A Revisit to Least Squares Orthogonal Distance Fitting of Parametric Curves and Surfaces

Yang Liu and Wenping Wang

Dept. of Computer Science, The University of Hong Kong,
Pokfulam Road, Hong Kong SAR, P.R. China
{yliu,wenping}@cs.hku.hk

Abstract. Fitting of data points by parametric curves and surfaces is demanded in many scientific fields. In this paper we review and analyze existing least squares orthogonal distance fitting techniques in a general numerical optimization framework. Two new geometric variant methods (GTDM and CDM) are proposed. The geometric meanings of existing and modified optimization methods are also revealed.

Keywords: orthogonal distance fitting, parametric curve and surface fitting, nonlinear least squares, numerical optimization.

1 Introduction

Effective and accurate curve/surface fitting plays an important role and serves as a basic module in CAGD, computer graphics, computer vision and other scientific and engineering fields. We consider a common problem which occurs often in practical applications: fit a parametric curve/surface $\mathbf{C}(\mathbf{P};\mathbf{t})$ (whose parametric form is known but the parameter values are to be determined) to a set of given data points $\{\mathbf{X}_j\}_{j=1}^n \subset \mathbb{R}^s$. Here $\mathbf{P}$ are the shape parameters and $\mathbf{t} = (\tau_1, \ldots, \tau_m)$ are location parameters (For instance, $\mathbf{t}$ of a 3D parametric surface $\mathbf{C}(u,v)$ is (u,v)). This problem is usually stated as a standard nonlinear least squares problem:

$$\min_{\mathbf{P},\mathbf{t}_1,\ldots,\mathbf{t}_n} \sum_{j=1}^{n} \|\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\|^2 \tag{1}$$

Where $\mathbf{t}_j$ is associated with the data point $\mathbf{X}_j$.

There exists vast literature about this problem in mathematics, statistics and computer science. Despite the differences of existing methods in variant contexts, The basics of most methods are the classical optimization theory and the optimization techniques such as decent methods and Gauss-Newton methods [1] appear in different forms.

First we introduce the traditional way of fitting a parametric curve/surface to a given data set in CAGD [2] [3] [4]. The first step is the parametrization which associates the location parameter $\mathbf{t}_j$ to each data point $\mathbf{X}_j$. After substituting $\mathbf{t}_j$ into (1), the second step is solving a linear least squares problem if the shape parameters occur in linear form; for instance, $\mathbf{P}$ are the control points of the B-spline curve/surface. By executing these two steps iteratively, improved location parameters and shape parameters are obtained.

F. Chen and B. Jüttler (Eds.): GMP 2008, LNCS 4975, pp. 384–397, 2008.

This approach has been widely used because of its simplicity. However its convergence rate is slow and is proven to be linear [5]. On the other hand, without separating $\mathbf{P}$ and $\mathbf{t}_1,\ldots,\mathbf{t}_n$, the general optimization techniques of course can be applied. One can optimize $\mathbf{P},\mathbf{t}_1,\ldots,\mathbf{t}_n$ simultaneously [6] [7]. Moreover if $\mathbf{P}$ are in the linear form, the separable nonlinear least squares method (variable projection) can be employed and is better than the simultaneous method [5] [8] [9]. But the size of corresponding nonlinear least squares problem becomes larger when n increases. Therefore these methods are not suitable for fitting a large number of data points. In the metrology and pattern recognition communities people prefer the least squares orthogonal distance technique which is an iterative method and considers the relationship between shape parameters and location parameters. We refer the reader to the papers [10] [11] for detailed references. In [12] a curvature-based squared distance minimization(SDM) is proposed for orthogonal distance fitting for B-spline curve fitting. In this paper we consider general parametric curve/surface fitting problems, which are not limited in 2D, 3D curves and surfaces.

Contributions: Inspired by the approaches in [10] [12], we aim to analyze the existing orthogonal distance techniques by rephrasing them into a general optimization framework. We propose two modified methods CDM and GTDM based on geometric and optimizational analysis. We reveal that the existing and our proposed methods have clear geometric meanings. This better understanding will benefit the general parametric models fitting.

The paper is organized as follows: the basic concepts and necessary optimization techniques are introduced in Section 2; In Section 3 the detailed analysis of orthogonal distance fitting is presented including the derivation of the geometric meanings and the modified methods; in Section 4, we illustrate the effectiveness of different methods by numerical examples; finally we close the paper by the conclusion in Section 5.

2 Preliminary

2.1 Notations

Let $\mathbf{C}(\mathbf{P};\mathbf{t}) \subset \mathbb{R}^s$ represent a family of parametric curves or surfaces. A set of points $\{\mathbf{X}_j\}_{j=1}^n \subset \mathbb{R}^s$ are to be approximated by $\mathbf{C}(\mathbf{P};\mathbf{t})$. Here $\mathbf{t} = (\tau_1,\ldots,\tau_m) \in \mathbb{R}^m$ is the location parameter and $\mathbf{P} = (p_1,\ldots,p_r)$ is the shape parameter. For instance, if $m = 1$, $\mathbf{C}(\mathbf{P};\mathbf{t})$ represents a parametric curve. We assume that $\mathbf{C}(\mathbf{P};\mathbf{t})$ has C^2 continuity. In this paper vectors and matrices are denoted by bold face and vectors are in the column format. The first-order partial derivatives of $\mathbf{C}(\mathbf{P};\mathbf{t})$ are denoted as follows:

$$\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t})}{\partial \mathbf{P}} = \left[\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t})}{\partial p_1},\ldots,\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t})}{\partial p_r}\right], \quad \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t})}{\partial \mathbf{t}} = \left[\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t})}{\partial \tau_1},\ldots,\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t})}{\partial \tau_m}\right]$$

$$\nabla_{\mathbf{P}}\mathbf{t} = \begin{bmatrix} \frac{\partial \tau_1}{\partial p_1} & \cdots & \frac{\partial \tau_m}{\partial p_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \tau_1}{\partial p_r} & \cdots & \frac{\partial \tau_m}{\partial p_r} \end{bmatrix}.$$

In many curves and surfaces fitting applications, the initial positions of data points and the model are not aligned well. The data points or the model is allowed to be transformed in the fitting process. By introducing proper transformation, the fitting process can be accelerated and overcome some local minimum cases. The most common transformation is rigid transformation[10], [11]. Combined with rigid transformation, we have shown in [13] that the convergence speed of the fitting algorithm can be faster and high accuracy also can be achieved. Although the transformation can be applied to the data points or the model, for unifying our analysis we assume the transformation is applied on the parametric model, i.e. the shape parameter $\mathbf{P}$ can contain the transformation parameters if needed.

2.2 Nonlinear Least Squares

We consider a standard nonlinear least squares problem which minimizes the objective function $F(\mathbf{X})$:

$$\min_{\mathbf{X}} \frac{1}{2}\sum_{i=1}^{n} f_i^2(\mathbf{X}) \triangleq F(\mathbf{X}) \tag{2}$$

The residual vector is defined as $\mathbf{r}(\mathbf{X}) = (f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_n(\mathbf{X}))^T$. The first derivative of $F(\mathbf{X})$ can be expressed in terms of the Jacobian of r: $\mathbf{J}(\mathbf{X}) = \begin{pmatrix} \nabla f_1(\mathbf{X}) \\ \vdots \\ \nabla f_n(\mathbf{X}) \end{pmatrix}$, where $\nabla f_i(\mathbf{X})$ is the gradient of f_i with respect to $\mathbf{X}$. The gradient and Hessian of $F(\mathbf{X})$ have the following forms

$$\nabla F(\mathbf{X}) = \mathbf{J}(\mathbf{X})^T \mathbf{r}(\mathbf{X}); \qquad \mathbf{H} = \nabla^2 F(x) = \mathbf{J}(\mathbf{X})^T \mathbf{J}(\mathbf{X}) + \sum_{i=1}^{n} f_i(\mathbf{X}) \nabla^2 f_i(\mathbf{X})$$

The Gauss-Newton method approximates the Hessian by $\mathbf{J}(\mathbf{X})^T \mathbf{J}(\mathbf{X})$. In practice the line search strategy or the Levenberg-Marquardt method

$$\left(\mathbf{J}(\mathbf{X})^T \mathbf{J}(\mathbf{X}) + \lambda \mathbf{I}\right) \delta \mathbf{X} = -\mathbf{J}(\mathbf{X})^T \mathbf{r}(\mathbf{X})$$

is incorporated with the Gauss-Newton method. The Quasi-Newton type method approximates the Hessian or the inverse of the Hessian by a positive-definite matrix which is updated at each iteration with some specified schemes such as **BFGS** [1]. But in this paper we mainly focus on Gauss-Newton type methods.

2.3 Principal Directions and Curvatures of Parametric Curves and Surfaces

For a smooth parametric curve/surface $\mathbf{C}(\mathbf{t})$, its first-order derivatives $\partial_{\tau_1}\mathbf{C}(\mathbf{t}_p)$, …, $\partial_{\tau_m}\mathbf{C}(\mathbf{t}_p)$ at point $\mathbf{C}(\mathbf{t}_p)$ span a tangential space $\top_p\mathbf{C}$. Its orthogonal complement defines the normal space $\perp_p\mathbf{C}$. For a given unit normal vector $\mathbf{n}_p \in \perp_p\mathbf{C}$, we can define the principal vectors and curvatures with respect to $\mathbf{n}_p$. The details can be found in Section 2.2 of [14]. Let $\mathbf{T}_1, \ldots, \mathbf{T}_m$ be the principle vectors which span $\top_p\mathbf{C}$ and $\kappa_1, \ldots, \kappa_m$ be the corresponding principle curvatures with respect to $\mathbf{n}_p$. The orthonormal basis

of $\perp_p \mathbf{C}$ are $\mathbf{N}_{m+1}, \ldots, \mathbf{N}_s$. One identity about the orthonormal basis will be useful in the paper:

$$\mathbf{I}_s = \mathbf{T}_1\mathbf{T}_1^T + \cdots + \mathbf{T}_m\mathbf{T}_m^T + \mathbf{N}_{m+1}\mathbf{N}_{m+1}^T + \cdots + \mathbf{N}_s\mathbf{N}_s^T. \tag{3}$$

Where $\mathbf{I}_s$ is a $s \times s$ identity matrix.

Remark: For a 3D parametric curve, the curvature K and curvature direction $\mathbf{N}^0$ are well defined from differential geometry. Since in our discussion $\mathbf{N}$ is not necessarily coincident with $\mathbf{N}^0$, we have $\kappa = K \cdot < \mathbf{N}, \mathbf{N}^0 >$. $< \star, \star >$ is the inner product of two vectors.

3 Orthogonal Distance Fitting

The optimization process of orthogonal distance fitting contains two steps which are executed repeatedly:

1. Reparametrization: compute the foot-point of $\mathbf{X}_j$ on $\mathbf{C}(\mathbf{P};\mathbf{t})$, i.e, minimize the distance from $\mathbf{X}_j$ to $\mathbf{C}$:

$$\min_{\mathbf{t}_j} \|\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\|, j = 1, \ldots, n \tag{4}$$

2. minimize one of the following objective functions by applying one step of optimization techniques such as Gauss-Newton methods:

$$\min_{\mathbf{P}} \left\| \left(\|\mathbf{C}(\mathbf{P};\mathbf{t}_1(\mathbf{P})) - \mathbf{X}_1\|, \ldots, \|\mathbf{C}(\mathbf{P};\mathbf{t}_n(\mathbf{P})) - \mathbf{X}_n\| \right)^T \right\| \tag{5}$$

or

$$\min_{\mathbf{P}} \left\| \left(\mathbf{C}(\mathbf{P};\mathbf{t}_1(\mathbf{P}))^T - \mathbf{X}_1^T, \ldots, \mathbf{C}(\mathbf{P};\mathbf{t}_n(\mathbf{P}))^T - \mathbf{X}_n^T \right)^T \right\| \tag{6}$$

Since (5) minimizes the l_2 norm of the residual vector $\mathbf{r}_d$:

$$\mathbf{r}_d = \left(\|\mathbf{C}(\mathbf{P};\mathbf{t}_1(\mathbf{P})) - \mathbf{X}_1\|, \ldots, \|\mathbf{C}(\mathbf{P};\mathbf{t}_n(\mathbf{P})) - \mathbf{X}_n\| \right)^T, \tag{7}$$

the corresponding method is called *Distance-based* method; also since (6) minimizes the l_2 norm of the residual vector $\mathbf{r}_c$:

$$\mathbf{r}_c = \left(\mathbf{C}(\mathbf{P};\mathbf{t}_1(\mathbf{P}))^T - \mathbf{X}_1^T, \ldots, \mathbf{C}(\mathbf{P};\mathbf{t}_n(\mathbf{P}))^T - \mathbf{X}_n^T \right)^T, \tag{8}$$

the corresponding method is called *Coordinate-based* method. By applying nonlinear least squares optimization technique these two methods produce different results. Atieg and Watson present their analysis on *Distance-based* and *Coordinate-based* Gauss-Newton approaches in [10]. We will show the geometry behind these two methods and their variations.

Orthogonality: Because $\mathbf{t}_j$ is the minimizer of (4), the orthogonality condition (9) below always holds in each step, except when the foot-point is at the boundary of $\mathbf{C}$.

$$\left\langle \mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j, \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_k} \right\rangle = 0, \qquad j = 1, \ldots, n; k = 1, \ldots, m \tag{9}$$

The orthogonality condition (9) plays an important role in parametrization correction and optimization. Many effective foot-point computation methods are available in literature [3] [15] [16]. If the explicit foot-point formula is not available, one can apply Newton-like optimization methods on (9) to obtain the foot-point and corresponding location parameter. But the initial guess $\mathbf{t}^0$ is a key issue in foot-point computation. For complex parametric curves/surfaces, one good strategy is to build a k-D tree from the sample points $\{\mathbf{C}(\mathbf{P};\mathbf{t}_k), k = 1,\ldots,L\}$ then find the nearest point for $\mathbf{X}_j$ which serves as the initial foot-point.

3.1 Distance-Based Gauss-Newton Method

Distance-based methods are widely used in metrology. Here the l_2 norm of residual vector $\mathbf{r}_d$ is to be minimized. Depending on whether considering the association between the shape parameter $\mathbf{P}$ and the local parameter $\mathbf{t}$, Gauss-Newton distance-based methods can be categorized to two types: the separated method and the standard method.

(1) Separated distance-based Gauss-Newton method

The residual vector $\mathbf{r}_d$ in the separated distance-based Gauss-Newton method is defined as

$$\mathbf{r}_d = \left(\left\|\mathbf{C}\left(\mathbf{P};\mathbf{t}_1\right) - \mathbf{X}_1\right\|, \ldots, \left\|\mathbf{C}\left(\mathbf{P};\mathbf{t}_n\right) - \mathbf{X}_n\right\|\right)^T,$$

where each $\mathbf{t}_j$ is fixed. The first-order total derivative of $\left\|\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right) - \mathbf{X}_j\right\|$ with respect to $\mathbf{P}$ is

$$\nabla_{\mathbf{P}} \left\|\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right) - \mathbf{X}_j\right\| = \frac{\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)^T - \mathbf{X}_j^T}{\left\|\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right) - \mathbf{X}_j\right\|} \frac{\partial \mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)}{\partial \mathbf{P}}, \tag{10}$$

where it must be assumed that $\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right) \neq \mathbf{X}_j$ such that the derivative exists. Numerical computation can be unstable when $\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)$ approaches $\mathbf{X}_j$. Notice that if $\mathbf{C}$ is a 2D parametric curve or a 3D parametric surface, the vector $\dfrac{\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right) - \mathbf{X}_j}{\left\|\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right) - \mathbf{X}_j\right\|} := \mathbf{N}_j$ actually is the unit normal at $\mathbf{C}(\mathbf{t}_j)$ whose sign may be positive or negative. Thus the instability can be eliminated if we replace it with the unit normal. The Jacobian of $\mathbf{r}_d$ at $\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)$ can be written as

$$\mathbf{J}_1 = \begin{pmatrix} \mathbf{N}_1^T \dfrac{\partial \mathbf{C}\left(\mathbf{P};\mathbf{t}_1\right)}{\partial \mathbf{P}} \\ \vdots \\ \mathbf{N}_n^T \dfrac{\partial \mathbf{C}\left(\mathbf{P};\mathbf{t}_n\right)}{\partial \mathbf{P}} \end{pmatrix}.$$

From the normal equation $\mathbf{J}_1^T \mathbf{J}_1 \cdot \delta\mathbf{P} = -\mathbf{J}_1^T \mathbf{r}_d$, we can derive that

$$\sum_{j=1}^{n} \frac{\partial \mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)^T}{\partial \mathbf{P}} \mathbf{N}_j \mathbf{N}_j^T \frac{\partial \mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)}{\partial \mathbf{P}} \cdot \delta\mathbf{P} = -\sum_{j=1}^{n} \frac{\partial \mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)^T}{\partial \mathbf{P}} \left(\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right) - \mathbf{X}_j\right), \tag{11}$$

where $\delta\mathbf{P}$ is the increment of the shape parameter $\mathbf{P}$.

Now we show the geometric meaning behind (11). the right hand side of (11) can be rewritten as:

$$\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)^T}{\partial \mathbf{P}}(\mathbf{C}(\mathbf{P};\mathbf{t}_j)-\mathbf{X}_j)=\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)^T}{\partial \mathbf{P}}\mathbf{N}_j\cdot\mathbf{N}_j^T(\mathbf{C}(\mathbf{P};\mathbf{t}_j)-\mathbf{X}_j) \tag{12}$$

Now Eqn.(11) actually minimizes the squared distance from data points to their tangent planes at the foot-points:

$$\min_{\mathbf{P}}\sum_{j=1}^{n}\left[\mathbf{N}_j^T\cdot(\mathbf{C}(\mathbf{P};\mathbf{t}_j)-\mathbf{X}_j)\right]^2 \tag{13}$$

It is easy to verify the normal equation of Eqn.(13) is Eqn.(11) just by applying the Gauss-Newton method on Eqn.(13). We call this kind of geometric minimization TDM (**T**angent **D**istance **M**inimization) [12].

As we have pointed out, there is no numerical problem for 2D parametric curves and 3D parametric surfaces if we replace $\mathbf{N}_j$ with curves/surfaces' normals. For high dimension parametric curves/surfaces ($m<s-1$), TDM is not suitable when the data points are almost contained in a low dimension space $\mathbb{R}^l, l<s$. For instance, fitting a 3D parametric curve to a set of points in a plane causes the ill-conditioning of Jacobian matrix [10]. We use a simple example to illustrate this problem. Assume that a 3D curve has the following parametric form (at^2,bt^3,c), where a,b,c are shape parameters and the data points lie in the *x*-*y* plane. The third component of $\mathbf{N}_j$ will be always zero. It means that c does not appear in $\mathbf{N}_j^T\cdot(\mathbf{C}(\mathbf{P};\mathbf{t}_j)-\mathbf{X}_j)$. Therefore the normal equations will be singular.

(2) Standard distance-based Gauss-Newton method

With the consideration of the association between $\mathbf{t}$ and $\mathbf{P}$, in the standard distance-based Gauss-Newton method the residual vector $\mathbf{r}_d$ is defined as in (7). The first-order total derivative of each element of $\mathbf{r}_d$ with respect to $\mathbf{P}$ is

$$\begin{aligned}
&\nabla_{\mathbf{P}}\left\|\mathbf{C}(\mathbf{P};\mathbf{t}_j(\mathbf{P}))-\mathbf{X}_j\right\|=\frac{\mathbf{C}(\mathbf{P};\mathbf{t}_j)^T-\mathbf{X}_j^T}{\left\|\mathbf{C}(\mathbf{P};\mathbf{t}_j)-\mathbf{X}_j\right\|}\nabla_{\mathbf{P}}\mathbf{C}(\mathbf{P};\mathbf{t}_j(\mathbf{P}))\\
&=\frac{\mathbf{C}(\mathbf{P};\mathbf{t}_j)^T-\mathbf{X}_j^T}{\left\|\mathbf{C}(\mathbf{P};\mathbf{t}_j)-\mathbf{X}_j\right\|}\left[\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}}+\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{t}}\nabla_{\mathbf{P}}\mathbf{t}_j\right]\\
&=\mathbf{N}_j^T\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}},
\end{aligned} \tag{14}$$

where the term $\left(\mathbf{C}(\mathbf{P};\mathbf{t}_j(\mathbf{P}))^T-\mathbf{X}_j^T\right)\cdot\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{t}}\nabla_{\mathbf{P}}\mathbf{t}_j$ is eliminated due to the orthogonality condition. The result (14) is the same as (10), which means that both separated and standard distance-based approaches produce the same geometric minimization scheme – TDM.

3.2 Coordinate-Based Gauss-Newton Method

Now we consider the coordinate-based Gauss-Newton method based on the objective function (6), which is widely used in pattern recognition community.

(1) Separated coordinate-based Gauss-Newton method

In the separated coordinate-based Gauss-Newton method the residual vector $\mathbf{r}_c$ is defined as

$$\mathbf{r}_c = \left(\mathbf{C}(\mathbf{P};\mathbf{t}_1)^T - \mathbf{X}_1^T, \ldots, \mathbf{C}(\mathbf{P};\mathbf{t}_n)^T - \mathbf{X}_n^T\right)^T.$$

The first-order total derivative of $\mathbf{C}(\mathbf{P};\mathbf{t}_j)^T - \mathbf{X}_j^T$ with respect to $\mathbf{P}$ is $\dfrac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)^T}{\partial \mathbf{P}}$. So the Jacobian $\mathbf{J}_2$ of $\mathbf{r}_c$ is

$$\begin{pmatrix} \dfrac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_1)}{\partial \mathbf{P}} \\ \vdots \\ \dfrac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_n)}{\partial \mathbf{P}} \end{pmatrix}.$$

Still from the normal equation $\mathbf{J}_2^T\mathbf{J}_2 \cdot \delta\mathbf{P} = -\mathbf{J}_2^T\mathbf{r}_c$, we obtain

$$\sum_{j=1}^{n} \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)^T}{\partial \mathbf{P}} \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} \cdot \delta\mathbf{P} = -\sum_{j=1}^{n} \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} \left(\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\right) \tag{15}$$

The normal equation actually represents a geometric minimization

$$\min_{\mathbf{P}} \sum_{j=1}^{n} \left[\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\right]^2 \tag{16}$$

which penalizes the squared distance from data points to foot points, we call this method PDM (**P**oint **D**istance **M**inimization). It is widely used in CAGD community because of its simplicity. Especially when $\mathbf{P}$ is in the linear form in $\mathbf{C}$, one just needs to solve a linear equation and the $\|\mathbf{r}_c\|$ always decreases. However PDM only exhibits linear convergence [12].

(2) Standard coordinate-based Gauss-Newton method

In the standard distance-based Gauss-Newton method the residual vector $\mathbf{r}_c$ is (8), where $\mathbf{t}_j$ is associated with $\mathbf{P}$ through (9). The first-order total derivative of each element with respect to $\mathbf{P}$ is

$$\nabla_{\mathbf{P}} \left(\mathbf{C}(\mathbf{P};\mathbf{t}_j(\mathbf{P})) - \mathbf{X}_j\right) = \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} + \sum_{k=1}^{m} \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,k}} \nabla_{\mathbf{P}} \tau_{j,k}(\mathbf{P}) \tag{17}$$

In general the explicit expression of $\tau_{j,k}(\mathbf{P})$ with respect to $\mathbf{P}$ is not always available. So we use the implicit procedure presented in [10]. Since the orthogonality condition (9)

holds and it is an identity in $\mathbf{P}$, its total derivative with respect to $\mathbf{P}$ is still $\mathbf{0}$. Therefore we have

$$\begin{aligned}\mathbf{0} &= \nabla_{\mathbf{P}} \left\langle \mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j, \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,k}} \right\rangle \\ &= \left\langle \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} + \sum_{l=1}^{m} \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,l}} \nabla_{\mathbf{P}} \tau_{j,l}(\mathbf{P}), \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,k}} \right\rangle + \\ &\quad \left\langle \mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j, \frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,k} \partial \mathbf{P}} + \sum_{l=1}^{m} \frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,k} \partial \tau_{j,l}} \nabla_{\mathbf{P}} \tau_{j,l}(\mathbf{P}) \right\rangle\end{aligned}$$

Without loss of generality, suppose $\mathbf{C}(\mathbf{P};\mathbf{t}_j)$ is a local regular parametrization such that $\tau_{j,1}$-, …, $\tau_{j,m}$- direction vectors are unit principle direction vectors $\mathbf{T}_{j,1}, \ldots, \mathbf{T}_{j,m}$ with respect to $\mathbf{N}_j$ (see Section 2.3). The above equation can be simplified as

$$\begin{aligned}\mathbf{0} &= \left\langle \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} + \sum_{l=1}^{m} \mathbf{T}_{j,l} \nabla_{\mathbf{P}} \tau_{j,l}(\mathbf{P}), \mathbf{T}_{j,k} \right\rangle + \\ &\quad \left\langle \mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j, \frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,k} \partial \mathbf{P}} + \frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial {\tau_{j,k}}^2} \nabla_{\mathbf{P}} \tau_{j,k}(\mathbf{P}) \right\rangle \\ &= \mathbf{T}_{j,k}^T \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} + \sum_{l=1}^{m} \mathbf{T}_{j,k}^T \mathbf{T}_{j,l} \nabla_{\mathbf{P}} \tau_{j,l}(\mathbf{P}) + (\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j)^T \frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,k} \partial \mathbf{P}} + d_j \mathbf{N}_j^T \kappa_{j,k} \mathbf{N}_j \nabla_{\mathbf{P}} \tau_{j,k}(\mathbf{P}) \\ &= \mathbf{T}_{j,k}^T \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} + (1 + d_j \kappa_{j,k}) \nabla_{\mathbf{P}} \tau_{j,k}(\mathbf{P}) + (\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j)^T \frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,k} \partial \mathbf{P}},\end{aligned}$$

where $d_j = \|\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\|$, $\kappa_{j,k}$ is the principle curvature along $\mathbf{T}_{j,k}$ with respect to $\mathbf{N}_j$. Then we obtain

$$\nabla_{\mathbf{P}} \tau_{j,k} = -\frac{\mathbf{T}_{j,k}^T \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} + (\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j)^T \frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \tau_{j,k} \partial \mathbf{P}}}{1 + d_j \kappa_{j,k}} \tag{18}$$

We can rewrite (17) as

$$\nabla_{\mathbf{P}} (\mathbf{C}(\mathbf{P};\mathbf{t}_j(\mathbf{P})) - \mathbf{X}_j) = \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} - \sum_{k=1}^{m} \frac{\mathbf{T}_{j,k} \mathbf{T}_{j,k}^T \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} + d_j \mathbf{T}_{j,k} \mathbf{N}_j^T \frac{\partial \mathbf{T}_{j,k}}{\partial \mathbf{P}}}{1 + d_j \kappa_{j,k}} \tag{19}$$

In the degenerate case when $1 + d_j \kappa_{j,k} \approx 0$, one can modify the denominator to $1 + d_j |\kappa_{j,k}|$ to improve the condition number of the normal equation. We note that this degenerate case is not addressed in the literature of orthogonal distance fitting, such as [10], [11]. But it can happen in practice. For example, let a parametric circle be $(r\cos t, r\sin t)$ and one data point $\mathbf{X}_j$ be near to the origin. We have $1 + d_j \kappa_{j,k} \approx 1 + r \cdot \frac{-1}{r} = 0$.

(3) Modified standard coordinate-based Gauss-Newton methods

The computational cost of the second-order derivatives $\frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{t}^2}$ and $\frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}\partial \mathbf{t}}$ may be high in some applications. So we shall derive two kinds of modified standard Gauss-Newton methods with less computation cost and clear geometric meanings.

First we recall the notations in Section 2.3. At the point $\mathbf{C}(\mathbf{t}_j)$, $\mathbf{T}_{j,1},\ldots,\mathbf{T}_{j,m}$ span a tangent vector space $\top_{j,p}\mathbf{C}$ and $\mathbf{N}_{j,m+1},\ldots,\mathbf{N}_{j,s}$ denote the orthonormal basis of $\top_{j,p}\mathbf{C}$'s orthogonal complement space $\perp_{j,p}\mathbf{C}$. The following identity always holds

$$\mathbf{I} = \mathbf{T}_{j,1}\mathbf{T}_{j,1}^T + \cdots + \mathbf{T}_{j,m}\mathbf{T}_{j,m}^T + \mathbf{N}_{j,m+1}\mathbf{N}_{j,m+1}^T + \cdots + \mathbf{N}_{j,s}\mathbf{N}_{j,s}^T. \tag{20}$$

We will use this identity in our following derivation. By dropping the second-order derivatives from Eqn. 19, we will derive two methods.

1. Drop $\frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}\partial \mathbf{t}}$. This leads to

$$\begin{aligned}
\nabla_{\mathbf{P}}\left(\mathbf{C}(\mathbf{P};\mathbf{t}_j(\mathbf{P})) - \mathbf{X}_j\right) &\approx \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} - \sum_{k=1}^{m} \frac{\mathbf{T}_{j,k}\mathbf{T}_{j,k}^T \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}}}{1+d_j\kappa_{j,k}} \\
&= \mathbf{I}\cdot\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}} - \sum_{k=1}^{m} \frac{\mathbf{T}_{j,k}\mathbf{T}_{j,k}^T \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}}}{1+d_j\kappa_{j,k}} \\
&= \left(\sum_{k=1}^{m} \frac{d_j\kappa_{j,k}\mathbf{T}_{j,k}\mathbf{T}_{j,k}^T}{1+d_j\kappa_{j,k}} + \sum_{k=m+1}^{s} \mathbf{N}_{j,k}\mathbf{N}_{j,k}^T\right)\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}}
\end{aligned}$$

Substituting the above result into the normal equation $\mathbf{J}^T\mathbf{J}\cdot\delta\mathbf{P} = -\mathbf{J}^T\mathbf{r}_c$, we obtain

$$\begin{aligned}
&\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)^T}{\partial \mathbf{P}}\left(\sum_{k=1}^{m} \frac{\left(d_j\kappa_{j,k}\right)^2\mathbf{T}_{j,k}\mathbf{T}_{j,k}^T}{\left(1+d_j\kappa_{j,k}\right)^2} + \sum_{k=m+1}^{s} \mathbf{N}_{j,k}\mathbf{N}_{j,k}^T\right)\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}}\delta\mathbf{P} = \\
&\frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)^T}{\partial \mathbf{P}}\left(\sum_{k=1}^{m} \frac{d_j\kappa_{j,k}\mathbf{T}_{j,k}\mathbf{T}_{j,k}^T}{1+d_j\kappa_{j,k}} + \sum_{k=m+1}^{s} \mathbf{N}_{j,k}\mathbf{N}_{j,k}^T\right)\left(\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\right).
\end{aligned}$$

The normal equation represents the following geometric minimization

$$\begin{aligned}
\min_{\mathbf{P}} \sum_{j=1}^{n} \Bigg\{ &\sum_{k=1}^{m} \frac{\left(d_j\kappa_{j,k}\right)^2}{\left(1+d_j\kappa_{j,k}\right)^2}\left[\mathbf{T}_{j,k}^T\cdot\left(\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\right)\right]^2 + \\
&\sum_{k=m+1}^{s}\left[\mathbf{N}_{j,k}^T\cdot\left(\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\right)\right]^2\Bigg\}
\end{aligned} \tag{21}$$

We will call it CDM (**C**urvature **D**istance **M**inimization).

2. Drop $\frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}\partial \mathbf{t}}$ and $\frac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{t}^2}$. It is easy to verify in this case that the normal equation corresponds to the following geometric minimization

$$\min_{\mathbf{P}} \sum_{j=1}^{n} \left\{ \sum_{k=m+1}^{s}\left[\mathbf{N}_{j,k}^T\cdot\left(\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\right)\right]^2 \right\} \tag{22}$$

Since (22) only penalizes the squared distance from data point $\mathbf{X}_j$ to the tangent space $\mathbf{T}_{j,p}(\mathbf{C})$ we call it GTDM (**G**eneralized **T**angent **D**istance **M**inimization). This scheme does not suffer from the ill conditioning problem of high dimension parametric curves/surfaces ($m < s-1$) which is mentioned before. Still using the same example at the end of the last subsection, let one normal be $\mathbf{N}_j$ and another normal be $\mathbf{N}_z = (0,0,1)^T$. The variable c will appear in $\mathbf{N}_z^T\left(\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\right)$, so that rank deficiency of the normal equation is avoided.

3.3 SDM - Modified Hessian Approximation

So far our discussion is based on Gauss-Newton methods. Now we look at the Hessian directly. Wang et al. [12] proposed a curvature based squared distance minimization method called SDM where the Hessian is modified to be definite-positive. We do not go into the details and just describe the basic idea here. For each $\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j$, its second-order derivatives $\dfrac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}^2}$ and $\dfrac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}\partial \mathbf{t}}$ are dropped. So the modified Hessian is

$$\widetilde{\mathbf{H}} = \sum_{j=1}^{n} \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)^T}{\partial \mathbf{P}} \left[\sum_{k=1}^{m} \frac{d_j \kappa_{j,k}}{1 + d_j \kappa_{j,k}} \mathbf{T}_{j,k} \mathbf{T}_{j,k}^T + \sum_{k=m+1}^{s} \mathbf{N}_{j,k} \mathbf{N}_{j,k}^T \right] \frac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}}. \tag{23}$$

The corresponding geometric minimization is

$$\min_{\mathbf{P}} \sum_{j=1}^{n} \left\{ \sum_{k=1}^{m} \frac{d_j \kappa_{j,k}}{1 + d_j \kappa_{j,k}} \left[\mathbf{T}_{j,k}^T \cdot \left(\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j \right) \right]^2 + \sum_{k=m+1}^{s} \left[\mathbf{N}_{j,k}^T \cdot \left(\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j \right) \right]^2 \right\}. \tag{24}$$

Remark: If, besides $\dfrac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}^2}$ and $\dfrac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}\partial \mathbf{t}}$, we also drop $\dfrac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{t}^2}$ from the Hessian, SDM will become GTDM. Thus GTDM also is an approximation of the Hessian.

3.4 Comparisons

We summarize the geometric minimization schemes introduced in previous sections in Table 1 and compare them in several aspects.

Computational cost: The standard coordinate-based Gauss-Newton method (for short, we call it GN) is the most expensive method because of computations of the second-order derivatives $\dfrac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{t}^2}$ and $\dfrac{\partial^2 \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{P}\partial \mathbf{t}}$. Since CDM and SDM have similar expressions, their computational cost are the same. GTDM only involves computation of $\dfrac{\partial \mathbf{C}(\mathbf{P};\mathbf{t}_j)}{\partial \mathbf{t}}$ for constructing the normal space if $m < s-1$. TDM and PDM do not need to compute any derivative of $\mathbf{C}(\mathbf{P};\mathbf{t})$ with respect to $\mathbf{t}$. thus they are more efficient than the others in constructing the approximated Hessian.

Applicability: With proper step-size control or combining Levenberg-Marquardt methods, most methods are suitable for general parametric curve/surface fitting. Only TDM

Table 1. Geometric minimization schemes

Method	Geometric terms
PDM	$\left[\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)-\mathbf{X}_j\right]^2$
TDM	$\left[\mathbf{N}_j^T\cdot\left(\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)-\mathbf{X}_j\right)\right]^2$
GTDM	$\sum_{k=m+1}^{s}\left[\mathbf{N}_{j,k}^T\cdot\left(\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)-\mathbf{X}_j\right)\right]^2$
CDM	$\sum_{k=1}^{m}\frac{\left(d_j\kappa_{j,k}\right)^2}{\left(1+d_j\kappa_{j,k}\right)^2}\left[\mathbf{T}_{j,k}^T\cdot\left(\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)-\mathbf{X}_j\right)\right]^2+\sum_{k=m+1}^{s}\left[\mathbf{N}_{j,k}^T\cdot\left(\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)-\mathbf{X}_j\right)\right]^2$
SDM	$\sum_{k=1}^{m}\frac{d_j\kappa_{j,k}}{1+d_j\kappa_{j,k}}\left[\mathbf{T}_{j,k}^T\cdot\left(\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)-\mathbf{X}_j\right)\right]^2+\sum_{k=m+1}^{s}\left[\mathbf{N}_{j,k}^T\cdot\left(\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)-\mathbf{X}_j\right)\right]^2$

may have problems in fitting high dimension parametric curves/surfaces, i.e, when $m<s-1$.

Convergence: Because GN and TDM are standard Gauss-Newton methods, they show quadratic convergence for zero residual problems, super-linear convergence for small residual problems and linear convergence in other cases. For our modified methods CDM and GTDM, they also have the same convergence as TDM. One can see that when $\|\mathbf{C}\left(\mathbf{P};\mathbf{t}_j\right)-\mathbf{X}_j\|$ approaches zero, the second-order derivatives in Eqn. (19) can be ignored so that the Hessian is still well approximated. Unfortunately PDM is an alternating method which is a typical optimization technique for solving a separable nonlinear least squares problem and is known to have only linear convergence [5].

Remark: For high dimension curves/surfaces fitting, i.e. $s>3$, the principle curvature computation can be expensive. In this case GTDM is a good candidate under the consideration of performance and effectiveness. Also when $m=s-1$, GTDM is reduced to TDM actually.

4 Numerical Experiments

Now we compare the methods introduced in Section 3: PDM, TDM, CDM, GTDM, GN, SDM. For demonstrating the effectiveness of GTDM, we choose a planar ellipse in 3D space as our parametric models and a point cloud with different scale noises(For general comparison in 2D/3D curve and surface fitting, we refer the reader to the references [10,11,13,12]). In our implementation the Levenberg-Marquardt method is integrated.

Example: We consider fitting an ellipse to 200 data points sampled from an ellipse: $(\cos\frac{2\pi i}{200},2\sin\frac{2\pi i}{200},0), i=0,1,\ldots,199$ in 3D. The parametric ellipse has the following form which involves rotation and translation

$$\begin{pmatrix}x\\y\\z\end{pmatrix}=\mathbf{R}_x\cdot\mathbf{R}_y\cdot\mathbf{R}_z\cdot\begin{pmatrix}a\cos t\\b\sin t\\0\end{pmatrix}+\begin{pmatrix}c_x\\c_y\\c_z\end{pmatrix}$$

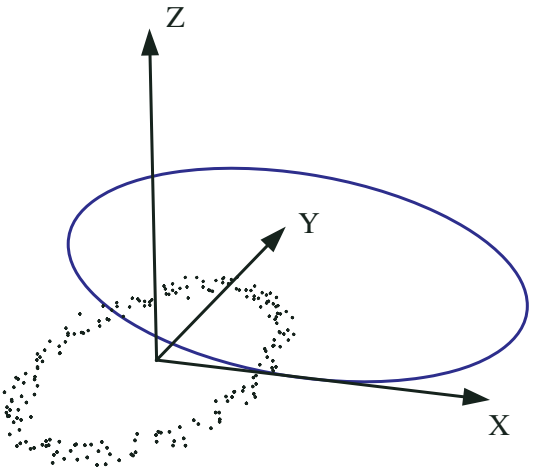

Fig. 1. The initial ellipse and data points of Case 4

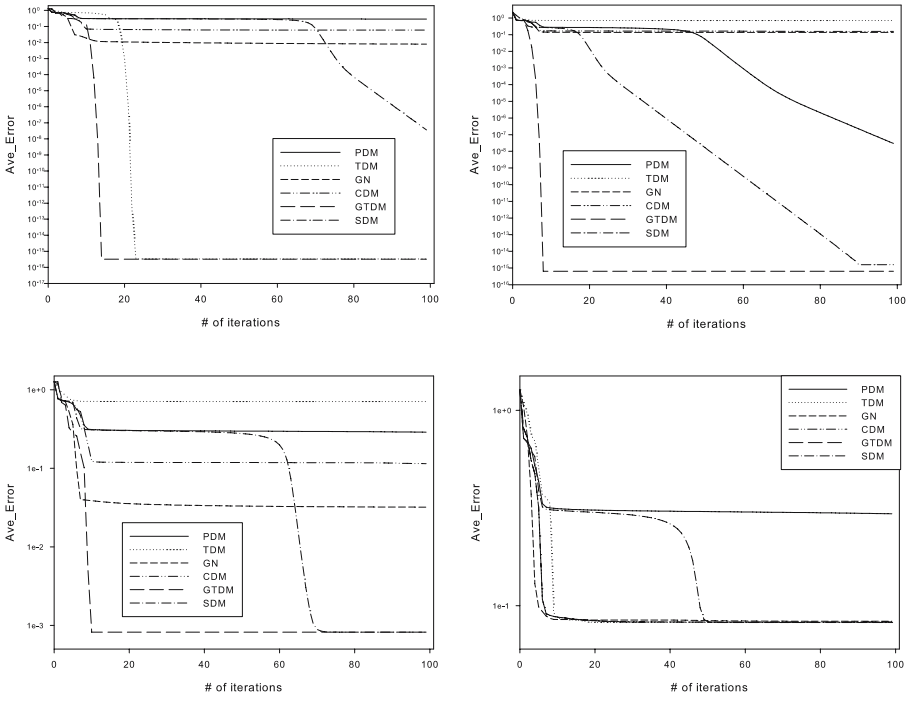

Fig. 2. Comparisons of the six methods on a set of 200 data points. Upper left: Case 1; upper right: Case 2; lower left: Case 3; lower right: Case 4

Where $\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix}$, $\mathbf{R}_y = \begin{pmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{pmatrix}$, $\mathbf{R}_z = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$.
The shape parameters are $\mathbf{P} = [a, b, c_x, c_y, c_z, \alpha, \beta, \gamma]$. We choose four cases to illustrate the convergence of each method, with the following initial values for $\mathbf{P}$

- Case 1: $\mathbf{P} = [3.1, 1.0, 1.0, 2.0, 0.2, 4.0, 1.0, 6.0]$;
- Case 2: $\mathbf{P} = [0.1, 4.0, 2.0, 0.0, 1.0, 1.0, -1.0, 2.0]$;
- Case 3: same $\mathbf{P}$ as in Case 1 but perturb the data points with random noise distributed uniformly in $[-0.001, 0.001]$;
- Case 4: same $\mathbf{P}$ as in Case 1 but perturb the data points with random noise distributed uniformly in $[-0.1, 0.1]$. (See Fig. 1)

Fig.2 shows that the the average error versus the number of iterations of the six methods. The average error is defined as $\sqrt{\frac{\sum_{j=1}^{n} \|\mathbf{C}(\mathbf{P};\mathbf{t}_j) - \mathbf{X}_j\|^2}{n}}$. From the figure we find the surprising fact that GTDM is much better than the other methods. It converges very fast and only needs several iterations. The behaviors of TDM in the four cases are different. In Case 2 and 3 TDM is easy to be trapped in the local minimum. In Case 4, since the data points are not nearly planar, TDM shows good performance. In all the cases GN is a little better than CDM but is still slower than GTDM and SDM.

From our experience in 2D/3D curves and surfaces fitting [12,13], actually there is no strong evidence and theoretical guarantee that shows which method (TDM, CDM, GTDM, GN, SDM) is best for most fitting problems since the integrated step-control strategy like line search or the Levenberg-Marquardt method affects the behavior and unexpected local minimum may stop the optimization. Also for large residual problems all the methods exhibit linear convergence which is similar to PDM. For instance, see Case 4 of the example. But in general GTDM is as good as the others at least in most cases. By considering the computational cost and overall performance, we strongly recommend GTDM for general parametric curve and surface fitting including parametric sub-manifold fitting (i.e, when $m < s - 1$) due to its clear geometric meaning and its simplicity since it does not need to compute the principle curvatures and directions.

5 Conclusions

A systematic geometrical and optimizational analysis on least squares orthogonal distance fitting of parametric curves and surfaces is presented in this paper. We give the geometric characterization of existing techniques and propose two modified versions based on geometric meanings. We show how principle curvature and directions are embedded in optimization methods. The presented geometric understanding of optimization techniques will benefit efficient and effective curve/surface fitting. Also for further research, it is interesting to study the geometry behind methods for implicit curve/surface fitting.

Acknowledgements

The work was partially supported by the National Key Basic Research Project of China under 2004CB318000.

References

1. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, Heidelberg (2006)
2. Hoschek, J.: Intrinsic parameterization for approximation. Computer Aided Geometric Design 5(1), 27–31 (1988)
3. J. Hoschek and D. Lasser. Fundamentals of Computer Aided Geometric Design. Peters, A. K. (1993)
4. Weiss, V., Andor, L., Renner, G., Várady, T.: Advanced surface fitting techniques. Computer Aided Geometric Design 19(1), 19–42 (2002)
5. Bjorck, A.: Numerical Methods for Least Squares Problems. In: Mathematics Society for Industrial and Applied Mathematics, Philadelphia (1996)
6. Helfrich, H.P., Zwick, D.: A trust region algorithm for parametric curve and surface fitting. Journal of Computational and Applied Mathematics 73(1-2), 119–134 (1996)
7. Speer, T., Kuppe, M., Hoschek, J.: Global reparameterization for curve approximation. Computer Aided Geometric Design 15(9), 869–877 (1998)
8. Golub, G., Pereyra, V.: Separable nonlinear least squares: the variable projection method and its applications. Inverse Problems 19, 1–26 (2003)
9. Borges, C.F., Pastva, T.: Total least squares fitting of Bézier and B-spline curves to ordered data. Computer Aided Geometric Design 19(4), 275–289 (2002)
10. Atieg, A., Watson, G.A.: A class of methods for fitting a curve or surface to data by minimizing the sum of squares of orthogonal distances. Journal of Computational and Applied Mathematics 158, 227–296 (2003)
11. Ahn, S.J.: Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space. LNCS, vol. 3151. Springer, Heidelberg (2004)
12. Wang, W., Pottmann, H., Liu, Y.: Fitting B-spline curves to point clouds by curvature-based squared distance minimization. ACM Transactions on Graphics 25, 214–238 (2006)
13. Liu, Y., Pottmann, H., Wang, W.: Constrained 3D shape reconstruction using a combination of surface fitting and registration. Computer Aided Design 38(6), 572–583 (2006)
14. Wallner, J.: Gliding spline motions and applications. Computer Aided Geometric Design 21(1), 3–21 (2004)
15. Piegl, L.A., Tiller, W.: The NURBS Book, 2nd edn. Springer, Heidelberg (1996)
16. Saux, E., Daniel, M.: An improved hoschek intrinsic parametrization. Computer Aided Geometric Design 20(8-9), 513–521 (2003)