated as [4]

$$\boldsymbol{\tau}^{(n)} = (1-\psi)\cdot\boldsymbol{\tau}^{(n-1)} + \psi\cdot\boldsymbol{\tau}^{(0)}, \quad (3)$$

where ψ is the *pheromone decay coefficient*. Note that the ant colony system [6] performs two update operations (i.e, (2) and (3)) for updating the pheromone matrix, while the ant system [3] only performs one operation (i.e., (3)).

III. Proposed ACO-based Image Edge Detection Approach

The proposed ACO-based image edge detection approach aims to utilize a number of ants to move on a 2-D image for constructing a pheromone matrix, each entry of which represents the edge information at each pixel location of the image. Furthermore, the movements of the ants are steered by the local variation of the image's intensity values.

The proposed approach starts from the *initialization process*, and then runs for N iterations to construct the pheromone matrix by iteratively performing both the *construction process* and the *update process*. Finally, the *decision process* is performed to determine the edge. Each of these process is presented in detail as follows, respectively.

A. Initialization Process

Totally K ants are randomly assigned on an image $\mathbf{I}$ with a size of $M_1 \times M_2$, each pixel of which can be viewed as a node. The initial value of each component of the pheromone matrix $\boldsymbol{\tau}^{(0)}$ is set to be a constant τ_{init}.

B. Construction Process

At the n-th construction-step, one ant is randomly selected from the above-mentioned total K ants, and this ant will consecutively move on the image for L movement-steps. This ant moves from the node (l,m) to its neighboring node (i,j) according to a transition probability that is defined as

$$p_{(l,m),(i,j)}^{(n)} = \frac{\left(\tau_{i,j}^{(n-1)}\right)^{\alpha}(\eta_{i,j})^{\beta}}{\sum_{(i,j)\in\Omega_{(l,m)}}\left(\tau_{i,j}^{(n-1)}\right)^{\alpha}(\eta_{i,j})^{\beta}}, \quad (4)$$

where $\tau_{i,j}^{(n-1)}$ is the pheromone value of the node (i,j), $\Omega_{(l,m)}$ is the neighborhood nodes of the node (l,m), $\eta_{i,j}$ represents the heuristic information at the node (i,j). The constants α and β represent the influence of the pheromone matrix and the heuristic matrix, respectively.

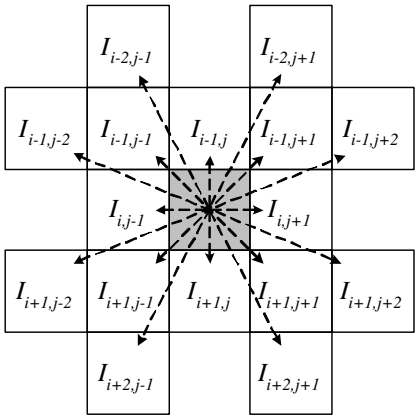

Fig. 1. A local configuration at the pixel position $I_{i,j}$ for computing the variation $V_c(I_{i,j})$ defined in (6). The pixel $I_{i,j}$ is marked as gray square.

There are two crucial issues in the construction process. The first issue is the determination of the heuristic information $\eta_{i,j}$ in (4). In this paper, it is proposed to be determined by the local statistics at the pixel position (i,j) as

$$\eta_{i,j} = \frac{1}{Z}V_c(I_{i,j}), \quad (5)$$

where $Z = \sum_{i=1:M_1}\sum_{j=1:M_2}V_c(I_{i,j})$, which is a normalization factor, $I_{i,j}$ is the intensity value of the pixel at the position (i,j) of the image $\mathbf{I}$, the function $V_c(I_{i,j})$ is a function of a local group of pixels c (called the *clique*), and its value depends on the variation of image's intensity values on the clique c (as shown in Figure 1). More specifically, for the pixel $I_{i,j}$ under consideration, the function $V_c(I_{i,j})$ is

$$\begin{aligned} V_c(I_{i,j}) = f\,(&|I_{i-2,j-1} - I_{i+2,j+1}| + |I_{i-2,j+1} - I_{i+2,j-1}| \\ &+|I_{i-1,j-2} - I_{i+1,j+2}| + |I_{i-1,j-1} - I_{i+1,j+1}| \\ &+|I_{i-1,j} - I_{i+1,j}| + |I_{i-1,j+1} - I_{i-1,j-1}| \\ &+|I_{i-1,j+2} - I_{i-1,j-2}| + |I_{i,j-1} - I_{i,j+1}|)\,. \end{aligned} \quad (6)$$

To determine the function $f(\cdot)$ in (6), the following four functions are considered in this paper; they are mathematically expressed as follows and illustrated in Figure 2, respectively.

$$f(x) = \lambda x, \quad \text{for} \quad x \geq 0, \quad (7)$$

$$f(x) = \lambda x^2, \quad \text{for} \quad x \geq 0, \quad (8)$$

$$f(x) = \begin{cases} \sin\left(\frac{\pi x}{2\lambda}\right) & 0 \leq x \leq \lambda; \\ 0 & \text{else.} \end{cases} \quad (9)$$

$$f(x) = \begin{cases} \frac{\pi x \sin(\frac{\pi x}{\lambda})}{\lambda} & 0 \leq x \leq \lambda; \\ 0 & \text{else.} \end{cases} \quad (10)$$

The parameter λ in each of above functions (7)-(10) adjusts the functions' respective shapes.

The second issue is to determine the permissable range of the ant's movement (i.e., $\Omega_{(l,m)}$ in (4)) at the position (l,m). In this paper, it is proposed to be either the 4-connectivity neighborhood or the 8-connectivity neighborhood, both of which are demonstrated in Figure 3.

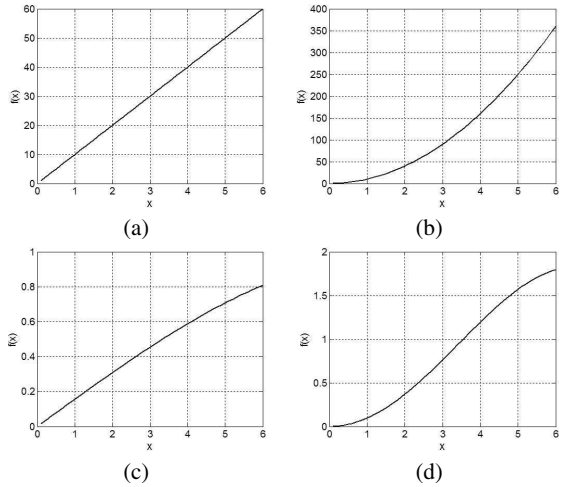

Fig. 2. Various functions with the parameter $\lambda = 10$: (a) the function defined in (7); (b) the function defined in (8); (c) the function defined in (9); and (d) the function defined in (10).

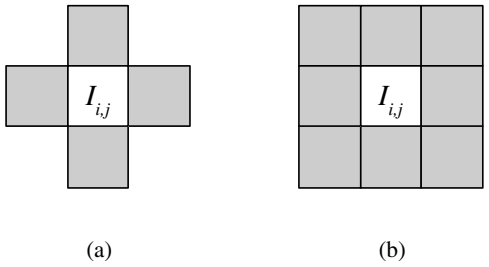

Fig. 3. Various neighborhoods (marked as gray regions) of the pixel $I_{i,j}$: (a) 4-connectivity neighborhood; and (b) 8-connectivity neighborhood.

C. Update Process

The proposed approach performs two updates operations for updating the pheromone matrix.

- The first update is performed after the movement of each ant within each construction-step. Each component of the pheromone matrix is updated according to

$$\tau_{i,j}^{(n-1)} = \begin{cases} (1-\rho)\cdot\tau_{i,j}^{(n-1)} + \rho\cdot\Delta_{i,j}^{(k)}, & \text{if } (i,j) \text{ is visited by the current } k\text{-th ant;} \\ \tau_{i,j}^{(n-1)}, & \text{otherwise.} \end{cases} \tag{11}$$

  where ρ is defined in (2), $\Delta_{i,j}^{(k)}$ is determined by the heuristic matrix; that is, $\Delta_{i,j}^{(k)} = \eta_{i,j}$.
- The second update is carried out after the movement of all ants within each construction-step according to

$$\boldsymbol{\tau}^{(n)} = (1-\psi)\cdot\boldsymbol{\tau}^{(n-1)} + \psi\cdot\boldsymbol{\tau}^{(0)}, \tag{12}$$

  where ψ is defined in (3).

D. Decision Process

In this step, a binary decision is made at each pixel location to determine whether it is edge or not, by applying a threshold T on the final pheromone matrix $\boldsymbol{\tau}^{(N)}$. In this paper, the above-mentioned T is proposed to be adaptively computed based on the method developed in [20].

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 $T^{(0)}$. 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 (in terms of a user-defined tolerance ϵ). The above iterative procedure can be summarized as follows.

Step 1: Initialize $T^{(0)}$ as

$$T^{(0)} = \frac{\sum_{i=1:M_1}\sum_{j=1:M_2}\tau_{i,j}^{(N)}}{M_1 M_2}, \tag{13}$$

and set the iteration index as $l = 0$.

Step 2: Separate the pheromone matrix $\boldsymbol{\tau}^{(N)}$ into two class using $T^{(l)}$, where the fist class consists entries of $\boldsymbol{\tau}$ that have smaller values than $T^{(l)}$, while the second class consists the rest entries of $\boldsymbol{\tau}$. Next, calculate the mean of each of the above two categories via

$$m_L^{(l)} = \frac{\sum_{i=1:M_1}\sum_{j=1:M_2} g^L_{T^{(l)}}(\tau_{i,j}^{(N)})}{\sum_{i=1:M_1}\sum_{j=1:M_2} h^L_{T^{(l)}}(\tau_{i,j}^{(N)})}, \tag{14}$$

$$m_U^{(l)} = \frac{\sum_{i=1:M_1}\sum_{j=1:M_2} g^U_{T^{(l)}}(\tau_{i,j}^{(N)})}{\sum_{i=1:M_1}\sum_{j=1:M_2} h^U_{T^{(l)}}(\tau_{i,j}^{(N)})}, \tag{15}$$

where

$$g^L_{T^{(l)}}(x) = \begin{cases} x, & \text{if } x \le T^{(l)}; \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

$$h^L_{T^{(l)}}(x) = \begin{cases} 1, & \text{if } x \le T^{(l)}; \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

$$g^U_{T^{(l)}}(x) = \begin{cases} x, & \text{if } x \ge T^{(l)}; \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

$$h^U_{T^{(l)}}(x) = \begin{cases} 1, & \text{if } x \ge T^{(l)}; \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

Step 3: Set the iteration index $l = l + 1$, and update the threshold as

$$T^{(l)} = \frac{m_L^{(l)} + m_U^{(l)}}{2}. \tag{20}$$

Step 4: If $|T^{(l)} - T^{(n-1)}| > \epsilon$, then go to *Step 2*; otherwise, the iteration process is terminated and a binary decision is made on each pixel position (i,j) to determine whether it is edge (i.e., $E_{i,j} = 1$) or not (i.e., $E_{i,j} = 0$), based on the criterion

$$E_{i,j} = \begin{cases} 1, & \text{if } \tau_{i,j}^{(N)} \ge T^{(l)}; \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

E. The Summary of The Proposed Approach

A summary of the implementation of the proposed approach is presented in Figure 4.

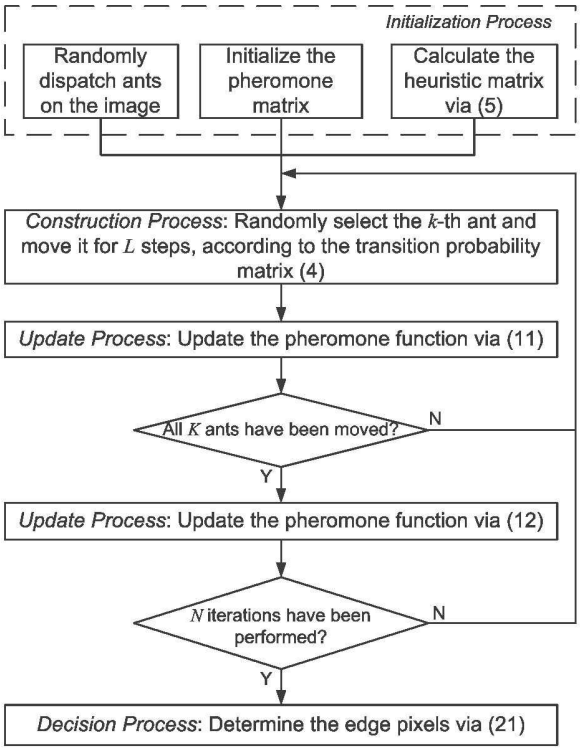

Fig. 4. A summary of the implementation of the proposed ACO-based image edge detection approach.

IV. Experimental Results

Experiments are conducted to evaluate the performance of the proposed approach using four test images, $Camera$, $House$, $Lena$, and $Pepper$, which are shown in Figure 5. Furthermore, various parameters of the proposed approach are set as follows.

- $K = \lfloor \sqrt{M_1 \times M_2} \rfloor$: the total number of ants, where the function $\lfloor x \rfloor$ represents the highest integer value that is smaller than or equals to x.
- $\tau_{init} = 0.0001$: the initial value of each component of the pheromone matrix.
- $\alpha = 1$: the weighting factor of the pheromone information in (4).
- $\beta = 0.1$: the weighting factor of the heuristic information in (4).
- $\Omega = 8$-connectivity neighborhood: the permissable ant's movement range in (4), as shown in Figure 3 (b).
- $\lambda = 1$: the adjusting factor of the functions in (7)-(10).
- $\rho = 0.1$: the evaporation rate in (11).
- $L = 40$: total number of ant's movement-steps within each construction-step.
- $\psi = 0.05$: the pheromone decay coefficient in (12).
- $\epsilon = 0.1$: the user-defined tolerance value used in the decision process of the proposed method.
- $N = 4$: total number of construction-steps.

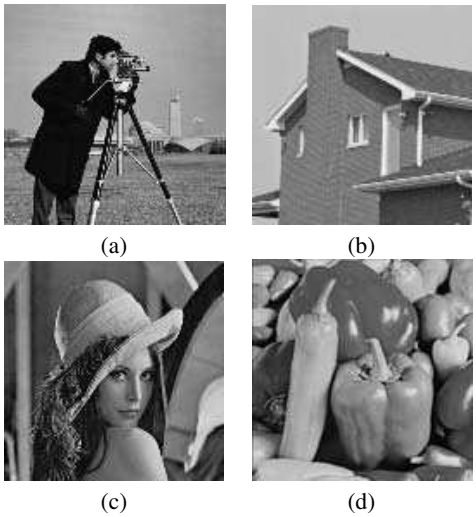

Fig. 5. Test images used in this paper: (a) $Camera$ (128×128); (b) $House$ (128×128); (c) $Lena$ (128×128); and (d) $Pepper$ (128×128).

The determination of above parameters are critical to the performance of the proposed approach; this issue will be reported elsewhere.

Experimental results are provided to compare the proposed approach with Nezamabadi-Pour *et al.*'s edge detection method [18]. To provide a fair comparison, the morphological thinning operation of [18] is neglected, since it is performed as a post-processing to further refine the edge information that is extracted by ACO [18]. Furthermore, to present how the determination of the heuristic matrix (i.e., (5)) is crucial to the proposed method, various functions defined in (7)-(10) are individually incorporated into (6) of the proposed approach, and their resulted performances are presented. Figures 6, 7, 8 and 9 present the results of test images $Camera$, $House$, $Lena$ and $Pepper$, respectively. As seen from above Figures, the proposed approach always outperforms Nezamabadi-Pour *et al.*'s method [18], in terms of visual quality of the extracted edge information.

The proposed ACO-based edge detection approach is implemented using the $Matlab$ programming language and run on a PC with a Intel Core^{TM} DUO 2.13 GHz CPU and a 1 GB RAM. The computational times of the proposed approach are 64.90 seconds, 64.15 seconds, 64.46 seconds and 64.57 seconds, for the test image $Camera$, with the incorporation of the functions in (7)-(10), respectively.

V. Conclusions

In this paper, an ACO-based image edge detection approach has been successfully developed. The proposed approach yields superior subjective performance to that of the existing edge detection algorithm [18], as verified in our experiments. Furthermore, the parallel ACO algorithm [21] can be exploited to further reduce the computational load of the proposed algorithm, for future research work.

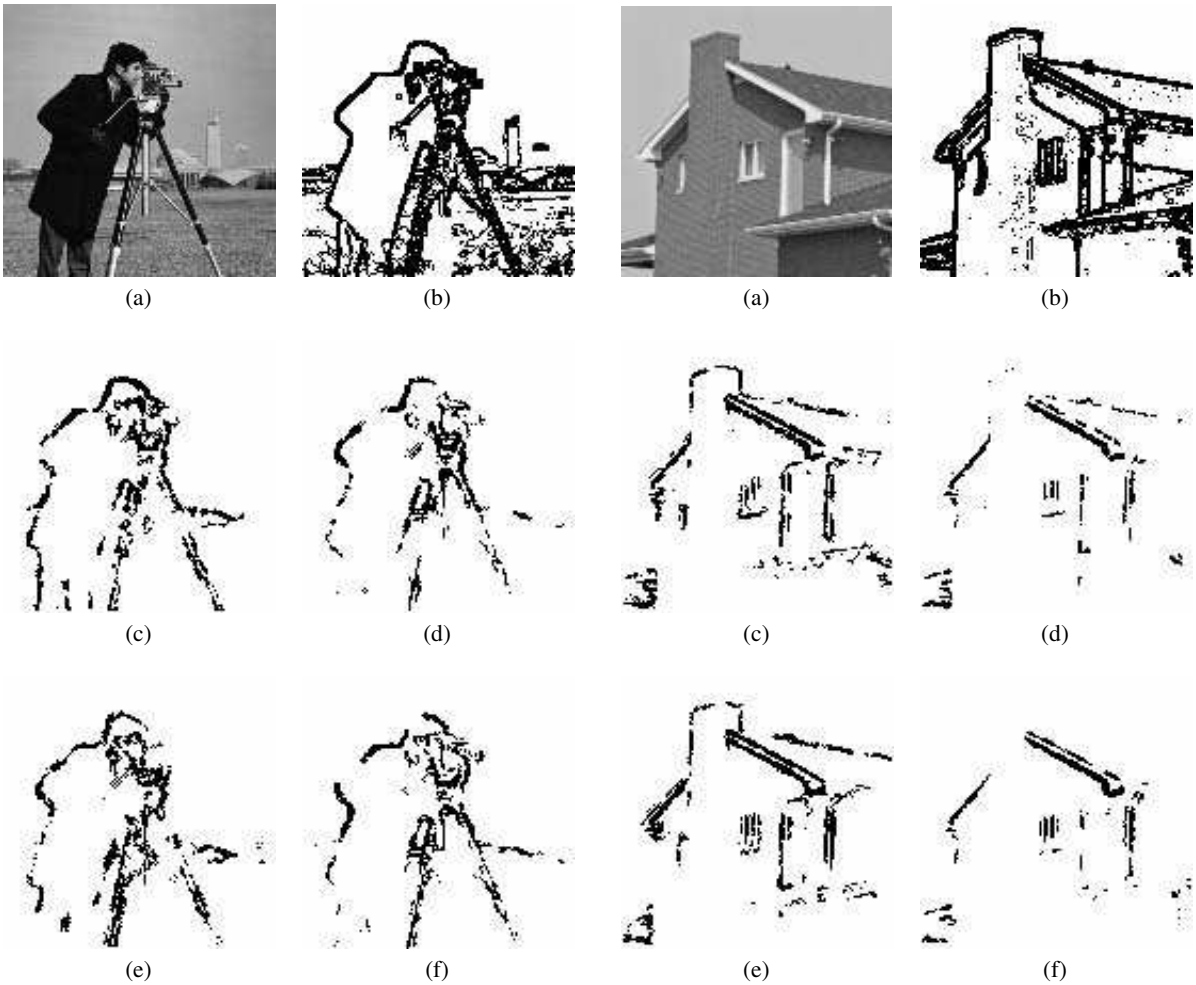

Fig. 6. Various extracted edge information of the test image $Camera$: (a) the original image; (b) Nezamabadi-Pour *et al.*'s method [18]; (c) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (7); (d) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (8); (e) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (9); and (f) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (10).

Fig. 7. Various extracted edge information of the test image $House$: (a) the original image; (b) Nezamabadi-Pour *et al.*'s method [18]; (c) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (7); (d) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (8); (e) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (9); and (f) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (10).

ACKNOWLEDGEMENT

This work was supported by the Key Program of National Natural Science Foundation of China (Grant No. U0635001), the Guangdong Natural Science Foundation (Grant No. 06300098) and the State Key Laboratory of Video and Audio Perception, Peking University (Grant No. 0610).

REFERENCES

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

[2] H.-B. Duan, *Ant Colony Algorithms: Theory and Applications*. Beijing: Science Press, 2005.

[3] 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, pp. 29–41, Feb. 1996.

[4] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, pp. 28–39, Nov. 2006.

[5] T. Stutzle and H. Holger H, "Max-Min ant system," *Future Generation Computer Systems*, vol. 16, pp. 889–914, Jun. 2000.

[6] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Trans. on Evolutionary Computation*, vol. 1, pp. 53–66, Apr. 1997.

[7] M. Dorigo, G. D. Caro, and T. Stutzle, *Special Issue on Ant Algorithms, Future Generation Computer Systems*, vol. 16, Jun. 2000.

[8] O. Cordon, F. Herrera, and T. Stutzle, *Special Issue on Ant Colony Optimization: Models and Applications, Mathware and Soft Computing*, vol. 9, Dec. 2002.

[9] M. Dorigo, L. M. Gambardella, M. Middendorf, and T. Stutzle, *Special Issue on Ant Colony Optimization, IEEE Transactions on Evolutionary Computation*, vol. 6, Jul. 2002.

[10] H. Zheng, A. Wong, and S. Nahavandi, "Hybrid ant colony algorithm for texture classification," in *Proc. IEEE Congress on Evolutionary*

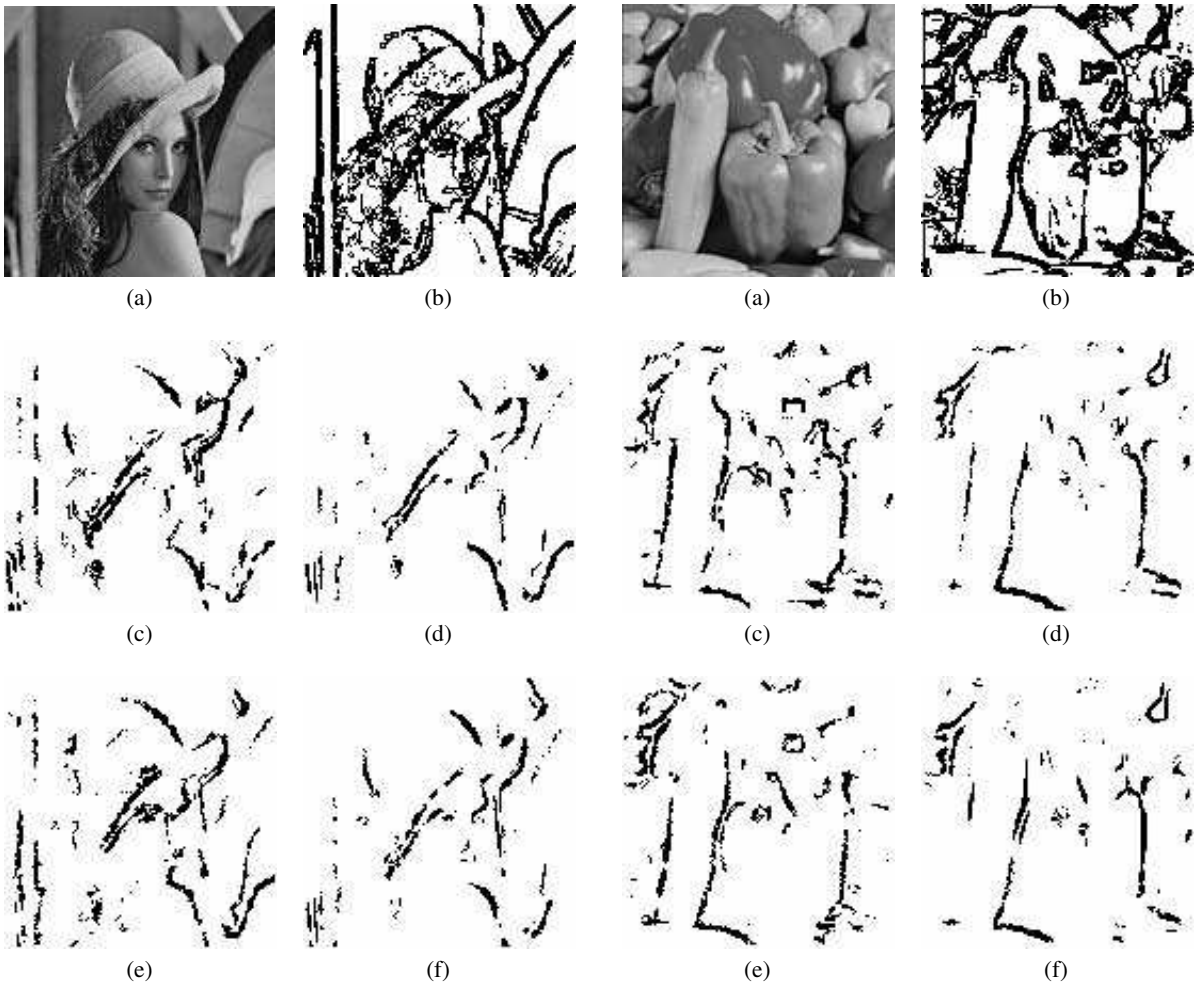

Fig. 8. Various extracted edge information of the test image $Lena$: (a) the original image; (b) Nezamabadi-Pour *et al.*'s method [18]; (c) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (7); (d) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (8); (e) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (9); and (f) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (10).

Fig. 9. Various extracted edge information of the test image $Pepper$: (a) the original image; (b) Nezamabadi-Pour *et al.*'s method [18]; (c) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (7); (d) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (8); (e) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (9); and (f) the proposed ACO-based image edge detection algorithm with the incorporation of the function defined in (10).

Computation, Canberra, Australia, Dec. 2003, pp. 2648–2652.

[11] D. Martens, M. D. Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification with ant colony optimization," *IEEE Trans. on Evolutionary Computation*, vol. 11, pp. 651–665, Oct. 2007.

[12] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Trans. on Evolutionary Computation*, vol. 6, pp. 321–332, Aug. 2002.

[13] S. Ouadfel and M. Batouche, "Ant colony system with local search for Markov random field image segmentation," in *Proc. IEEE Int. Conf. on Image Processing*, Barcelona, Spain, Sep. 2003, pp. 133–136.

[14] S. L. Hegarat-Mascle, A. Kallel, and X. Descombes, "Ant colony optimization for image regularization based on a nonstationary Markov modeling," *IEEE Trans. on Image Processing*, vol. 16, pp. 865–878, Mar. 2007.

[15] A. T. Ghanbarian, E. Kabir, and N. M. Charkari, "Color reduction based on ant colony," *Pattern Recognition Letters*, vol. 28, pp. 1383–1390, Sep. 2007.

[16] A. R. Malisia and H. R. Tizhoosh, "Image thresholding using ant colony optimization," in *Proc. Canadian Conf. on Computer and Robot Vision*, Quebec, Canada, Jun. 2006, pp. 26–26.

[17] R. C. Gonzalez and R. E. Woods, *Digital image processing*. Harlow: Prentice Hall, 2007.

[18] H. Nezamabadi-Pour, S. Saryazdi, and E. Rashedi, "Edge detection using ant algorithms," *Soft Computing*, vol. 10, pp. 623–628, May 2006.

[19] D.-S. Lu and C.-C. Chen, "Edge detection improvement by ant colony optimization," *Pattern Recognition Letters*, vol. 29, pp. 416–425, Mar. 2008.

[20] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, pp. 62–66, Jan. 1979.

[21] M. Randall and A. Lewis, "A parallel implementation of ant colony optimization," *Journal of Parallel and Distributed Computing*, vol. 62, pp. 1421–1432, Sep. 2002.