Three-Dimensional Surface Recognition Based on Genetic Algorithms

C.Y. Liu and Y.H. Chen

Department of Mechanical Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
Hong Kong, P.R. China
cyliu@hkusua.hku.hk, yhchen@hku.hk

Abstract

A robust method for 3-D surface recognition based on genetic algorithms is proposed in this paper. With the help of a specific least square fitting algorithm, the model-based 3-D surface recognition problem is first converted into an optimization problem. Then, a genetic algorithm is used to solve the optimal problem in order to extract general quadric surfaces. Finally, all the extracted surfaces are classified into specific quadric surface representations by a classification table. Our experimental results show that the proposed surface extraction method can reliably extract 3-D surfaces from both 3-D synthetic and 3-D real measurement data.

Introduction

Three-dimensional (3-D) object recognition is one of the most critical part of computer vision and reverse engineering. From the view-point of boundary-representation (B-rep.), any objects in the real world can be represented by their boundary surfaces. And in most of the cases, objects are described by quadric surfaces. So, the object recognition problem is transformed into quadric surface extraction problem in the case when an object's B-rep. is to be constructed.

Investigations in different aspects of object recognition have been widely reported (Puntambekar, Jablokow, and Sommer 1994; Fitzgibbon, Eggert, and Fisher 1997), especially in recognizing simple objects with plane or natural quadric surfaces such as sphere surfaces. Most of the previous works are mainly concerned about surface extraction from 2D or range image.

From the methodology point of view, these techniques can be classified into two categories (Varady, Martin, and Cox 1997; Chao 1994). The first one is data-driven extraction. These methods are mainly based on the detected features, such as surface curvatures approximated by a small set of local neighborhood points. No strong previous model hypothesis are required for these methods. The second one is model-driven surface extraction. These methods are discussed in more detail.

There are many kinds of model-driven methods, such as robust statistics approaches, connectionist networks and so on. Puntambekar (Puntambekar, Jablokow, and Sommer 1994) discussed a model-driven surface extraction method by fitting surfaces to identified features from deformable initial models and sliced data. This work is effective for reverse engineering, but it relies on boundary detection and topology identification which are difficult problems themselves. Oshima et al. (Oshima and Shirai 1983) proposed a system to recognize stacked objects using range data. The basic idea of Oshima's system is to match the description of the scene to some predefined models one at a time. Oshima's models are much like the conventional templates used for range image processing. Sullivan et al. (Sullivan, Sandford, and Ponce 1994) proposed a method for constructing algebraic surface models from 2-D and 3-D images. These models are used in pose computation and object recognition.

Another kind of model-driven surface extraction methods are based on Hough transform (HT). HT-based methods are very effective for object recognition in 2-D space (Leavers 1993). These methods can not only work for models which can be represented by equations, but can also work for any other kind of templates. Due to the large data storage requirement in 3-D space, HT is not practical for 3-D applications.

In order to overcome the drawbacks of the above methods, some object extraction methods using genetic algorithm (GA) have been investigated recently. The first GA-based method was proposed by Hill and Taylor in 1992 (Hill and Taylor 1992). Their method is mainly aimed at extracting object with a fixed template for 2-D image interpretation.

After that, GA-based geometric primitive extraction methods are further discussed by Roth et al. (Roth and Levine 1993; Roth and Levine 1994). Roth's method is an efficient geometric primitive extraction method. But it requires the explicit expression of each model which is to be extracted. In other words, previous GA-based works were concentrated on extracting a given model from 2-D images or simple range images based on fixed templates or algebraic distance.

In this paper, a general quadric surface extraction method using genetic algorithms is proposed for reverse engineering application. Instead of using fixed templates or algebraic distance, the proposed method uses the more flexible and reliable pseudo geometric distance. Further more, the proposed method is capable of recognizing all quadric surfaces including planes based on a single quadric surface expression.

Motivations and Definitions

It has recently been shown that extracting the best geometric primitive from a given set of geometric data is equivalent to finding the optimum value of a cost function (Roth and Levine 1994). Thus the goal of any extraction algorithm is to find the global optimum from among many local optima. A genetic algorithm is an optimization approach based on the evolutionary metaphor. It has been shown to consistently outperform both gradient methods and random search in solving optimization problems if the cost function is noisy, multidimensional and has multiple local optima (Eiben, Raue, and Ruttkay 1995; Goldberg 1989; Man et al. 1997). So, in this paper, a GA is used to find global optimum of a set of optimal models set up for quadric surface extraction.

Since GAs have rarely been used in reverse engineering, some definitions and important operations of GAs are briefly described in the following.

- Chromosome: the code of a point in the variable space is called a chromosome. It is often a binary code in a conventional GA. In recent research, other encoding methods have been proposed. For example, a solution $\mathbf{x}=(x_1, x_2, \ldots, x_r)$ can be encoded as a string of genes to form a chromosome representing an individual.
- Population: a set of N chromosomes (or say individual in some cases) is called a population. Here N is the size of the current population and it is a predefined constant before evolution.
- Crossover: the operation that two parent chromosomes are combined in some way to generate two new individuals or children.
- Mutation: the operation of selecting a gene randomly for a given individual and mutating the allele for that gene.
- Cost function: the objective function in an optimal research.

At the beginning of a GA, data points in the searching space are encoded to form a set of individuals. Then, an initial population can be created by certain selecting procedures, such as random selection.

After that, the evolution cycles begin. In each evolution cycle, a pair of parents are selected from the current population to generate children by some operators such as crossover and mutation. Though there are many different kinds of crossover and mutation operators, we use the uniform crossover and mutation operators in this research. If the cost of the new child is greater than the minimum cost individual in the current generation, the minimum cost individual is replaced by the new child to keep the generation size invariant. The procedures continue until costs are high enough for all individuals. The individual with maximum cost is the solution of the optimal problem.

Let $\mathbf{x}_i$ be the ith measurement data point, and $\boldsymbol{D}=\{\mathbf{x}_i, i=1,2,\ldots,n\}$ be the whole set of measurement data points. Some of the important definitions used for our algorithm are given below:

- General Quadric Surfaces (GQS): all of the possible first and second degree surfaces in 3-D are called GQS. A specific surface in GQS is denoted as $\boldsymbol{S}$. The equation of GQS is represented by Equation (1).

$$f(\mathbf{x}) = a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5xz + a_6yz + a_7x + a_8y + a_9z + a_{10} = 0 \quad (1)$$

- Algebraic Distance (AD): the absolute value of $f(\mathbf{x})$ for a data point $\mathbf{x}$ is defined as the algebraic distance between a point $\mathbf{x}$ and $\boldsymbol{S}$.
- Pseudo Geometric Distance (PGD): the minimum Euclidean distance between a point $\mathbf{x}$ and $\boldsymbol{S}$ along the x, y and z axes respectively.
- Generalized Minimal Subset (GMS): each subset of 9 data points in $\boldsymbol{D}$ is called a GMS. The ith GMS is denoted as $\boldsymbol{D}_i$.

A GMS is the minimum data points required for the solution of Equation (1). In fact, 10 parameters in Equation (1) must be determined, but there are only 9 of them are free by adding a extra constrain equation.

In our algorithm, a chromosome string is presented by the indices of its member points for effective computation. For example, if a possible GMS is $\boldsymbol{D}_{i0}=\{P_1, P_{12}, P_3, P_{24}, P_{25}, P_6, P_{17}, P_8, P_{39}\}$, the corresponding chromosome string will be represented as {1, 12, 3, 24, 25, 6, 17, 8, 39}. In this paper, we will not distinguish between a GMS and its corresponding chromosome string.

A conventional minimal subset is the smallest number of points necessary to define a unique instance of a geometric primitive , and the primitive passes through the points in the minimal subset exactly without error of fit (Roth and Levine 1994). A GMS is different from the conventional minimal subset in that a GMS may not consist any part of any quadric surfaces, because some GMSs can not be fitted successfully by the least-square fitting method.

- Cost Function of a GMS (CFG): the function that counts the number of points within a small PGD to the quadric surface fitted by the current GMS.

The CFG of a GMS $\boldsymbol{D}_i$ is denoted as $h(\boldsymbol{D}_i)$. For example, if the current chromosome is $\boldsymbol{D}_{i0}=\{2, 4, 19, 1, 33, 21, 23, 8, 9\}$ and there are 14 points within a small PGD to the current fitted surface of $\boldsymbol{D}_{i0}$, the value of cost function (abbreviated as cost in the later parts of this paper) will be 14, that is, $h(\boldsymbol{D}_{i0}) = 14$.

- Minimum cost of a population (MCP): the minimum value of CFG in a population is called the MCP of this population.

Algorithm

Our GA-based quadric surface extracting algorithm is composed of two main components. The first one is quadric surface parameter finding by least square fitting (Marshall and Martin 1992) for all GMS. From the view point of feature extraction, this component can be taken as data-driven. For the second one, a modified GA working on GMS is used to extract the surface that has the maximum CFG. This component relies mainly on the fitted surface parameters, so it can be taken as model-driven.

Quadric Surface Parameter Calculation by Lagrangian Multipliers

In our research, the least square fitting algorithm for quadric surfaces is based on Lagrangian multipliers and matrix eigenvalue decomposition (Marshall and Martin 1992; Pettofrezzo and Lacatena 1970). And because it is a linear problem, no iterative method is required.

A general quadric surface shown in Equation (1) can be expressed alternatively,

$$\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{x}^T\mathbf{v} + d = 0 \qquad (2)$$

Matrix **A** gives information about the geometric form of the quadric surface.

The natural geometric interpretation of error in least squares fitting is the distance of each point to the surface. However, in the case of quadric surfaces, using such a measure would cause the mathematics of the approximation to become very awkward. Instead, Martshall et al. (Marshall and Martin 1992) suggested to use quantities similar to those used in the case of planes to keep the approximation simple. Thus the error in the fit of the set of points $\{\mathbf{x}_i\}$, $i=1, \ldots, n$, to a quadric surface is defined by using Equation (2) as

$$\chi^2 = \sum\left(\mathbf{x}_i^T\mathbf{A}\mathbf{x}_i + \mathbf{x}_i\mathbf{v} + d\right)^2 \qquad (3)$$

Unlike plane fitting, there are 10 parameters for quadric surface fitting. As only 9 parameters are independent, there must exist a natural geometric constraint. A known method for choosing constraints which are invariant with respect to geometric transformations is

$$\mathrm{Tr}(\mathbf{A}\mathbf{A}^T) = \sum_{i=1}^{6} a_i^2 = 1 \qquad (4)$$

Let $\mathbf{p}=(a_1, a_2, \ldots, a_{10})^T$, $\mathbf{p}_1=(a_1, a_2, \ldots, a_6)^T$, $\mathbf{p}_2=(a_7, \ldots, a_{10})^T$, then, Equation (4) can be rewritten as

$$\|\mathbf{p}_1\| = 1 \qquad (5)$$

Because Equation (3) is a quadratic function of the components of **p**, it can be written as

$$\chi^2 = \sum\mathbf{p}^T\mathbf{M}_i\mathbf{p} = \mathbf{p}^T\mathbf{M}\mathbf{p} \qquad (6)$$

where $\mathbf{M} = \sum\mathbf{M}_i$, and the summation is over all data points $\mathbf{x}_i=(x_i, y_i, z_i)$. $\mathbf{M}_i$ is the symmetric matrix

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{B}_i & \mathbf{C}_i \\ \mathbf{C}_i^T & \mathbf{D}_i \end{bmatrix} \qquad (7)$$

By comparing coefficients of Equation (4) and (5), it is not difficult to get $\mathbf{B}_i$, $\mathbf{C}_i$ and $\mathbf{D}_i$.

Then, the minimization of χ^2, with constraint as shown in Equation (4) can be solved by using the method of Lagrangian multipliers (Marshall and Martin 1992). The problem reduces to finding the minimum λ and corresponding vector **p** such that

$$\mathbf{M}\mathbf{p} - \mathbf{p}_1 = 0\,. \qquad (8)$$

Since $\mathbf{p}=(\mathbf{p}_1, \mathbf{p}_2)$, the system of Equation (8) can be split into two parts

$$\mathbf{B}\mathbf{p}_1 + \mathbf{C}\mathbf{p}_2 - \lambda\mathbf{p}_1 = 0 \qquad (9)$$

$$\mathbf{C}^T\mathbf{p}_1 + \mathbf{D}\mathbf{p}_2 = 0 \qquad (10)$$

According to the Lagrangian multipliers method, the minimum value of λ, say λ_{min}, in Equation (9) is the lowest eigenvalue of the matrix $\mathbf{B}\text{-}\mathbf{C}\mathbf{D}^{-1}\mathbf{C}^T$. The unit eigenvector corresponding to λ_{min} is the solution for $\mathbf{p}_1$. Knowing $\mathbf{p}_1$, the solution for $\mathbf{p}_2$ can be obtained from Equation (10) by solving the linear system of equations

$$\mathbf{D}\mathbf{p}_2 = -\mathbf{C}^T\mathbf{p}_1 \qquad (11)$$

Quadric Surface Extraction by a GA

The first step is the creation of an initial population. The initial population is randomly selected from the data points. If there are totally N data points, the maximum size of the population can be C_N^9. In order to accelerate computation, the population size is set to $N-9$.

In the evaluation process, Equation (1) is fitted to all GMS by the Lagrangian multipliers method as described in the above section. For each successful fitting, its CFG is accumulated by checking each data point's PGD.

The following important step is to generate new generations by crossover and mutation. The parent GMSs are randomly selected to generate two children GMSs by crossover. If the cost of either one of the two new children GMSs is greater than the MCP of the population, the corresponding GMS is taken as a new member of the new population, and the GMS which has the MCP is removed from the current population; If the first child GMS's cost is less than the MCP of the current population, it is mutated with another randomly selected GMS. The uniform crossover and mutation operators which switch each gene (element of GMS) according to a uniform probability distribution are used in our research.

The terminating condition is set to be either a threshold C_G for MCP or the maximum number of iterations. According to our experimental experience, C_G should be a positive integer greater than 10. Whenever a run terminates, the fitted surface which has the maximum CFG is taken as the current extracted surface.

After that, all points within a small PGD threshold δ_s are deleted from the measurement data sets, and the above procedures continue to run on the remaining data points until no more surfaces can be extracted.

Finally, every extracted general quadric surface is classified into a special quadric surface according to a classification table given by Levin (Levin 1976).

Experimental Results

Two examples are demonstrated with the proposed method. The first one uses simple synthetic points to demonstrate the recognition of a sphere surface. The second example shows surface extraction from a real set of data points.

Quadric Surface Extraction from 3-D Synthetic Data Points

A set of synthetic data points are generated by selecting 30 points on a unit sphere whose center is at the origin, and 30 points on the XY plane as shown in Figure 1.

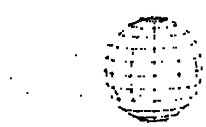

Figure 1: Synthetic points on a unit sphere and XY plane

Set the threshold value of MCP and PGD to C_0=15 and δ_0=0.01 respectively. After 118 iterations, a surface with the following equation is extracted.

$$x^2+0.995y^2+0.998z^2+0.002xy-0.016xz+0.019yz-0.004x-0.005y-0.006z-0.991=0 \quad (12)$$

From Equation (12), all parameters can be used to calculate the related variables for the classification table given by Levin (Levin 1976). It is not difficult to show that the extracted surface is a sphere. Further more, the extracted surface expression can easily be identified as a unit sphere even by simply observe the parameters of Equation (12). It also demonstrates the effectiveness of the proposed method.

Quadric Surface Extraction from 3-D Measurement Data Points

A set of real data points digitized from a hair blower's upper surface are examined by our algorithm. The data points are digitized by a Mitutoyo BLN122 CMM in automatic line-by-line scan mode. The measurement points are shown in Figure 2.

After running the algorithm, 6 surfaces are extracted. For example, in the first run, a surface expression is extracted after 357 iterations. The extracted surface has the following equation.

$$x^2-0.004y^2+1.065z^2-0.012xy-0.01xz-0.032yz-90.18x+3.08y-1.22z-119.13=0 \quad (13)$$

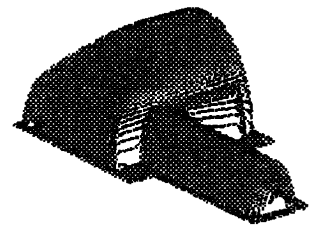

Figure 2: Data points on a hair blower's upper surface

According to the classification table given in reference (Levin 1976), this surface is classified as a hyperboloid of one sheet. All points on this surface are shown in Figure 3.

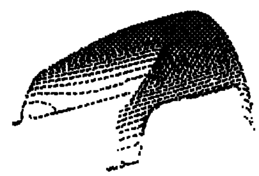

Figure 3: Data points on the first extracted surface

With the remaining points after first run, the above procedures are executed again. Another quadric surface is extracted after 572 iterations. Its equation is shown below.

$$x^2+171.8y^2+116.4z^2+1.4xy+3.0xz-5.9yz-451.8x-31835.0y-534.9z+1457649.4=0 \quad (14)$$

In the same way, the second extracted surface is classified to be an ellipsoid. And all points on this surface are shown in Figure 4.

Figure 4: Data points on the second extracted surface

The above procedure is repeated until not enough points (less than $2C_0$=30 in this experiment) are left in the remaining measurement data set.

Conclusions and Discussions

From the above analysis and experiments, it can be seen that the proposed method is very effective for quadric surfaces extraction. Because most mechanical parts are designed by planes and quadric surfaces, this method can work well for the majority of reverse engineering applications. Take ellipsoid extraction for example, no matter what the size is and where the three main axis's directions are, it can be extracted in just one run.

Another merit of the proposed method is that it does not rely on surface segmentation which is a very difficult problem itself. In fact, the segmentation problem is solved at the same time as surfaces are extracted because boundary curves can be calculated by surface intersections. Because of the combined features of both data-driven and model-driven, the proposed method is more flexible and robust in surface extraction.

Another feature of our algorithm is that it uses pseudo geometric distance in stead of algebraic distance for judging if a point is close enough to the fitted surface. The pseudo geometric distance has an intrinsic geometric significance and also can be easily calculated.

Further works are investigated aiming at accelerating the speed of the algorithm. One of the most important work to be done is the data reduction which can not only reduce the number of data points, but also keep the important points with characteristics of surface properties. These points often have extreme curvature change, and must be kept after data reduction.

References

Puntambekar, N. V.; Jablokow, A. G.; and Sommer III, H J. 1994. Unified Review of 3D Model Generation for Reverse Engineering. *Computer Integrated Manufacturing Systems* 7(4):259-268.

Fitzgibbon, A. W.; Eggert, D. W.; and Fisher, R. B. 1997. High-Level CAD Model Acquisition from Range Images. *Computer-Aided Design* 29(4):321-330.

Varady, T.; Martin, R. R.; and Cox, J. 1997. Reverse Engineering of Geometric Models - An Introduction. *Computer-Aided Design* 29(4):255-268.

Chao, C. H. 1994. Artificial Neural Networks and Model-Based Recognition of Three-Dimensional Objects from Two-Dimensional Images. Ph.D. Diss.. University of Cincinnati.

Oshima, M.; and Shirai, Y. 1983. Object Recognition Using Three-Dimensional Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5(4):353-361.

Sullivan, S.; Sandford, L.; and Ponce, J. 1994. Using Geometric Distance Fits for 3-D Object Modeling and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(12):1183-1196.

Leavers, V. F. 1993. Which Hough transform?. *CVGIP: Image Understanding* 58(1):1-22.

Hill, A.; and Taylor, C. J. 1992. Model-Based Image Interpretation Using Genetic Algorithms. *Image and Vision Computing* 10(5):295-300.

Roth, G.; and Levine, M. D. 1993. Extracting Geometric Primitives. *CVGIP: Image Understanding* 58(1):1-22.

Roth, G.; and Levine, M. D. 1994. Geometric Primitive Extraction Using a Genetic Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(9):901-905.

Eiben, A. E.; Raue, P. E.; and Ruttkay, Z. 1995. How to Apply Genetic Algorithms to Constrained Problems. In Practical Handbook of Genetic Algorithms: Applications, Volume 1, 307-366. Florida, USA: CRC Press, Inc.

Goldberg, D. E. 1989. *Genetic Algorithm in Search, Optimization and Machine Learning.* Reading, Mass.: Addison-Wesley.

Man, K. F.; Tang, K. S.; Kwong S.; and Halang, W. A. 1997. *Genetic Algorithms for Control and Signal Processing.* Great Britain: Springer-Verlag London Limited.

Marshall, A. D.; and Martin, R. R. 1992. *Computer Vision, Models and Inspection.* Singapore: World Scientific.

Levin, J. A. 1976. Parametric Algorithm for Drawing Pictures of Solid Objects Composed of Quadric Surfaces. *Graphics and Image Processing* 19(10):555-563.

Pettofrezzo, A. J.; and Lacatena, M. M. 1970. *Analytic Geometry with Vectors.* Scott:Foresman and Company.