

A NOVEL HOUGH TRANSFORM BASED ON ELIMINATING PARTICLE SWARM OPTIMIZATION AND ITS APPLICATIONS

Yanhui Guo^{1,2} H.D. Cheng^{1,2} , , Wei Zhao¹ , Yingtao Zhang¹

¹School of Computer Science and technology, Harbin Institute of Technology, Harbin, China, 150001

²Department of Computer Science, Utah State University, Logan, UT 84322 U.S.A.

Abstract

Hough transform (HT) is a well established method for curve detection and recognition due to its robustness and insensitiveness to noise, and its parallel processing capability. However, HT is quite time-consuming. In this paper, an eliminating particle swarm optimization (EPSO) algorithm is studied to improve the speed of a Hough transform. The solutions of Hough transformation are considered as the particles positions, and the EPSO algorithm searches the optimum solution by eliminating the “weakest” particles to speed up the computation. An accumulation array in Hough transformation is utilized as a fitness function of the EPSO algorithm. The experiments on numerous images show that the proposed approach can be used to detect curves or contours of both noise-free and noisy images with much better performance. Especially, for noisy images, it can archive much better results than that obtained by using the existing HT algorithms.

Keywords: Hough transform, Particle swarm optimization, Eliminating PSO, Curve detection.

1. Introduction

Hough transform (HT) is an important and popular curve detection method which could be employed to recognize and detect analytically defined curves, including lines, circles, etc. [1, 2], and the generalized Hough transform can be used to detect arbitrary curves under certain conditions [3-5].

Hough transform is very robust in the presence of additional structures as well as being insensitive to noise, suitable to parallel processing and could search several curves in one process. It has been applied widely to image processing, pattern recognition and image vision, motion detection [6, 7], temporal signal monitoring [8], chirp detection [9], character recognition [10], defect detection [11], etc.

HT consists of three main steps. First, a pixel is transformed into a parameterized curve. Second, valid curve's parameters are binned into an accumulator where the numbers in a bin are defined as the curves' score. Finally, curves with the scores higher than the threshold are detected [12].

The conventional HT treats all angles equally, which results in heavy computation, huge parameter space, and less-salient peaks. Various literatures were proposed to modify the conventional HT. They mainly focus on how to select an-

gles to conduct the transform accurately and efficiently.

HT is insensitive to missing parts of lines, to noise, and to other non-line structure co-existing in the image, and it may search for several curves in one pass of the process, however, HT suffers several flaws [13]:

- (1) Huge computation and memory storage
- (2) Low speed
- (3) Depending on the number of parameters and the split of the parameter space
- (4) Difficulties in finding local maxima if peak is not properly defined.

In order to reduce the computational time and improve the performance on noisy images, in this paper, an eliminating particle swarm optimization Hough transformation (EPSOHT) algorithm is studied. In this method, the solutions of Hough transformation are considered as the particles positions, and EPSO algorithm searches the optimum solution by eliminating the “weakest” particles to speed up the computation. An accumulation array in Hough transformation is utilized as a fitness function of the EPSO algorithm. The experiments on numerous images demonstrate that the proposed approach can detect curves with better accuracy and higher speed, especially, in noisy cases.

2. Proposed approach

2.1. Particle swarm optimization algorithm

Particle swarm optimization (PSO) algorithm is an evolutionary computation technique utilizing random search inspired by the mechanics of natural selection and genetics to emulate the evolutionary behaviors of biological systems. The PSO was introduced in [14], which simulated simplified swarm social models such as bird flocking and fish schooling.

PSO has a fitness function to compute each position’s fitness value [14]. The position with the highest fitness value in the entire run is called the global best solution P_{Best} . Each particle also tracks its highest fitness value. The location of this value is called the personal best solution P_i . The algorithm involves: casting a population of particles over the search space and remembering the best solution encountered. At each iteration, every particle adjusts its velocity vector based on its momentum and the influence of both its best solution and the global best solutions of its neighbors, then a new point is examined. The studies show that the PSO has more chances to “fly” into better solution areas quickly; hence, it can discover a reasonable solution much faster than other evolutionary algorithms. The detail of PSO can be found in [14, 15]:

Assume P_i represents the i th particle, whose position and velocity in a d -dimensional space are defined as X_{id} and V_{id} , respectively. The position and velocity are updated according to the following formulas:

$$V_{id}(t) = \omega V_{id}(t-1) + c_1 rand()(P_{id}(t-1) - X_{id}(t-1)) + c_2 rand()(P_{ig}(t-1) - X_{id}(t-1)) \quad (1)$$

$$X_{id}(t) = X_{id}(t-1) + V_{id}(t-1) \quad (2)$$

where $X_{id}(t)$ is the position of the i th particle in a d -dimensional space at time step t , and V_i is the velocity of $P_i(t)$. Parameters c_1 and c_2 are learning factors, usually, $c_1 = c_2 = 2$, ω is an inertia weight and $rand()$ is a random function.

2.2. Eliminating PSO algorithm

A modified PSO, eliminating PSO (EPSO), is based on the idea of survival of the superior and weeding out the inferior. M particles are initialized. The velocities and positions are updated accord-

ingly, and the positions' fitness values are calculated and sorted in a list with the descending order. Then L particles are eliminated whose fitness values are in the last L positions of the list. This will reduce the computational time, while the precision of the solution is not affected. The process is iterated until the maximum iteration number is reached or the minimum error condition is satisfied.

The procedure of EPSO is described below:

- (1) Select M particles (primary population number) and put them into the primary swarm $S(1) = \{P_1, P_1, \dots, P_M\}$, and initialize the positions X_{id} of swarm S randomly;
- (2) Randomly initialize the velocities V_{id} ;
- (3) Evaluate the fitness of each particle $Fit(X_{id}(t))$;
- (4) Compare the personal best of each particle in the new swarm $S(t+1)$ with its current fitness value, and set $P_{id}(t)$ to the better one:

$$P_{id}(t+1) = \begin{cases} P_{id}(t) & Fit(P_{id}(t)) > Fit(X_{id}(t)) \\ X_{id}(t) & Fit(P_{id}(t)) \leq Fit(X_{id}(t)) \end{cases}$$
- (5) Set the global best $P_{gd}(t+1)$ to the position of the particle with the best fitness in the swarm;
- (6) Sort the particles according to the fitness values. A new swarm $S(t+1)$ is obtained by eliminating the L particles whose fitness values are in the last L positions of the list;
- (7) Change the velocity vector $V_{id}(t+1)$ for each particle according to Eq. (1);
- (8) Update each particle position in $S(t+1)$;
- (9) Go to step (3), and repeat the process until the maximum iteration number is reached.

2.3. EPSO Hough transformation

EPSO Hough transformation (EPSOHT) is a novel Hough transformation based on EPSO algorithm. In this method, the parameters of a solution after Hough transformation are considered as the particles positions, and EPSO algorithm is employed to search the optimum solution by eliminating the "weakest" particles to speed up the computation. An accumulation array in Hough transformation is utilized as a fitness function of EPSO algorithm.

Hough transformation is usually employed for circle detection. In circle detection, three parameters, x_0 (the x-coordinate of the center of the circle), y_0 (the y-coordinate of the center of the circle), and r (the radius of the circle) should be determined. Here, Let us use circle detection as an example to demonstrate the procedure of EPSOHT.

The procedure of circle detection using EPSOHT is described below:

- (1) Select M particles (primary population number) and put them into the primary swarm $S(1) = \{P_1, P_1, \dots, P_M\}$, determine the parameters in Hough transformation, x_0 (the x-coordinate of the center of the circle), y_0 (the y-coordinate of the center of the circle), and r (the radius of the circle);
- (2) Construct an edge set E ;
- (3) Select three points P_1, P_2 and P_3 in E randomly, and calculate a circle's parameters determined by these three points, finally, initialize a particle's position using the circle's parameters X_{id} (positions of swarm S);
- (4) Initialize all particles in the primary swarm using the method in step (3);
- (5) Randomly initialize the velocities V_{id} ;
- (6) Evaluate the fitness of each particle $Fit(X_{id}(t; x_0, y_0, r))$:

$$Fit(X_{id}(t; x_0, y_0, r)) = Accumulator(x_0, y_0, r) = \#(\delta(x_0, y_0, r))$$

$$\delta(x_0, y_0, r_0) = \begin{cases} 1 & x = x_0 \text{ and } y = y_0 \\ & \text{and } r = r_0 \\ 0 & \text{otherwise} \end{cases}$$

(7) Compare the personal best of each particle in the new swarm $S(t+1)$ with its current fitness value, and set $P_{id}(t)$ to the better one.

$$P_{id}(t+1) = \begin{cases} P_{id}(t) & \text{Fit}(P_{id}(t)) > \text{Fit}(X_{id}(t)) \\ X_{id}(t) & \text{Fit}(P_{id}(t)) \leq \text{Fit}(X_{id}(t)) \end{cases}$$

(8) Set the global best $P_{gd}(t+1)$ to the position of the particle with the best fitness in the swarm;

(9) Sort the particles according to the fitness values. A new swarm $S(t+1)$ is obtained by eliminating the L particles whose fitness values are in the last L positions of the list;

(10) Change the velocity vector $V_{id}(t+1)$ for each particle according to Eq. (1);

(11) Update each particle position in $S(t+1)$;

(12) Go to step (5), and repeat the process until the maximum iteration number is reached or the best fitness value is greater than a threshold value.

3. Experimental results

3.1. Experiments on NOISE FREE images

In this section, EPSOHT is compared with the random Hough Transformation (RHT) [13]. In our experiments, primary particles number M is 100 and eliminating particle number L is 10.

Eight circles with different radii are shown in Fig. 1. The comparison of computational time of EPSOHT and RHT is shown in Fig. 2. In Fig. 2, the computational times of the EPSOHT and RHT are described by black and gray bars, respectively. As shown in Fig. 2, the proposed method could reduce the computational

time greatly compared with SHT, and achieves the same speed as RHT for the “clean” images.

In order to evaluate the accuracy of the circle detection results, a metric, error sum Es , is defined to measure the errors and differences between the detection results and true results.

$$Es = abs(x_{det} - x_{true}) + abs(y_{det} - y_{true}) + abs(r_{det} - r_{true}) \quad (3)$$

where x_{true} , y_{true} and r_{true} are the true coordinates of circle’s center and radius, while x_{det} , y_{det} and r_{det} are the detected coordinates of circle’s center and radius, respectively.

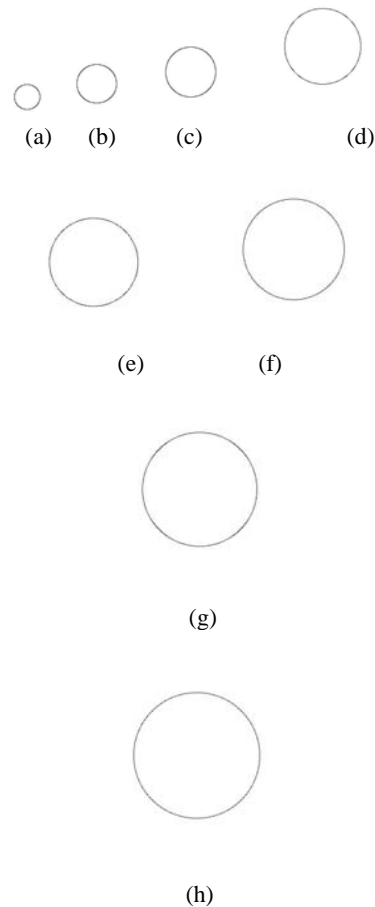


Fig. 1 Eight images having circles with different sizes.

(a) 100X100, (b) 150X150, (c) 200X200, (d) 300X300, (e) 350X350, (f) 400X400, (g) 450X450, (h) 500X500.

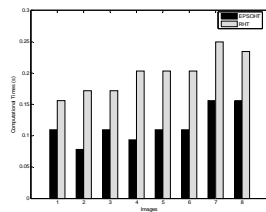


Fig. 2 The computational times of EPSOHT and RHT.

The average of EPSOHT method's error sum is 2.9 and the average of computational time is 0.1152 second.

The EPSOHT has the error sum nearly the same as RHT, but reduces the computational time by 42.53% of RHT's computational time in average, especially, for large size circles. The algorithm of EPSOHT is implemented using Matlab 7.1 and the program was executed on a PC with a single processing unit AMD XP2500 and 256MB random access memory. The average execution time was 0.1152 second per image, whose average size is 275X275, while the average execution time using RHT was 0.1992 second per image, i.e., EPSOHT is twice faster.

3.2. Experiments on images with different noise levels

An image, whose size is 256X256, has a circle in the center, added different kinds of noise (Gaussian, Salt and pepper, Speckle) with different noise levels. Then, the noisy images are processed by EPSOHT and RHT, and the detection results and computational times are compared.

The images having Gaussian noise at different levels are shown in Fig. 3, and Fig. 4 shows the computational time of EPSOHT and RHT described by black and gray bars, respectively.

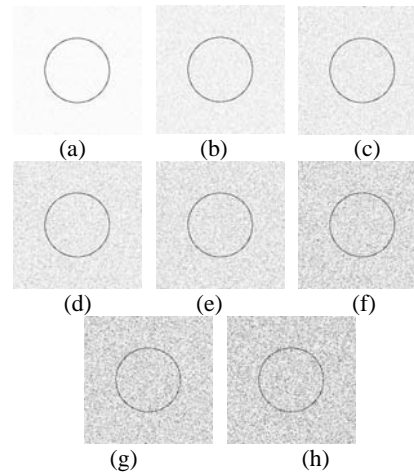


Fig. 3 Images with Gaussian noise, whose mean $m=0$ and standard deviations v having different values: (a) $v = 10$, (b) $v = 30$, (c) $v = 40$, (d) $v = 50$, (e) $v = 60$, (f) $v = 80$, (g) $v = 90$, (h) $v = 100$.

From the experimental results, the proposed EPSOHT achieves better performance in detection speed and detection accuracy.

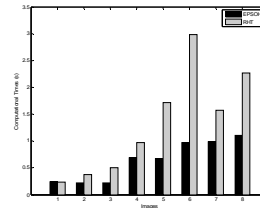


Fig. 4 The computational times of EPSOHT and RHT for processing the images in Fig. 3.

4. Conclusions

In this paper, the EPSOHT algorithm is developed to improve the processing speed. The parameters of a solution in EPSOHT are considered as the particles positions, and the EPSO is employed to search the optimum solution by eliminating the "weaker" particles to speed up the computation. The experiments on numerous images show that the proposed approach can detect curves of "clean" or noisy images with better accuracy and higher speed. The proposed approach

may find wide applications in image processing, pattern recognition and computer vision, especially, in real-time processing cases.

5. Acknowledgement

The work was supported, in part, by Natural Scientific Research Innovation Foundation in Harbin Institute of Technology, Project HIT.NSRIF.2008.48, and Natural Science Foundation of China No.60873142 and No. 30670546.

6. References

- [1] V. F. Leavers, "Which Hough transform?" *CVGIP: Image Underst.*, vol. 58, no. 2, pp. 250-264, 1993.
- [2] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Comput. Vision Graph. Image Process.*, vol. 44, no. 1, pp. 87-116, 1988.
- [3] Euijin Kim, M. Haseyama, and H. Kitajima, "Fast line extraction from digital images using line segments," *Systems and Computers in Japan*, vol. 34, no. 10, pp. 76-89, 2003.
- [4] T. Achalakul and S. Madarasmi, "A concurrent modified algorithm for Generalized Hough Transform," pp. 965, 2002.
- [5] A. Sakai, Y. Nomura, and Y. Mitsuya, "Matching for affined transformed pictures using Hough planes," MVA'96 IAPR Workshop on Machine Vision Applications, pp. 381-384, 1996.
- [6] H. Kalviainen, " Detecting multiple moving objects by the randomized Hough transform, in time-varying image processing and moving object recognition," Proc. 4th Internat. Workshop on Time-Varying Image Processing and Moving Object Recognition, pp. 375-382., 1993.
- [7] H. Kalviainen, E. Oja, and L. Xu, " Motion detection using randomized Hough transform," Proc. 7th Scandinavian Conf. on Image Analysis, pp. 72-79, 1991.
- [8] A. Imiya, "Detection of piecewise-linear signals by the randomized Hough transform," *Pattern Recognition Letters*, vol. 17, no. 7, pp. 771-776, 1996.
- [9] Y. Sun and P. Willett, "The Hough transform for long chirp detection," Proc. 40th IEEE Conf. on Decision and Control (Cat. No. 01CH37228), pp. 958-963, 2001.
- [10] O. Shiku, H. Takahira, A. Nakamura, and H. Kuroda, "A method for character string extraction from binary images using Hough transform," MVA'96 IAPR Workshop on Machine Vision Applications, pp. 498-501, 1996.
- [11] T. M. Meksen, R. Draï, and F. Sellidj., "Pattern Recognition in Ultrasonic Imagery Using the Hough Transform," World Congress on Ultrasonics, Paris, 2003.
- [12] L. G. Shapiro and G. C. Stockman, *Computer vision*: Prentice Hall Upper Saddle River, NJ, 2001.
- [13] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: randomized Hough transform (RHT)," *Pattern Recognition Letters*, vol. 11, no. 5, pp. 331-338, 1990.
- [14] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE International Conference on*, pp. 1942-1948, 1995.
- [15] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources," *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 81-86, 2001.