International Journal of Hybrid Information Technology
Vol. 9, No.12 (2016), pp. 117-126
http://dx.doi.org/10.14257/ijhit.2016.9.12.11

High Accuracy Extraction of Sharp Feature Based on Discrete Points

Xuefei Shi

School of Automation and Electrical Engineering, Beijing University of Science & Technology, Beijing 100083, P.R. China
sxf1245@ies.ustb.edu.cn

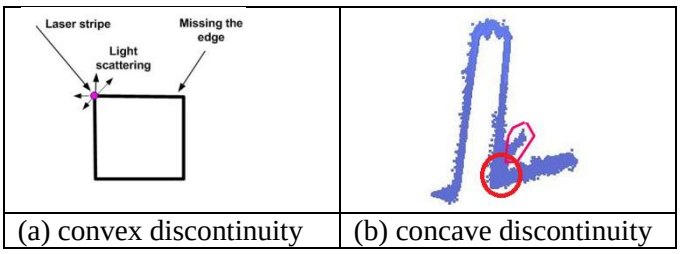
Abstract

With the fast advances of sensing technologies in the past decade, a significant amount of discrete point data has been obtained. Some of these points are of poor quality because of measurement uncertainty at the geometric discontinuity of mechanical parts. In this paper, a particle swarm algorithm is developed for optimally-constrained multiple-line fitting of discrete data points. It contains two important technical components: a) constrained least-squares fitting of multiple lines, b) particle swarm search for optimal corner/edge points. The algorithm is applied to two-dimensional and three-dimensional cases. Numerical experiments indicate the effectiveness and high accuracy of the proposed approach, compared to the conventional method. It can be used for the accurate determination of sharp edges or corners based on discrete data points measured in high-precision inspection and manufacturing.

Keywords: *geometric discontinuity, particle swarm algorithm, sharp feature extraction, discrete data point*

1. Introduction

With the fast development of laser scanning technology, reverse engineering of mechanical parts/components produced a vast amount of discrete data for the digital models of the measured parts/components. These loosely-connected point clouds lack structural information about physical objects. Moreover, a limited number of laser rays likely miss the exact location of corner or edge points. This leads to a significant amount of data errors around the sharp features, as in Figure 1.



(a) convex discontinuity (b) concave discontinuity

Figure 1. Data Abnormality at Geometric Discontinuity

It is therefore a meaningful research how to handle such a kind of data abnormality at geometric discontinuity. A number of studies have been conducted to preserve sharp features during a mesh smoothing process. One technique was anisotropic diffusion that was used in height fields[1], triangle meshes[2-6], level set surfaces[7], surface and function[8], surface reconstruction[9-11]; and the other method was bilateral filtering[12, 13], which was extended to a trilateral filter[14]. Recently, several new methods were proposed for the denoising at geometric discontinuity [15-22]. The aforementioned

ISSN: 1738-9968 IJHIT

methods were designed primarily for entertainment industry where geometric tolerance and small gap between piecewise fitting are not a concern. In essence, smoothing is an iterative process in which the sharp features are inevitably altered to a certain extent, causing unacceptable numerical errors in the field of high-precision manufacturing and industrial inspection. Yet, some other approaches[16] did not change data points at the geometric discontinuity, meaning no data rectification at all.

Particle Swarm Optimization (PSO) is an evolutionary computation technique discovered through the simulation of a simplified social model[23]. It has been successfully applied in many optimization tasks[24-28]. In PSO, each individual *(i.e.,* particle) utilizes two kinds of information in its evolutionary process. The first type is the particles' own experience; the particles have memory and always keep track of their previous best position. The other is intelligence of the whole swarm: each particle adjusts itself towards the global best position[29]. Note that the quality of initial particles can affect the speed of optimization. Normally, the size of initial population and its corresponding positions are randomly generated. In this paper a specially-designed particle swarm algorithm is proposed to solve optimally-constrained line-line fitting. In our approach, the quality of initial particles is improved by estimating the most potential solution space because a corner point always has its specific characteristic.

This paper is organized as follows. Section 2 provides some details about constrained line-line fitting. In Section 3, the proposed particle swarm optimization is applied to search for the optimal corner point. Experimental results are described in the following part, Section 4. Finally, Section 5 provides some conclusions of this study.

2. Constrained Least-squares Fitting of Multiple Lines

The problem of traditional piece-wise regression of measurement data at G^0 discontinuity is the generation of gaps at the location of discontinuity, as illustrated in Figure 2. The gap in this figure is enlarged for the purpose of clarity. In the case of high-precision digital manufacturing, an opening of several hundreds of micrometers would be problematic. To avoid artificial G^0 discontinuity caused by piece-wise numerical fitting, constrained fitting has to be conducted. However, the location of corner points or edge points is not known before the fitting. Thus, the first important step for a constrained fitting is to determine the location of discontinuity.

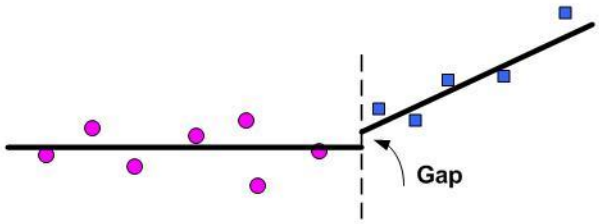


Figure 2. Opening of Two Regression Line Segments at the Location of G^0 Discontinuity

2.1. Two-Dimension Case

In general, the location of corner or edge points should be located in a high-curvature region. The curvature can be computed by a numerical estimate [16] based on principal component analysis. The drawback of principal component analysis is its high computational cost. In the cases of two-dimensional and three-dimensional lines, a simple line propagation scheme can be designed as follows. The basic idea is to start from one end point (point A in Figure 3) and to propagate along one direction of a line. The

propagation continues until the tangent direction is significantly changed (point C in Figure 3). Similarly, a new round of propagation begins with another end point (point B in Figure 3) and terminates at point D. Points C and D define an approximate region for the corner point where two lines are supposed to meet.

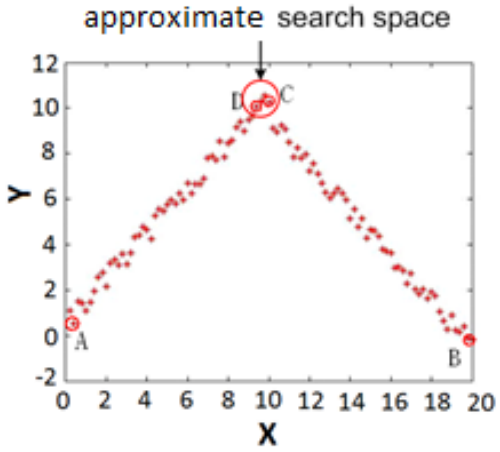


Figure 3. An Approximate Search Space for Corner Location Determined By Line Propagation

In this paper, the exact location of the corner point is searched by a specially-designed particle swarm algorithm. Its details are given in section 3 for both two and three dimensional cases. As a reference of fair comparison, one approach (Scheme 1) is designed to select the best data point as the corner for constrained fitting. This best data point is the point with the minimum fitting error among all the data points within the approximate search space.

In our case, the corner point has to be constrained on the two intersecting lines that meet at this point, as shown in Figure 4. Note that these two intersecting lines are not the same as the lines fitted piecewise by the least squares. The former represents globally-fitted multiple lines, while the latter refers to two locally-fitted lines. The piecewise (*i.e.*, local) fitting means that one line is fitted by the first part of data point on the left, and the other line is produced from the second part of data points on the right, as in Figure 4. In this paper, the slopes of globally-fitted multiple lines are chosen to be the same as the slopes of the locally-fitted lines, respectively. In Figure 4, the slopes of L_1 and L_3 are respectively the same as those of L_2 and L_4.

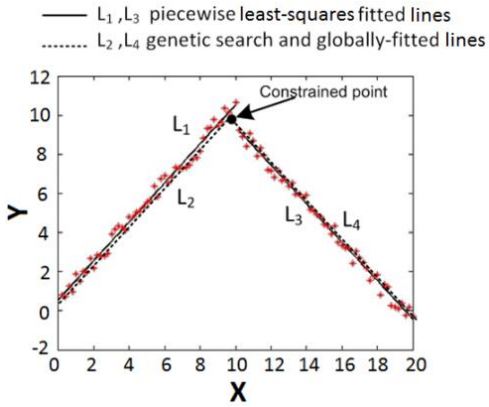


Figure 4.Globally-Fitted Lines and Piece-Wise Fitted Lines.

2.2. Three-dimension Case

We apply a special line-fitting method for three-dimensional cases based on the formulas in [30]. A symmetric form of line equation in three dimensions is given as:

$$\frac{x-x_0}{X}=\frac{y-y_0}{Y}=\frac{z-z_0}{Z}. \tag{1}$$

where x_0, y_0, z_0 represent the coordinates of the point passed by the line. X,Y,Z are direction numbers of the line. Then, we have its parametric equation:

$$\begin{cases} x=\frac{X}{Z}(z-z_0)+x_0=az+b \\ y=\frac{Y}{Z}(z-z_0)+y_0=cz+d \end{cases}, \tag{2}$$

where $a=\frac{X}{Z}, b=x_0-\frac{X}{Z}z_0, c=\frac{Y}{Z}, d=y_0-\frac{Y}{Z}z_0$. A fitting process is therefore converted to the determination of coefficients (a,b,c,d), which minimize two following terms:

$$\begin{cases} Q_x=\sum_{i=1}^{n}[x_i-(az_i+b)]^2 \\ Q_y=\sum_{i=1}^{n}[y_i-(cz_i+d)]^2 \end{cases}. \tag{3}$$

In the above equation, (x_i, y_i, z_i) refers to the i-th discrete data point. The differentiation of the above two terms with respect to a,b,c,d leads to:

$$\frac{\partial Q_x}{\partial a}=0, \frac{\partial Q_x}{\partial b}=0, \frac{\partial Q_y}{\partial c}=0, \frac{\partial Q_y}{\partial d}=0, \tag{4}$$

which are further arranged to

$$\begin{cases} b\cdot n+a\sum_{i=1}^{n}z_i=\sum_{i=1}^{n}x_i \\ b\sum_{i=1}^{n}z_i+a\sum_{i=1}^{n}z_i^2=\sum_{i=1}^{n}x_iz_i \end{cases}, \quad \begin{cases} d\cdot n+c\sum_{i=1}^{n}z_i=\sum_{i=1}^{n}y_i \\ d\sum_{i=1}^{n}z_i+c\sum_{i=1}^{n}z_i^2=\sum_{i=1}^{n}y_iz_i \end{cases}. \tag{5}$$

Equation (5) is then rewritten to a matrix form [30]:

$$\mathbf{FF^TA=FX, FF^TB=FY}, \tag{6}$$

where

$$\mathbf{F}=\begin{bmatrix} z_1 & z_2 & \cdots & z_n \\ 1 & 1 & & 1 \end{bmatrix}, \mathbf{A}=[a \quad b]^{\mathbf{T}}, \mathbf{X}=[x_1 \quad x_2 \cdots x_n]^{\mathbf{T}}, \mathbf{B}=[c \quad d]^{\mathbf{T}}, \quad \mathbf{Y}=[y_1 \quad y_2 \cdots y_n]^{\mathbf{T}}. \tag{7}$$

Since $\mathbf{X, Y, F, F^T}$ are given, the unknown coefficients in A and B can be easily determined by solving the linear equations in Eq. (6).

Any 3D line can then be represented by a parametric equation:

$$\begin{cases} x=at+b \\ y=ct+d \\ z=t \end{cases}, \tag{8}$$

where t is the parameter that has the same range as z. If one line is perpendicular to the z axis, a rotation of related discrete data points about x or y axis needs to be performed before using Eq. (8). After the determination of the corner point, another reverse rotation is conducted.

3. Particle Swarm Optimization for Exact Location of Corner

In our PSO system, each particle is treated as a point in two-dimensional/three-dimensional space. For instance, the k-th position of the i-th particle is represented as $q_i^k = (x_k, y_k)$ for two dimension and $q_i^k = (x_k, y_k, z_k)$ for three dimension. They are updated in the same way by the following equations:

$$v_i^{k+1} = w \times v_i^k + c_1 \times r_1 \times (p_i - q_i^k) + c_2 \times r_2 \times (p_g - q_i^k), \quad (9)$$

$$q^{k+1} = q^k + v_i^{k+1}. \quad (10)$$

Where $i = 1,2,...,N$ and N is the size of initial population (*i.e.*, the number of data points within the approximate corner region). k is the iteration number of particle swarm optimization. w is the inertia weight, and is set to 0.6 in this paper. c_1 and c_2 are two positive constants, and called the cognitive and social parameters, respectively; both of them are 2 in this study. r_1 and r_2 are random numbers that are uniformly distributed within a range [0,0.2]; since our search space is reduced after the high-quality initialization, we provide a smaller range instead of traditional [0,1].

The item $c_1 \times r_1 \times (p_i - q_i^k)$ in Eq. (9) represents that each individual has its memory and tracks its own travelling experience; p_i is the best previous position of the i-th particle. The item $c_2 \times r_2 \times (p_g - q_i^k)$ means that the PSO system shares the same goal of the whole particle set, and each particle adjusts its position towards the best particle among the k-th population; p_g is the position of the globally-best individual. The definition of fitness is discussed in the next section.

Figure 5 shows the description of velocity and position updates of a particle in a two-dimensional search space [29]. Eq. (9) is used to calculate the i-th particle's new velocity v_i^{k+1} at each iteration. (v_i^k is computed in the previous iteration when k is greater than 1, and is generated randomly within range [0, 0.2] when k is equal to 1.) Eq. (10) provides the new position of the i-th particle q_i^{k+1} by adding its new velocity v_i^{k+1} to its current position q_i^k. Note that q_i^1 is determined by the propagation mentioned in section 2.

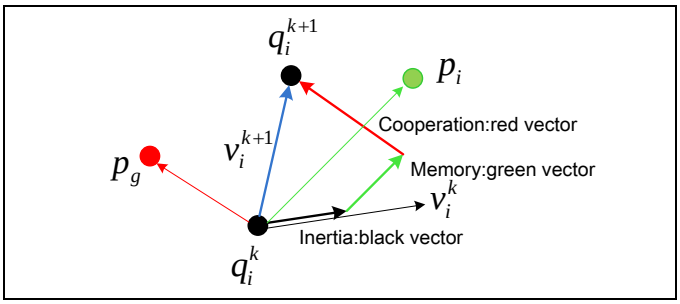


Figure 5. Description of Velocity and Position Updates In A PSO System

3.1. Two-dimension Fitness

For every candidate corner, its fitness can be evaluated by least-squares fitting errors. In our algorithm, the fitting error is determined by the shortest distance between every discrete data point and the fitted line. If the fitting-line equation is $Ax + By + C = 0$, the

error is given by

$$d_i = \frac{\left|Ax_i + By_i + C\right|}{\sqrt{A^2 + B^2}}, \tag{11}$$

where (x_i, y_i) is the i-th discrete data point.

Fitness function, $f(x, y)$, is defined by the sum of those distances:

$$f(x, y) = 1/\sum_{i=1}^{n} d_i . \tag{12}$$

3.2. Three-dimension Fitness

In a way similar to the two-dimensional cases, the sum of the shortest distance between every discrete data point and the fitted line is used to evaluate the fitness of each candidate corner. As in Figure 6, the distance between a data point and a spatial line is:

$$d_i = \frac{\left|\vec{\mathbf{v}} \times \overrightarrow{\mathbf{M_0 M_i}}\right|}{\left|\vec{\mathbf{v}}\right|}, \tag{13}$$

where $\vec{\mathbf{v}}$ is a direction vector of the fitted line, and defined by $(a, c, 1)$ on the basis of Eq. (8). M_i is the i-th discrete data point, and M_0 is a corner candidate point that must be on the fitted line.

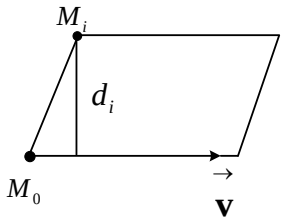


Figure 6. The Distance between a Data Point and a Spatial Line

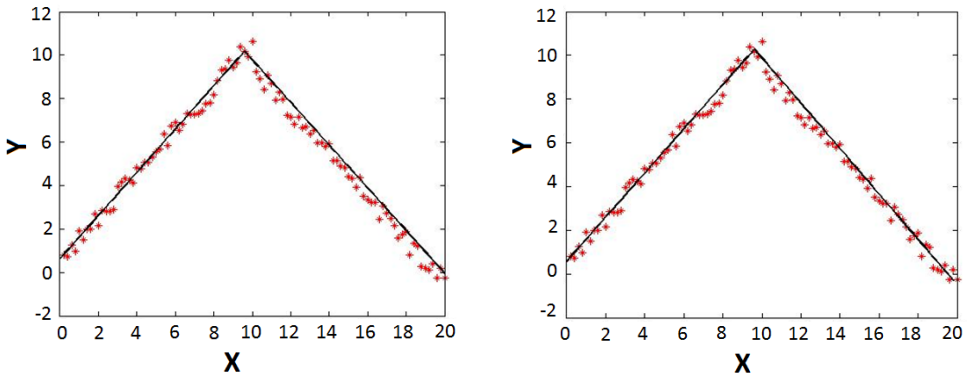
In this paper, a fitness function, $f(x, y, z)$, is defined as:

$$f(x, y, z) = 1/\sum_{i=1}^{n} d_i . \tag{14}$$

4. Experimental Results and Discussion

In this section, some experimental data and figures are computed and presented by using Matlab software. The proposed algorithm was validated in various test cases with different parameters. Noisy discrete data points were generated by using a random number generator on the basis of perturbation from the specified line segments.

First, a comparison was made between the results of Scheme 1 (best data point as constrained corner) and Scheme 2 (particle swarm search for optimal constrained point). Figure 6 demonstrates their different fitting errors with the same set of discrete data points. The error of the former is 24.94, while the error of the latter is 18.96. The coordinate values of their constrained corner are $(9.80, 9.93)$ and $(9.89, 10.28)$ respectively.



(a) best data point as constrained corner (b) particle swarm search for constrained corner

Figure 6. Comparison of Fitting Results between Scheme 1 and Scheme 2

In 3-D cases, the fitting error of the conventional approach (*i.e.*, Scheme 1) was 48.31 and the fitting error of proposed particle swarm optimization is 34.44 as shown in Figure 7.

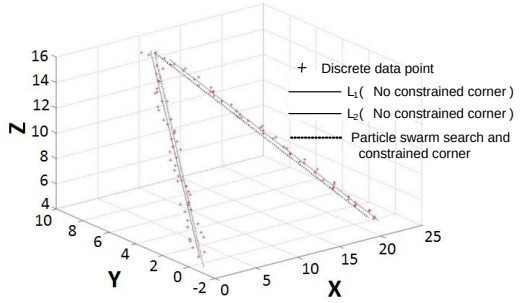


Figure 7. Particle Swarm Search of Constrained Corner

The convergence of particle swarm search for two-dimensional and three-dimensional cases is illustrated in Figures 8 and 9, respectively. In the two-dimension case, there is a clear increase and obvious optimization at the first 10 iterations. After that, there is a tiny improvement until almost no change occurs or even a reduction appears at iteration 20. For position search in three dimensional spaces, there is an obvious improvement during the first 5 iterations and a slight improvement in the next 5 iterations in Figure 9. After that, almost no change occurs. Similar convergence patterns were observed for all the two and three-dimensional cases.

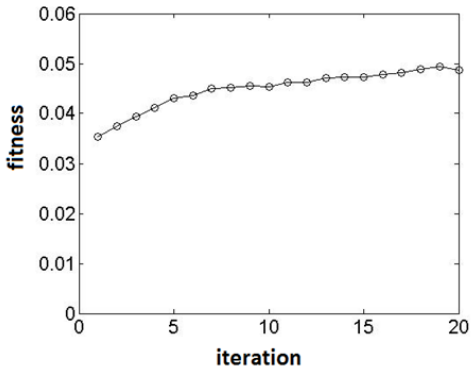


Figure 8. Convergence of Two-Dimensional Case

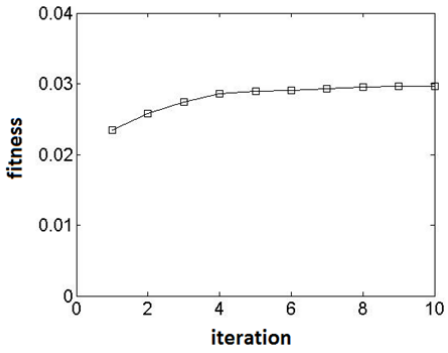
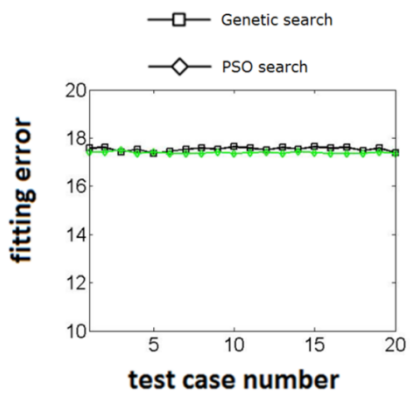
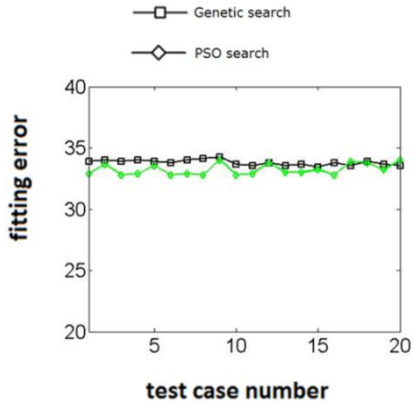


Figure 9. Convergence of In Three-Dimensional Case

Figure 10 illustrates the results of comparison between PSO search and genetic search. In both two-dimensional and three-dimensional cases, the fitting errors are in a very close range. This indicates that both search schemes achieve a similar fitting error in the context of this study. The average fitting errors for PSO and GA are 17.39 and 17.54 in Figure 10(a), while the errors are 33.23 and 33.8 respectively for PSO and GA in Figure 10(b). The proposed PSO search performances less computation time than that of genetic search.



two-dimensional case



three-dimensional case

Figure 10. Comparison between Particle Swarm Search and Genetic Search

5. Conclusions

In this paper, we focus on an accurate data rectification method for discrete points at geometric discontinuity. An improved PSO algorithm for optimally-constrained circle fitting is developed. Two-dimensional and three-dimensional cases are both considered in this study. A simple data line-fitting propagation method is designed for the quality improvement of initialized particles. The optimal corner point is constructed by this improved PSO approach. A comparison between PSO search and genetic algorithm is made with the same discrete data points and same iterations. The experimental results indicate that our improved PSO search method is a better way to locate exactly a constrained point at geometric discontinuity. The following conclusions can be drawn:

(1) The conventional approach (Scheme 1) does not produce an optimal corner point, even though the best data point is chosen within a potential corner space. The proposed PSO algorithm (Scheme 2) can find a much better location of constrained corner with less

fitting error both in two-dimension and three-dimension cases.
(2) The convergence of our proposed PSO (Scheme 2) is reasonable. In this paper, the maximum number of iterations is set to be 20 for two dimensions. For three-dimensional cases, the maximum number of iterations is 10.
(3) Both genetic search and PSO approach generate a similar fitting error in the context of this study. The proposed PSO performances better than genetic in less computation time.

Acknowledgement

The work was supported by a funding (No. 201206465020) from China Scholarship Council, this paper was in part supported by U.S. National Science Foundation DMI-0514900.

References

[1] M. Desbrun, M. Meyer, P. Schroder and A. H. Barr, "Anisotropic feature-preserving denoising of height fields and bivariate data", Graphics Interface,**(2000)**, pp. 145-152.
[2] M. Meyer, M. Desbrun, P. Schroder and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds", In Visualization and Mathematics III, Springer-Verlag, Heidelberg, **(2003)**, pp. 35-57.
[3] U. Clarenz, U. Diewald and M. Rumpf, "Anisotropic geometric diffusion in surface processing", Proceedings of IEEE Visualization, **(2000)**, pp.397-405.
[4] H. Zhang and E. L. Fiume, "Mesh smoothing with shape or feature preservation", In Advanced in modeling, animation, and rendering. (Ed. J. Vince and R. Earnshaw), Springer, **(2002)**, pp. 167-182.
[5] Y. Ohtake, A. Belyaev and I. A. Bogaevski, "Mesh regularization and adaptive smoothing", Computer Aided Design, vol. 33, no. 11, **(2001)**, pp.789-800.
[6] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features", Computer Graphics Forum, vol.23, no. 3, **(2004)**, pp.391-400.
[7] T. Tasdizen, R. T. Whitaker, P. Burchard and S. Osher, "Anisotropic geometric diffusion in surface processing", IEEE Visualization, **(2002)**, pp. 125-132.
[8] C. Bajaj and G. Xu, "Anisotropic diffusion of subdivision surfaces and functions on surfaces", ACM Transactions on Graphics, vol. 22, no. 1, **(2003)**, pp.4-32.
[9] T. Bulow, "Spherical diffusion for 3D surface smoothing", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 12, **(2004)**, pp.1650-1654.
[10] T. Tasdizen and R. T. Whitaker, "High-order nonlinear priors for surface reconstruction", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 7, **(2004)**, pp.878-891.
[11] B. Mederos, L. Velho and L. H. De Figueiredo, "Smoothing surface reconstruction from noisy clouds", Journal of the Brazilian Computer Society, vol. 9, no. 3, **(2004)**, pp.52-66.
[12] T. R. Jones, F. Durant and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing", Proceedings of the 30th Annual Conference on Computer Graphics and Interactive Techniques, **(2003)**, pp.943-949.
[13] S. Fleishman, I. Drori and D. Cohen-Or, "Bilateral mesh denoising", Proceedings of the 30th Annual Conference on Computer Graphics and Interactive Techniques, **(2003)**, pp. 950-953.
[14] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes", Christensen, Per. H. and Cohen, D. Proceedings of the Eurographics Symposium on Rendering, **(2003)**, pp.186-196**.**
[15] B. Li, R. Schnabel, R. Klein, Z. Cheng, G. Dang and S. Jin, "Robust normal estimation for point clouds with sharp features", Computers & Graphics, vol. 34, no. 2, **(2010)**, pp. 94-106.
[16] J. Shen, B. Maxim and K. Akingbehin, "Accurate Correction of Surface Noises of Polygonal Meshes", International Journal for Numerical Methods in Engineering, vol. 64, no. 12, **(2005)**, pp. 1678-1698.
[17] J. Shen, D. Yoon, D. Zhao and Y. Song, "Denoising Two-Dimensional Geometric Discontinuities". International Journal of Computers and Applications 32[2], pp.1-12. **(2010).**
[18] X. Sun, P. Rosin, R. Martin and F. Langbein. "Fast and effective feature-preserving mesh denoising", IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 5, **(2007)**, pp.925-938.
[19] X. Sun, P. Rosin, R. Martin and F. Langbein, "Random walks for feature-preserving mesh denoising", Computer Aided Geometric Design, vol. 25, no. 7, **(2008)**, pp.437-456.
[20] A. C. Oztireli, G. Guennebaud and M. Gross, "Feature preserving point set surfaces based on non-linear kernel regression", Computer Graphics Forum, vol. 28, no. 2, **(2009)**, pp.493-501.
[21] H. Avron, A. Sharf, C. Greif and D. Cohen-Or, "L1-sparse reconstruction of sharp point set surfaces", ACM Transactions on Graphics, vol. 29, no. 5, **(2010)**, pp.135-1-135-20.
[22] O. Schall, A. Belyaev and H. Seidel, "Feature-preserving non-local denoising of static and time-varying range data", Proceedings of the 2007 ACM Symposium on Solid and Physical Modeling**.** New York, NY,

USA, ACM, **(2007)**, pp.217-222.
[23] J. Kennedy, "The behavior of particles", in Lecture Notes in Computer Science. Evolutionary Programming VII: Proceedings of the 7-th Annual Conference on Evolutionary Programming, **(1998)**, pp. 581-589.
[24] K. Sedlaczek and P. Eberhard, "Constrained particle swarm optimization of mechanical systems", in Proceedings of the 6th World Congress of Structural and Multidsciplinary Optimization, **(2005)**, pp. 1-10.
[25] J. J. Liang, A. K. Qin, P. N. Suganthan and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", 10 ed, **(2006)**, pp. 281-295.
[26] Y. K. Baik, J. Kwon, H. S. Lee and K. M. Lee, "Geometric particle swarm optimization for robust visual ego-motion estimation via particle filtering", 31 ed., **(2013)**, pp. 565-579.
[27] X. Yang, J. Yuan, J. Yuan and H. Mao, "A modified particle swarm optimizer with dynamic adaptation", 189 ed., **(2007)**, pp. 1205-1213.
[28] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu and L. M. Wang, "An improved GA and a novel PSO-GA-based hybrid algorithm", 93 ed., **(2005)**, pp. 255-261.
[29] W. F. Abd-El-Wahed, A. A. Mousa and M. A. El-Shorbagy, "Integrating particle swarm optimization with genetic algorithms for solving nonlinear optimization problems", 235 ed., **(2011)**, pp. 1446-1453.
[30] X.Yang, "A method for fitting of a space straight line". Journal of Qiqihar University, vol.25, no.2,**(2009)**, pp.64-68.

Author

Xue-Fei Shi, she received the B.S. degree in Automation Instrument from University of Science & Technology Beijing in 1994 and the M.S. degree in Detection Technology and Automatic Equipment from USTB in 2000. She achieved the Ph.D. degree in Control Science and Engineering from USTB in 2011. She has been a visiting scholar of University of Texas at Arlington for four months in 2006. She is currently a visiting scholar of University of Michigan at Dearborn. Her research interest includes detection and control technology, computer modeling and optimization technology and machine vision.

