*Sensors & Transducers, Vol. 160, Issue 12, December 2013, pp. 653-659*

www.sensorsportal.com

**Sensors & Transducers**

© 2013 by IFSA

*http://www.sensorsportal.com*

# Straight Line Detection Based on Particle Swarm Optimization

**Shengzhou XU, Jun TIE**

College of computer science, South-Central University for Nationalities,
Wuhan, 430074, China
E-mail: whkaociya@gmail.com

*Received: 23 September 2013 /Accepted: 22 November 2013 /Published: 30 December 2013*

**Abstract:** In order to reduce the computational time and improve the performance for straight line detection, a method based on particle swarm optimization (PSO) is proposed in this paper. First, each particle, which represents a straight line, is initialized by randomly selected two edge points from the binary image. Then, the accumulated number of the edge points on the straight line is obtained by calculating the distance between the edge point and the straight line, and as the fitness value of the corresponding particle. At last, if the fitness value of the global best solution is larger than the pre-set threshold, extract the straight line from the position of the best particle, otherwise, the algorithm ends. Comparing with Hough transform and improved randomized Hough transform, the proposed method can effectively reduce the problem of double counting and improve the accuracy and efficiency. *Copyright © 2013 IFSA.*

**Keywords:** Straight line, Detection, Hough transform, Particle swarm optimization, Fitness.

## 1. Introduction

Straight line detection is a fundamental field in computer vision. So far, many methods have been proposed, and Hough transform (HT) is the most important and popular algorithm [1]. In HT, each edge pixel is voted upon a quantized parameter space. Each cell in the accumulator array for the quantized parameter space corresponds to a straight line. The cell with the local maximum of scores is selected, and its parameter coordinates are used to represent a line in the image space. HT is very robust to the presence of additional structures, insensitive to noise, suitable for parallel processing, and could search several lines in one process [2]. It has been applied widely to image processing, pattern recognition [3], character recognition [4], and defect detection [5], etc. However, HT also has the following limitations as Xu et al. pointed out [6]: 1) The accumulator array is practically predefined by windowing and sampling the parameter space in a heuristic way. It usually needs a large array taking up much computing time and storage. 2) For one pixel, not only the correct cell, but also many other cells are accumulated. This brings difficulties in finding the local maxima in the accumulator array. Xu et al. proposed a Randomized Hough Transform (RHT) for detecting curves from a binary image [6]. Each time, RHT randomly chooses $n$ edge pixels in the binary image with equal probability and fits them to a parameterized curve. This method can overcome the above mentioned limitations of HT. However, RHT performs better only on the "clean and simple" images; and its performance degraded greatly on "noisy and complex" images. Furthermore, noise increases the number of points to be processed, which in turn increases the computing time and the number of erroneous detections [7].

Cheng et al. proposed an eliminating particle swarm optimization Hough transform (EPSOHT) [2]. The parameters of the solution after HT are considered as the particle positions, and the EPSO algorithm searches the optimum solution by eliminating the "weakest" particles to speed up the computation. An accumulation array in HT is utilized as a fitness function of the EPSO algorithm.

In order to reduce the computational time and improve the performance for straight line detection, a method based on PSO is proposed in this paper. In this method, each particle of PSO represents a straight line. The number of the edge points on the straight line is counted, and the straight line indicated by the peak parameter is checked. The experimental results show the effectiveness of the algorithm.

## 2. Related Work

### 2.1. Hough Transform

HT is recognized as a powerful tool for graphic element extraction from images due to its global vision and robustness in noisy or degraded environment [8]. The standard HT for straight line is depicted by equation (1).

$$\rho = x\cos\theta + y\sin\theta, \tag{1}$$

where $(x, y)$ and $(\rho, \theta)$ represent a point in the image space and its parameter in the HT parameter space, respectively. All points on the same line in the image space will intersect at one point in the HT parameter space.

Generally, the HT for straight line detection and its variants consist of three basic steps:

1) Feature points in the image space are transformed into a parameterized curve of the parameter space.

2) Accumulate hits for each parameter in the HT space.

3) Detect peaks in the HT parameter space, verify the line indicated by the peak parameter.

Conventional HT requires huge parameter space, heavy computation and less-salient peaks. Randomized Hough transform has been proposed to improve the conventional HT [6].

### 2.2. Randomized Hough Transform

A straight line can be expressed by follows.

$$y = kx + b, \tag{2}$$

where $(x, y)$ and $(k, b)$ represent a point in the image space and its parameter in the HT parameter space, respectively. Two points can determine a straight line, i.e. points $(x_1, y_1)$, $(x_2, y_2)$ can be mapped into a point $(k, b)$ of the parameter space simply by solving the following equations:

$$\begin{cases} y_1 = kx_1 + b \\ y_2 = kx_2 + b \end{cases} \tag{3}$$

Based on this idea, RHT has been proposed [6]. First, two points $(x_i, y_i)$, $(x_{i+1}, y_{i+1})$ are randomly selected from all the bright points of the binary image. Then, a parameter point $p_i(k_i, b_i)$ is given simply by solving equations (3), and put into a parameter data set $P$. Repeat this process, after a certain number of steps, there will be a number of $p_i$ points with the same value accumulated. As a result, by finding out those accumulated points in the set $P$, all the straight lines in the image space can be detected.

In this paper, we propose an improved RHT (IRHT) method. In this method, all the edge points lying on the straight line currently detected are removed from the pixel data set before the detection of the next straight line start. As a result, the storage can be greatly reduced, and it will also improve efficiency.

## 3. Proposed Approach

### 3.1. Particle Swarm Optimization Algorithm

Particle swarm optimization (PSO) algorithm is a stochastic optimization method based on the simulating the movement organisms in societies such as bird flock and fish school [9]. The fundamental hypothesis of PSO suggests that social sharing of information among a group offers an evolutionary advantage. In other words, an individual can profit from the discoveries and previous experience of all other members in that group. Each bird and fish adjusts its position according to the positions of itself and other fellows to produce social movement.

Suppose the swarm is formed by $n$ particles. Each particle consists of two edge points in image space, representing a straight line. Then, the position and velocity of the i[th] particle is denoted by $\mathbf{x}_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2})^T$ and $\mathbf{v}_i = (v_{i1}, v_{i2}, v_{i3}, v_{i4})^T$ respectively, where $i = 1, 2, \ldots, n$, and $x_{i1}$, $y_{i2}$ denote the abscissa and ordinate value of one point, and $x_{i2}$, $y_{i2}$ for the other point. $v_{i1}$, $v_{i2}$, $v_{i3}$, $v_{i4}$ are velocities of the two point of the i[th] particle in the horizontal direction and the vertical direction, respectively.

Each particle "fly" in a 4-dimensional space. PSO has a fitness function to compute each position's fitness value. In this paper, the number of edge points on a straight line denoted by a particle is utilized as the fitness of this particle.

It is important to determine the strategy to adjust positions of particles. The velocity of a particle is changed depending on its flying inertia, current position, the best position itself occurred in the past, and the optimal position of the whole society. The main evaluation process of the swarm can be described as follow.

1) Create $n$ particles to form an initial swarm $P = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. Initialize the position $\mathbf{x}$ and velocity $\mathbf{v}$ of each particle randomly, and initialize the best positions visited so far by the i[th] particle $\mathbf{P}_i$ and the entire society $\mathbf{P}_g$, as follows.

$$\mathbf{P}_i = \mathbf{x}_i, 1 \le i \le n$$
$$\mathbf{P}_g = \arg\max_{1 \le i \le n} f(\mathbf{P}_i)$$

2) Compute fitness value for each particle and update its velocity and position as follows. For $i \in [1, n]$,

$$\mathbf{v}_i = \mathbf{v}_i + c_1 r_1 (\mathbf{P}_i - \mathbf{x}_i) + c_2 r_2 (\mathbf{P}_g - \mathbf{x}_i), \quad (4)$$

$$\mathbf{x}_i = \mathbf{x}_i + \mathbf{v}_i, \quad (5)$$

$$\text{If } f(\mathbf{x}_i) > f(\mathbf{P}_i) \text{ then } \mathbf{P}_i = \mathbf{x}_i, \quad (6)$$

$$\text{If } f(\mathbf{x}_i) > f(\mathbf{P}_g) \text{ then } \mathbf{P}_g = \mathbf{x}_i, \quad (7)$$

where $c_1$ and $c_2$ are the learning factors, $r_1$ and $r_2$ are the random numbers uniformly distributed in the range of [0, 1].

3) Repeat step 2 until termination criteria is met.

Owing to its simplicity, easy implementation and reliable convergence, PSO and its variants have been used in a wide range of optimization problems. In this paper, PSO is used to detect straight line in image, where, $c_1 = c_2 = 1.5$, $w = 0.7$.

## 3.2. Straight Line Detection

As previously analyzed, the conventional HT requires huge parameter space, heavy computation and less-salient peaks. RHT performs better only on the "clean and simple" image. For complex images, the number of points to be processed increases, and in turn increases the computing time and the number of erroneous detections. In order to improve the performance of straight line detection, a method based on PSO is proposed in this paper.

In this method, for point pairs randomly selected from the image space, we do not map them into parameter space like RHT, but use them to construct particles of PSO, which represent straight lines. Then, the accumulated number of the edge points lying on the straight line, which is the fitness value of the corresponding particle, is obtained by calculating the distance between the edge point and the straight line. At last, if the fitness value of the global best solution is larger than the pre-set threshold, extract the straight line from the position of the best particle. Otherwise, the algorithm ends. The algorithm flowchart is shown as Fig. 1.

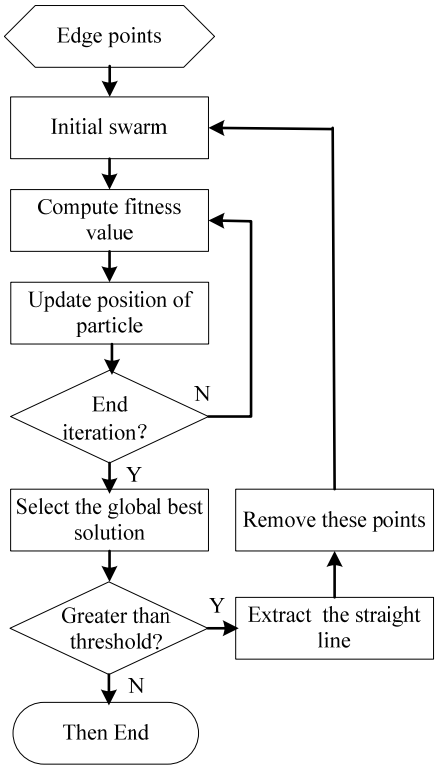


**Fig. 1.** The algorithm flowchart for straight line detection.

The detailed process of the algorithm is as follows:

For each particle $\mathbf{x}_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2})^T$, substituting $(x_{i1}, y_{i1})$ and $(x_{i2}, y_{i2})$ into equation (3):

$$k_i = \frac{y_{i2} - y_{i1}}{x_{i2} - x_{i1}} \quad (8)$$

$$b_i = y_{i1} - \frac{y_{i2} - y_{i1}}{x_{i2} - x_{i1}} x_{i1} \quad (9)$$

Then, the straight line represented by the particle $\mathbf{x}_i$ can be expressed as:

$$y = k_i x + b_i = \frac{y_{i2} - y_{i1}}{x_{i2} - x_{i1}} x + y_{i1} - \frac{y_{i2} - y_{i1}}{x_{i2} - x_{i1}} x_{i1} \quad (10)$$

Let $p(x, y)$ denotes an edge point to be detected. If it is on the straight line represented by (10), the

distance of the point $p$ to the straight line should be less than 1, i.e.

$$d = \frac{|y - k_i x - b_i|}{\sqrt{1 + k_i^2}} < 1 \tag{11}$$

That is:

$$|y - k_i x - b_i| < \sqrt{1 + k_i^2} \tag{12}$$

Due to

$$\sqrt{1 + k_i^2} \geq 1 \tag{13}$$

Inequality (12) can be replaced with the following inequality

$$|y - k_i x - b_i| < 1 \tag{14}$$

Then, the inequality (11) can be calculated simply by substituting (8) and (9) into inequality (14):

$$\left| y - \frac{y_{i2} - y_{i1}}{x_{i2} - x_{i1}} x - y_{i1} + \frac{y_{i2} - y_{i1}}{x_{i2} - x_{i1}} x_{i1} \right| < 1 \tag{15}$$

If the point $p(x, y)$ satisfies inequality (15), it indicates that the point $p$ on the straight line which determined by particle $\mathbf{x}_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2})^T$. Otherwise, the point is not on the straight line. Repeat this process on the other edge points of the binary image, and the accumulated number of the edge points on the straight line can be obtained. This accumulated number is used to be the fitness value of particle $\mathbf{x}_i$.

According to the fitness value, the personal best solution $\mathbf{P}_i$ and the global best solution $\mathbf{P}_g$ can be selected from all the particles. Then, all particles are updated in accordance with the formula (4) and (5). Repeat the above process until the maximum iteration number is reached. At last, the fitness value of the global best solution $\mathbf{P}_g$ is extracted and compared with a pre-set threshold. If the fitness value is larger than the threshold, extract the straight line represented by the position of particle $\mathbf{P}_g$, as the line to be detected. Otherwise no straight line meets the requirement, and the algorithm ends.

## 5. Experimental Results

Figs. 2 to 4 are results of straight line detection for simple image, noisy image and real image by HT, IRHT and PSO respectively. In Fig. 2 and Fig. 3 (both 292×293 pixels), there is no significant difference in appearance between the results of the various methods. In Fig. (384×288), HT and IRHT missed a straight line located in the center of the image, but PSO does not. The efficiency of these methods is shown in Table 1. The "number of actual lines" in Table 1 means the straight line can be distinguished by eyes.

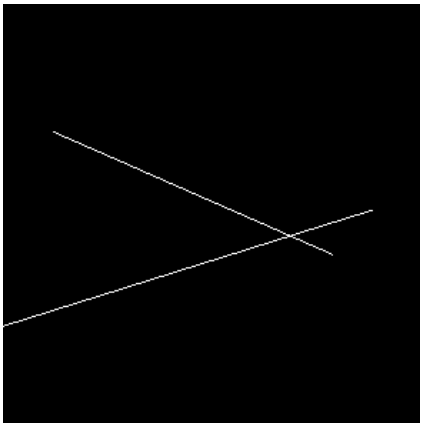

**Fig. 2 (a).** Results of straight line detection for simple image: original image.

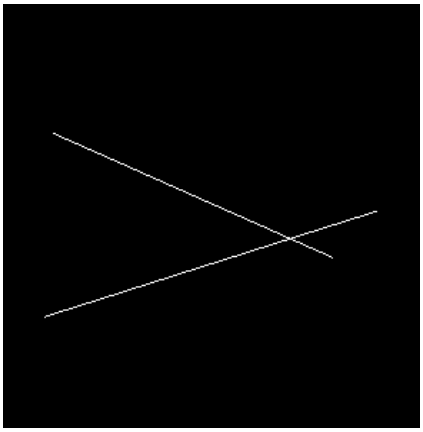

**Fig. 2 (b).** Results of straight line detection for simple image: detection result of HT.

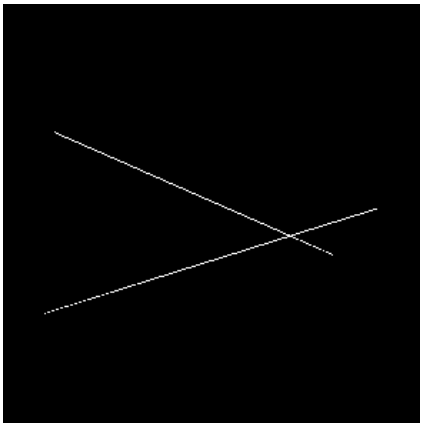

**Fig. 2 (c).** Results of straight line detection for simple image: detection result of RHT.

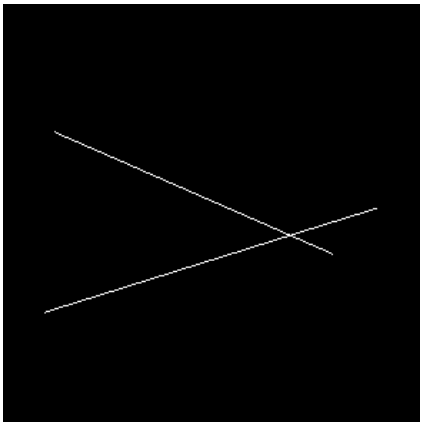

**Fig. 2 (d).** Results of straight line detection for simple image: detection result of PSO.

**Table 1.** Results contrast for different methods.

| Figure | Method | Time(ms) | No. of detected | No. of actual |
|---|---|---|---|---|
| Fig. 2 | HT | 51 | 6 | 2 |
| | RHT | 18 | 2 | 2 |
| | PSO | 15 | 2 | 2 |
| Fig. 3 | HT | 217 | 7 | 2 |
| | RHT | 128 | 2 | 2 |
| | PSO | 61 | 2 | 2 |
| Fig. 4 | HT | 456 | 37 | 9 |
| | RHT | 268 | 9 | 9 |
| | PSO | 215 | 9 | 9 |

In order to verify the effectiveness of the proposed algorithm, we compare the proposed method with the traditional HT and IRHT algorithm on simulated and real images. Experimental tools uses Matlab R2008, and all the experiments are completed on a PC with Intel i3 3.2 GHz, memory 4 G.

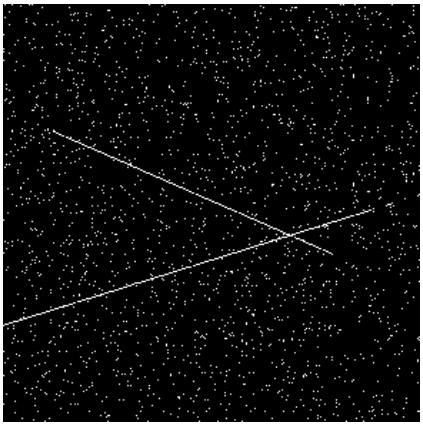

**Fig. 3 (a).** Results of straight line detection for noisy image: original image.

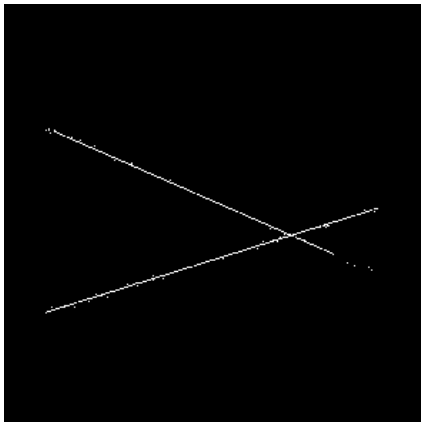

**Fig. 3 (b).** Results of straight line detection for noisy image: detection result of HT.

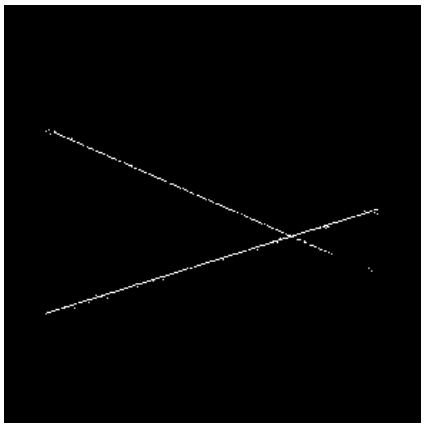

**Fig. 3 (c).** Results of straight line detection for noisy image: result of RHT.

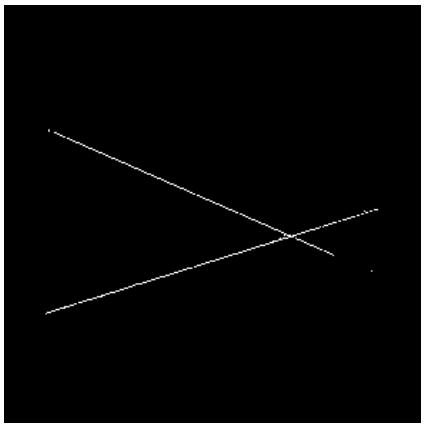

**Fig. 3 (d).** Results of straight line detection for noisy image: detection result of PSO.

It can be summarized from Table 1 as follows. For HT, there are a lot of double-counting which reduce the computational efficiency and effectiveness, and it is difficult to determine which line is double counting. While for IRHT and PSO, we remove all the edge points on the straight line currently detected from the pixel data set $E$ before

the detection of the next straight line. It helps to avoid double counting, and reduce the computational time. For complex image, the number of points to be processed by IRHT increases, and in turn increases the computing time and the number of erroneous detections. Therefore, IRHT performs well only on the "clean and simple" image, but PSO performed well in various circumstances.

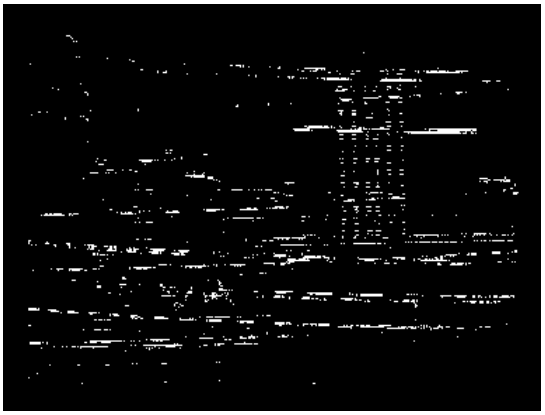

**Fig. 4 (a).** Results of straight line detection for real image: original image.

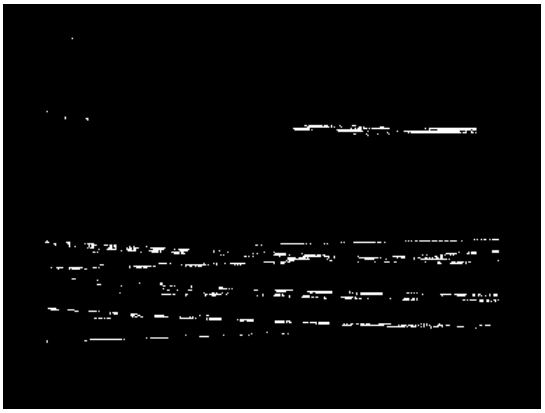

**Fig. 4 (b).** Results of straight line detection for real image: detection result of HT.

**Fig. 4 (c).** Results of straight line detection for real image: detection result of RHT.

**Fig. 4 (d).** Results of straight line detection for real image: detection result of PSO.

## 6. Conclusions

In order to reduce the computational time and improve the performance for straight line detection, a method based on PSO is proposed in this paper. The straight lines to be detected are considered as the particles positions, and the number of edge points lying on the straight line is used to be the fitness of particle. The experimental results indicate that our method is potentially useful.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NO. 61302192).

## References

[1]. P. V. C. Hough, Method and means for recognizing complex patterns, *US Patent*, 1962.

[2]. H. Cheng, Y. Guo, and Y. Zhang, A novel Hough transform based on eliminating particle swarm optimization and its applications, *Pattern Recognition*, Vol. 42, Issue 9, 2009, pp. 1959-1969.

[3]. S. Shylaja, K. N. B. Murthy, S. N. Nischith, R. Muthuraj, and S. Ajay, Feed forward neural network based eye localization and recognition using Hough transform, *International Journal of Advanced Computer Science and Applications*, Vol. 2, No. 3, 2011, pp. 104-109.

[4]. J. Tian, K. G. Hemantha, and H. Chethan, Skew correction for Chinese character using Hough transform, *International Journal of Computer Applications*, Vol. 22, Issue 2, 2011, pp. 33-36.

[5]. W. C. Li and D. M. Tsai, Defect inspection in low-contrast LCD images using Hough transform-based nonstationary line detection, *IEEE Transactions on Industrial Informatics*, Vol. 7, Issue 1, 2011, pp. 136-147.

[6]. L. Xu, E. Oja, and P. Kultanen, A new curve detection method: randomized Hough transform (RHT), *Pattern Recognition Letters*, Vol. 11, Issue 5, 1990, pp. 331-338.

[7]. E. Duquenoy and A. Taleb-Ahmed, Applying the Hough transform pseudo-linearity property to improve computing speed, *Pattern Recognition Letters*, Vol. 27, Issue 16, 2006, pp. 1893-1904.

[8]. J. Song and M. R. Lyu, A Hough transform based line recognition method utilizing both parameter space and image space, *Pattern Recognition*, Vol. 38, No. 4, 2005, pp. 539-552.

[9]. J. Kennedy and R. C. Eberhart, Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks*, Perth, WA, Vol. 4, 27 Nov 1995-01 Dec 1995, pp. 1942-1948.

---

2013 Copyright ©, International Frequency Sensor Association (IFSA). All rights reserved.
(http://www.sensorsportal.com)