2009 Second Asia-Pacific Conference on Computational Intelligence and Industrial Applications

An Image Template Matching Method Using Particle Swarm Optimization

Xiaoxiang Liu, Weigang Jiang, Jianwen Xie
Department of Computer Science of Zhuhai College
Jinan University
Zhuhai, Guangdong, 519070, China
tlxx@jnu.edu.cn, jfkdr@163.com, kenxie.4@163.com

Yitian Jia
Department of Accounting of Zhuhai College
Jinan University
Zhuhai, Guangdong, 519070, China
jyt116@hotmail.com

***Abstract*—At present, most of the image template matching algorithms involve large computational complexity, and can hardly be used in practical projects. This paper proposes that particle swarm optimization algorithm (PSO) be used in image template matching problems (ITMP). Template matching in fact is a matter of seeking optimization. The cross-correlation function of template and sub image is set as the objective function, and then a fast template matching algorithm can reached based on particle swarm optimization algorithm. Experiment results prove both the computational accuracy and efficiency of this algorithm.**

***Index Terms*—template matching, formatting, normalized cross correlation function, particle swarm optimization**

I. Introduction

Template matching is an important image recognizing and processing method. In this method, an identified object is taken as a template, and compared with the zones in the original image, so as to recognize the objective. Template matching is widely used in areas such as objective recognition, image search and virtual reality. There are two most used algorithms in template matching. One is to match via picking up the image's unique points (e.g. inflexion points and corner points). The other is to reflect their similarity through comparing the region (round or square template), and their attributes (gray information and frequency-domain analysis and so on). But most of the time it is extremely difficult to collect unique points; therefore the second algorithm has to be considered and taken.

Normalized cross correlation function, as a similarity measure, is frequently used in the second algorithm. It matches the object and the original image with the aid of the relationship between original data like pixel grey value. Integrity exhaustive method is resorted in traditional solving algorithm. However, as the scale of the template increases (including the pixel number and the size of the original image), the time complexity of this traditional algorithm increases on square scale, resulting in quite a few shortages as a large amount of processing information, high computational complexity, long computational time and so on. Thus it is quite necessary and significant to search for an efficient template matching method. Some scholars switch template matching problems into optimization problems and solve them with optimization algorithms. One of them is GA algorithm or revised GA algorithm. But with its complicated encoding, the efficiency and accuracy of GA algorithm can hardly be satisfying.

PSO (Particle Swarm Optimization) [5] is a bionic algorithm which simulates the flight of flocks of birds, making good effect for the various multidimensional continuous space optimization problems. In PSO algorithm, very few parameters are needed to be adjusted, which makes it particularly easy to implement. [6-7]. Similar to GA, PSO starts by initializing a population (i.e., swarm) as random solutions (i.e., particles) and searches for optima by updating generations. Instead of crossover and mutation operators, formulated equations are constructed to update the PSO particles according to their own experience and the best of other particles. In contrast to analytical or general heuristic methods, PSO is efficient in computation and has a great capability of escaping local optima. In addition, PSO has advantages over GA in its easy implementation, its relatively faster search process, and its effective performance. This paper uses PSO algorithm to solve ITMP. The results of the experiment show desirable effect.

II. Image Template Matching

Template matching is comparing the identified template with a region of the same size in the original image. The objective is to find out the optional matching position. At first,

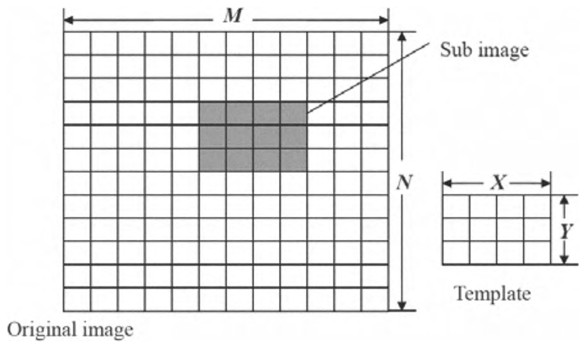

Figure 1. The source image and the template

978-1-4244-4607-0/09/$25.00 ©2009 IEEE PACIIA 2009

the up-left corner point of the template covers the counterpart of the original image, and the template and the region of the same size in the original image are compared. Then the template is moved to the next pixel, and again the comparison is made. After all the pixels are tried, the area with the least difference is the optional region.

As Fig. 1 shows, template $T(X \times Y$ pixels) is to be moved horizontally on the original image S. The area that the template covers on S is called sub image. (i, j) is the coordinate of the up-left corner of sub image on S.

The difference between the sub image and the template is measured by the sum of square errors. The size of the template is supposed to be $X \times Y$ (width times height), and the size of the original image is $M \times N$. The coordinate of a point in the template is (x, y). The grey value of this point is $T(x, y)$. The coordinate of the point that matches (x, y) is $(i\text{-}x, j\text{-}y)$, and its grey value is $S(i\text{-}x, j\text{-}y)$. Here it is denoted by $S^{i,j}(x, y)$. Then the sum of square errors of one matching is

$$D(i,j) = \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[S^{ij}(x,y) - T(x,y)]^2 \tag{1}$$

And it is expanded as follows:

$$D(i,j) = \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[S^{ij}(x,y)]^2 - 2\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[S^{ij}(x,y)\times T(x,y)] + \sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[T(x,y)]^2 \tag{2}$$

In the equality above, the first item on the right is called the energy of the template. It relates to the location of sub image, and changes slightly as the sub image moves. The second item is called the correlation between the template and the sub image. It changes with the location of the sub image (i, j), and reaches its maximum when $T(x, y)$ matches the corresponding region. The third item is the energy of the template. It is irrelevant to (i, j), and only needs to be calculated once.

When T and $S^{i,j}$ match, the second item achieves its maximum. So this one item can decide image matching. The following correlation function can be set to measure the similarity.

Given the item DS is a constant, which may give rise to inaccuracy and even worse may greatly affect the matching, it is considered in the normalization processing. Then a normalized cross correlation function is obtained below:

$$R(i,j) = \frac{\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[S^{ij}(x,y)\times T(x,y)]}{\sqrt{\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[S^{ij}(x,y)]^2}\sqrt{\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}[T(x,y)]^2}} \tag{3}$$

According to this equality, every $R(i,j)$ has a value. As (i,j) changes, the maximum of $R(i, j)$ indicates the best position for T, and thus the matching image is got. As can be seen, image matching involves a large amount of computation. One matching requires $X\times Y$ times of subtractions, $X\times Y$ times of squares, $X\times Y$-1 times of addition, and the whole image needs to be matched $(M\text{-}X+1) \times (N\text{-}Y+1)$ times.

Accordingly, in normalized cross correlation algorithm, relative calculation shall be done on $(M\text{-}X+1) \times (N\text{-}Y+1)$ points. Besides on the best matching point, the rest computation is worthless.

III. Particle Swarm Optimization Algorithm

PSO was presented by Kennedy and Eberhart [5] in 1995. The algorithm simulates flight behavior of flocks of birds. Birds aim to achieve optimization results through a collective collaboration between the groups. In the PSO system, each candidate solution is called a "particle". A number of particles coexist and cooperate to find optimization. Each particle "flies" to a better position in problem space in accordance with its own "experience" and the best "experience" of the adjacent particle swarm, searching the optimal solution.

Mathematical notation of PSO algorithm is defined as follows: Assume searching space is D-dimensional and the total number of particles is n. The ith particle location is denoted by the vector: $X_i=(x_{i1}, x_{i2},\ldots, x_{iD})$; The past optimal location of the ith particle in the "flight" history (that is, the location corresponds optimal solution) is $P_i=(p_{i1}, p_{i2},\ldots, p_{iD})$. The past optimal location P_g of the gth particle is optimal in all of $P_i(i=1,2,\ldots,n)$; The location changing rate (speed) of the ith particle is denoted by the vector $V_i=(v_{i1},v_{i2},\ldots,v_{iD})$. The location of each particle changes by the following formula:

$$v_{id}(t+1)=x_{id}(t)+c_1 rand()(p_{id}(t)-x_{id}(t))+c_2 rand()(p_{gd}(t)-x_{id}(t)), \tag{4}$$

$$X_{id}(t+1)=x_{id}(t)+v_{id}(t+1),\ (1\le i\le n, 1\le d\le D), \tag{5}$$

where c_1, c_2 are positive constants called accelerating factor; $rand()$ is a random number between 0 and 1; w is called inertia factor; w, set a litter greater, is suited to a wide range of exploration to solution space while smaller is suited to a small range. $[X_{\min}, X_{\max}]$ is the changing range of particle location. $[v_{\min}, v_{\max}]$ is the changing range of speed. If the location and speed exceed boundary range in iteration, given boundary value. Based on analysis of the above parameters, Maurice Clerc provides the parameter conditioning of PSO algorithm convergence [9].

IV. Image Matching Algorithm Based on PSO

A. Pretreatment of the Image

In template matching, at first the information of the image needs to be reserved in the memory. A grey image of the size of $M\times N$ in the memory can be described as a matrix:

$$T = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}_{m\times n}$$

$x_{11}, x_{12},\ldots, x_{mn}$ stand for grey value respectively.

As Fig. 1 indicates, the matching image can be found only through matching the up-left point of the template with each point in the original image. The fitness function of particle swarm can be expressed in equality (3).

B. Movement of the particles

Similar to other evolutional algorithms, particle swarm algorithm also evolves according to the adaptability of the individual (particle). Each individual is seen as a particle with no weight and volume flying at a certain speed in the N-dimensioned searching space. The particles start to move and iterate toward the best matching position till they meet the objective. The formula for particle movement is as equality (5). And the figure below better and more visually presents the movement of particles.

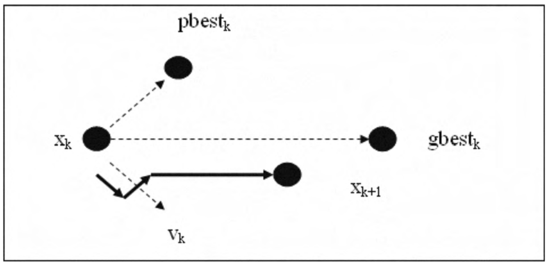

Figure 2. Movement of the particles

In Fig. 2, particles make movements according to the experience of both the group and themselves, which is the essence of the particle swarm optimization algorithm.

C. *Algorithm implementation*

In this paper, we use PSO to solve ITMP, whose pseudocode of algorithm is described in Fig. 3:

Set iteration times for *MaxN*, randomly generated initial particles *N*;
Calculate the fitness of initial particle to be l_0. According to the initial fitness of each particle, initialize $plbest_i$, $pxbest_i$, *glbest* and *gxbest*;
WHILE (iteration times< *MaxN*) DO
  FOR i=1: *MaxN*
    Calculate the ith particle X_i to be $X_i^{'}$ by Eq.5;
    Calculate fitness l_i according to current location by Eq.3;
    IF (l_i< $plbest_i$)
      $pxbest_i$=X_i, $plbest_i$=l_i;
    END IF
  END FOR
  Update $plbest_i$, $pxbest_i$, *glbest*, *gxbest*;
END WHILE
Finally, export *glbest* and *gxbest*.

Figure 3. Pseudocode of PSO algorithm to solve ITMP

V. CALCULATION EXAMPLE

The experiment is made with a 512×512 original image, as Fig. 4 shows. Fig. 5 is the matching template.

Figure.4. Source image

Figure.5. Template

We look for the sub image in Fig. 4 that matches the template Fig. 5 by PSO algorithm. The parameter setting in the algorithm is as follows: w = 1.0, $c1$=$c2$=1.8, and the particle dimension is 2. The algorithm is realized via VC++ programming, and the hardware environment is Inter Core 6300 1.86GHz PC. In order to make clear the properties of this algorithm, this paper analyses the results of the experiment in terms of population size and iteration number, as presented in Chart I below.

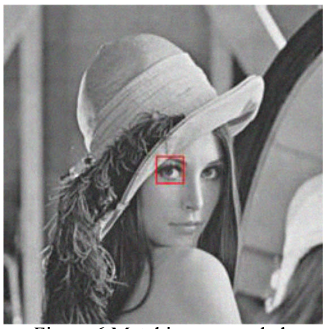

Figure.6 Matching succeeded

Figure.7 Matching failed

TABLE I. COMPARED RESULTS IN DIFFERENT NUMBER OF PARTICLES FOR SOLVING 50 TIMES

test conditions: test 50 times, iteration number is 300								
particles	5	10	20	30	40	50	60	70
correct	18	32	38	42	47	48	49	50
Rate	36	64	76	84	94	96	98	100
Average time /ms	31	63	125	218	265	328	406	484
test conditions: test 50 times, iteration number is 600								
particles	5	10	20	30	40	50	60	70
correct	20	35	40	47	49	50	50	50
Rate	40	70	80	94	98	100	100	100
average time /ms	60	125	255	344	516	657	781	938
The time for integrity exhaustive algorithm is 5313 ms.								

Experimental results show that the searching successful rate and the searching time of the proposed PSO algorithm in solving VRPTW problem are preferable. When the number of iteration was 300, n=70, the searching successful rate reached 100%, and the searching time was 484 ms; when the number of iteration was 600, n=50, the searching successful fate reached 100% as well, with the searching time 657 ms. The searching time of PSO above also is far less than that of exhaustive matching algorithm 5313 ms.

VI. CONCLUSION AND FUTURE WORK

This paper puts PSO algorithm into use in image template matching problems and achieves satisfying effects. The analysis of experiment results shows PSO algorithm possesses the following advantages: First, this algorithm has good properties with high efficiency. Second, the parameters of the algorithm can be adjusted to balance between searching time and searching accuracy, so as to better meet the requirements of actual problems and practices.

Undoubtedly, the speedy development of the society nowadays calls for efficient image matching algorithm to match fast and correctly. Accordingly, how to have an improved PSO algorithm in solving image template matching problems is the key of our research.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their valuable comments and suggestions that helped us for improving our work. This work is funded by the Natural Science Foundation of Guangdong, China under Grant 7300090.

REFERENCES

[1] Sun Ji-xiang. “Digital Image Processing”. Shijiazhuang: Hebei Education Press, 1993.

[2] Shoji Muramatsu, Yuji Otsuka, Yoshiki Kobayashi, et al. “Strategy of High-Speed Template Matching and Its Optimization by Using GA”. Systems and Computers in Japan, 2003, Vol.34, No.2, pp. 81-91.

[3] Gao Ying-hui, Lu Kai, Shen Zhen-kang. “Template matching based on angle-coding chromosome quantum genetic algorithm”. Application Research of Computer, 2008, Vol.25, No.11, pp. 3509-3513.

[4] Ying Hai-yan, Wang You-bo. “Self-Adaptive Genetic Algorithm Based Template Matching Study”. Microelectronics & Computer, 2008, Vol. 25, No.1, pp. 163-165.

[5] R C Eberhart, J Kennedy. “A New Optimizer Using Particles Swarm Theory”. In: Proc Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 1995.

[6] Jiang Wei-gang, Zhang Yuan-biao, Xie Jian-wen. “A Particle Swarm Optimization Algorithm Based on Diffusion-Repulsion and Application to Portfolio Selection”. 2008 International Symposium on Information Science and Engineering, 2008, vol. 2: pp. 498-501

[7] WU Yong, YE Chun-ming, MA Hui-min, et al. “Parallel particle swarm optimization algorithm for vehicle routing problems with time windows”. Computer Engineering and Applications, 2007, Vol.43, No.14, pp. 223-226.