Journal of Computational Design and Engineering, Vol. 1, No. 2 (2014) 116~127
www.jcde.org

Orthogonal projection of points in CAD/CAM applications: an overview

Kwanghee Ko[1,2*] and Takis Sakkalis[3]

[1] *School of Mechatronics, Gwangju Institute of Science and Technology, 123 Cheomdangwagiro, Bukgu, Gwangju, 500-712, Korea*
[2] *Korea Culture Technology Institute, Gwangju Institute of Science and Technology, 123 Cheomdangwagiro, Bukgu, Gwangju, 500-712, Korea*
[3] *Department of Science, Agricultural University of Athens, Athens, 118 55, Greece*

(Manuscript Received January 8, 2014; Revised February 24, 2014; Accepted February 25, 2014)

Abstract

This paper aims to review methods for computing orthogonal projection of points onto curves and surfaces, which are given in implicit or parametric form or as point clouds. Special emphasis is place on orthogonal projection onto conics along with reviews on orthogonal projection of points onto curves and surfaces in implicit and parametric form. Except for conics, computation methods are classified into two groups based on the core approaches: iterative and subdivision based. An extension of orthogonal projection of points to orthogonal projection of curves onto surfaces is briefly explored. Next, the discussion continues toward orthogonal projection of points onto point clouds, which spawns a different branch of algorithms in the context of orthogonal projection. The paper concludes with comments on guidance for an appropriate choice of methods for various applications.

Keywords: Orthogonal projection; Point projection; Curve projection; Registration; Minimum distance; Directed projection

1. Introduction

Orthogonal projection of a point is the process of finding a point on a curve or a surface such that the vector connecting the point in space and the point on the curve or the surface becomes perpendicular to the curve or the surface. It is one of the most critical operations in computer aided geometric design and applications, and efficient and robust computation of orthogonal projection is essential for various operations such as computation of closest point (foot-point) on a curve or a surface, parameter estimation of a point in space, intersection computation, and similarity.

Orthogonal projection is valid for pairs of a point and a curve, and a point and a surface. It is also extended to cover orthogonal projection of a curve onto a surface. Table 1 shows pairs of entities for which orthogonal projection can be considered. In this work, orthogonal projection of a point onto a curve or a surface is a primary operation.

Computation of a point on a curve or a surface that yields the minimum distance to a given point is an important application of orthogonal projection (Here, the case that orthogonal projection cannot be computed is excluded.) When a point is relatively close to a curve or a surface, the point on the curve or the surface that is the closest to the given point would be the orthogonal projection of the point onto the curve or the surface.

A typical example that requires such computation is localization. Localization, also denoted as registration, is the process of matching two objects as closely as possible. Consider a set of points and a surface in 3D space. Correspondence between the points and the surface are established by computing the points on the surface that yield the minimum distances between them. Such points are called the foot-points on the surface. Besl and McKay [1] proposed a registration algorithm, called the ICP algorithm, which is based on the computation of closest points for correspondence. The variants of the method have been presented in [2] -[5], to name a few.

They usually use additional information to improve convergence of iteration or to reduce the dependency of initial

Table 1. Classification for orthogonal projection depending on the geometric entities. Here, ‘x’ indicates that orthogonal projection is not defined. ‘t’ indicates that orthogonal projection can be defined, but may not be used in practice. ‘o’ means the case that orthogonal projection is valid.

	Point	Curve	Surface
Point	x	o	o
Curve	x	t	o
Surface	x	x	t

*Corresponding author. Tel.: +82-62-715-3225, Fax.: +82-62-715-2384
E-mail address: khko@gist.ac.kr

doi: 10.7315/JCDE.2014.012

positions of the points and the surface for convergence.

Parameter estimation is the process to estimate parametric values of given points for curve or surface approximation. This is an important step in reverse engineering or object reconstruction when a shape is defined by a set of points. Parametrization is a huge topic in CAD/CAM, Computer Graphics, etc. The complete coverage of this problem is beyond the scope of this paper. However, some of them are introduced here. Ma and Kruth [6] introduced the concept of base surface for parameter estimation of 3D points. A base surface is created using the 3D points, which is a crude approximation to the points. Next, the points are orthogonally projected onto the base surface, and the parameters of the projected points on the base surface are given as those of the 3D points. A similar approach was revisited by Piegl and Tiller [7]. They proposed a method of computing orthogonal projection efficiently.

Orthogonal projection can be used to compute the intersection of curves and surfaces. Limaien and Trochu [8] used the property of orthogonal projection that when two objects (curve or surface) intersect, successive orthogonal projection points from an initial point would converge to the intersection.

Escobar *et al*. [9] used orthogonal projection as one critical operation for aligning a surface triangulation with curves. Given a curve, the boundary edges of triangle meshes are aligned with the curve trajectory.

Flory and Hofer [10] used the orthogonal projection as one operation for designing a curve on a surface. They addressed the problem of fitting points on the surface with a curve, which is constrained to lie on the surface. Given a point cloud and an initial position of the fitting curve, the position of the curve is updated by minimizing a fitting error that is computed using the distance norm in the tangent space. The foot-points of the given points on the curve are computed through orthogonal projection, and the distances from the points to the foot-points are computed for use in the evaluation of the fitting error.

Orthogonal projection in a general situation was discussed by Zheng and Chen [11]. They addressed the problem of computing the shortest path between a point and a curve on a regular surface. Here, the shortest path from the given point to the curve becomes the geodesics between them. Moreover, it was found that the point on the curve giving the shortest path is the orthogonal projection point. The improvement of the proposed method over the existing one was realized by imposing the 'orthogonality' condition.

Orthogonal projection is used as a component for the fitting of geometric features to 2D or 3D point clouds. The fitting error is usually measured using the orthogonal distances from the given points to the geometric features. Minimizing such errors involves satisfaction of the 'orthogonality' condition, and the corresponding point on the geometric entity becomes the foot-point for the orthogonal projection. Although the orthogonal projection points may not be computed explicitly, this approach provides the basis for least squares fitting of points with various geometric entities. Ahn *et al*. [12] addressed the problem of fitting of circle, sphere, ellipse, hyperbola and parabola to given points in the least squares sense. An objective function, called the performance index, is derived, which provides the sum of squares of orthogonal distances. The optimum solution for this objective function is obtained using the Gauss-Newton-type iteration. This idea is also applied to implicit curves and surfaces in [13] and [14]. Further discussions on least squares fitting were made by numerous researchers such as [15] and [16], to name a few.

Mathematically, orthogonal point projection is not necessarily the same as the minimum distance computation [17]. There could be more than one orthogonal projection points, or no such points exist. On the other hand, the minimum distance can always be computed. The case that the orthogonal point projection in 3D space yields the foot-point for the minimum distance is when the point is closer to the interior of the curve or the surface than any other edge, boundary and corner points and when the curve or the surface is regular. When a set of points is given, the minimum distances from most of the points can be computed through the orthogonal point projection.

A lot of literature has been devoted to developing efficient and robust solution methods for computing orthogonal projection of a point onto a curve or a surface. In particular Newton-Raphson method, a local scheme to iteratively find orthogonal projection, has been frequently selected as a solution method. Although the method is fully analyzed mathematically and the advantages and disadvantages are recognized, no single method can satisfy robustness, efficiency and accuracy conditions simultaneously. In this paper, the problems of orthogonal projection are addressed for curves and surfaces. Mathematical definitions of orthogonal projection of points and curves are presented to provide a firm ground for thorough understanding of the problem. Next, various calculation methods are reviewed, which have been introduced so far in various disciplines. The methods are classified into two types: iteration based and subdivision based methods. The former computes orthogonal projection by iteratively searching for the solution algebraically and geometrically. The latter subdivides a curve or a surface into smaller ones and then selects those that contain the solution. For an accurate solution, an iteration-based method such as Newton-Raphson method can be employed. The discussion is extended to cover orthogonal projection of points onto point clouds. The paper concludes with comments on guidance for an appropriate choice of methods for various applications.

2. Mathematical definition of orthogonal projection

The term, *orthogonal projection*, has its origin in Euclidean geometry when one projects a point **P** onto (its foot-point **Q**) a plane $\mathbf{T}_P$ in 3D space. As the term *orthogonal* indicates, the vector **P**-**Q** is making an angle of 90° - right angle - with every vector lying in $\mathbf{T}_P$. Since every vector lying in $\mathbf{T}_P$ can be thought of as being tangent to $\mathbf{T}_P$, one can easily generalize this notion to an object that has tangent vectors, and thus to the field of differential geometry.

Definition 1. Let **S** be a non-empty subset of $\mathbf{R}^n$ so that for each point $\mathbf{a} \in \mathbf{S}$ there is a tangent vector $\mathbf{T}_a$ of **S** at **a**. For a point $\mathbf{P} \in \mathbf{R}^n$, the orthogonal projection of **P** onto **S** is the set of points $\mathbf{Q} \in \mathbf{S}$ so that the vector **P**-**Q** is perpendicular to $\mathbf{T}_Q$. □

In this work, we shall mostly deal with smooth (closed) curves and surfaces, unless otherwise stated.

2.1 Projection of point onto curve or surface

Let **c** be a (closed) parameterized curve in $\mathbf{R}^n$; that is $\mathbf{c}$: $t \in [0,1] \rightarrow \mathbf{R}^n$, which is smooth, except possibly at a finite number of points. For a point $\mathbf{P} \in \mathbf{R}^n$, the set of orthogonal projection consists of all points on **c** so that

$$\left(\mathbf{P} - \mathbf{c}(t)\right) \cdot \mathbf{c}'(t) = 0. \tag{1}$$

In the case where the orthogonal projection point is not a continuously varying point, but there is a tangent vector **T**, we can still define a set of orthogonal projection by invoking Definition 1. The definition can be directly applied when **S** is given parametrically.

Let now **s** be a (closed) smooth surface in $\mathbf{R}^n$, given by $\mathbf{s} = \{\mathbf{q} \mid F(\mathbf{q}) = 0, \mathbf{q} \in \mathbf{R}^n\}$, where F: $\mathbf{R}^n \rightarrow \mathbf{R}$ is a smooth function. Then, the set of orthogonal projection of $\mathbf{P} \in \mathbf{R}^n$ onto **s** is equal to

$$\Gamma_{orth} = \{\mathbf{Q} | (\mathbf{P} - \mathbf{Q}) // \nabla f(\mathbf{Q}), \mathbf{P} \in \mathbf{R}^n\}. \tag{2}$$

2.2 Projection of curve onto surface

Let us now concentrate on the 3D case, and consider $\mathbf{c} = (c_1, c_2, c_3) : [0,1] \rightarrow \mathbf{R}^3$ a parametric curve and **s** a smooth hypersurface in $\mathbf{R}^3$. Assume that **s** is given parametrically; that is, **s** is a regular function $\mathbf{s}$: $[0,1] \times [0,1] \rightarrow \mathbf{R}^3$, $\mathbf{s}(u,v) = (s_1, s_2, s_3)$. Then, $\mathbf{s}_u \times \mathbf{s}_v$ is a nonzero vector perpendicular to **s** at $\mathbf{s}(u,v)$. Therefore, the orthogonal projection of **c** onto **s** should be the set of points on **s** so that $(\mathbf{c}(t) - \mathbf{s}(u,v))\cdot\mathbf{s}_u = 0$ and $(\mathbf{c}(t) - \mathbf{s}(u,v))\cdot\mathbf{s}_v = 0$. We may derive similar equations for the case where **s** is defined implicitly.

2.3 Orthogonal projection vs. minimum distance

Orthogonal projection is related to the notions of minimum distance between a point and a set and distance projection of a point onto a set.

Definition 2. Let $\mathbf{P} \in \mathbf{R}^n$ and **S** a non-empty subset of $\mathbf{R}^n$. Then, the distance d(**P**,**S**) from **P** to **S** is defined as

$$d(\mathbf{P}, \mathbf{S}) = inf_{\mathbf{Q} \in \mathbf{S}} d(\mathbf{P}, \mathbf{Q}). \tag{3}$$

□

The distance projection of **P** onto **S** is defined as the set

$$\Gamma_{dist} = \{\mathbf{Q} | \mathbf{Q} \in cl(\mathbf{S})\ and\ d(\mathbf{P}, \mathbf{Q}) = d(\mathbf{P}, \mathbf{S})\}. \tag{4}$$

Here, *cl*(**S**) is the closure of **S**, to be the intersection of all closed subsets of $\mathbf{R}^n$ that contain **S**. It is true that if **C** and **S** are smooth and the distance projection $\Gamma_{dist} \subset Int(\mathbf{S})$, then $\Gamma_{dist} \subset \Gamma_{orth}$. Here, *Int*(**S**) is the interior of **S** defined by the union of all open sets contained in **S**. However, the converse is not true. Indeed, any critical point of the distance function $d(\mathbf{P},\mathbf{Q})$, $\mathbf{Q} \in \mathbf{S}$, is an orthogonal projection point, but not necessarily an infimum.

3. Computation methods of orthogonal projection

Orthogonal projection of a point onto a curve or a surface can be obtained by solving Eq. (1). Here, the curve and the surface are mathematically defined. The equation is, however, a nonlinear equation that cannot be easily solved analytically except for several special cases. Therefore, various numerical methods have been considered to find orthogonal projection based on Eq. (1).

Orthogonal projection is computed for general curves and surfaces. However, a group of researchers focused on conics and tried to develop methods for efficient computation of orthogonal projection onto conics.

3.1 Orthogonal projection onto conics

When conics (ellipses, hyperbolas, parabolas) are considered, the problem of orthogonal projection can be handled in a more mathematically rigorous manner. In general, this problem is addressed in the context of least squares fitting of quadratic geometric entities, and orthogonal projection is a critical operation in the fitting process. Chernov and Wijewickrema [18] addressed the problem of point projection onto quadratic curves. Their emphasis is placed on the robustness and practical aspects of the solution algorithms with good accuracy and simplicity. Although several approaches were discussed, which are relatively fast such as [12], [13] and [14], only a few methods are theoretically proven to be robust [16], [19] and [20]. These methods were analyzed, and a modified algorithm of [19] was presented in [18].

The problem of orthogonal projection of a point onto a conic is formulated as follows. A conic is defined by

$$U(x, y) = Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0. \tag{5}$$

Consider a point (x_p, y_p). Then orthogonal projection of the point onto the conic should satisfy the orthogonality conditions

$$x_p - x = t_p(Ax + By + D),$$
$$y_p - y = t_p(Bx + Cy + E), \quad (6)$$

for some t_p. Geometrically, these conditions are derived from the orthogonality property that the vector (x_p-x, y_p-y) is parallel to ∇U. Solving the equation system for x and y produces two expressions as functions of t_p, which are then substituted into the conic equation yielding

$$c_4 t_p^4 + c_3 t_p^3 + c_2 t_p^2 + c_1 t_p + c_0 = 0. \quad (7)$$

Here, the coefficients c_k are represented in terms of A, B, C, D, E and F. The solutions to the equation are the orthogonal projection points on the conic. However, finding out the solutions is not a simple task. Although there exists an analytical formula for the solution to the equation, it cannot be readily used in practice due to its complexity. Various numerical methods including local and global schemes can be employed to find the roots of the Eqs. [21]-[25]. However, in terms of performance and stability, they cannot be readily used in practice. In [16], an algorithm for orthogonal projection of a point onto a conic was proposed. With t_p eliminated, Eq. (6) can be rewritten as

$$R(x, y) = (x_p - x)(Bx + Cy + E)$$
$$-(y_p - y)(Ax + By + D) = 0. \quad (8)$$

At the orthogonal projection point, Eqs. (5) and (8) are satisfied. Using the matrix notation, we are able to rewrite Eqs. (5) and (8) as follows.

$$\mathbf{z}^T\mathbf{M}\mathbf{z} = 0, \ \mathbf{z}^T\mathbf{N}\mathbf{z} = 0, \quad (9)$$

where

$$\mathbf{M} = \begin{bmatrix} A & B & D \\ B & C & E \\ D & E & F \end{bmatrix}$$

$$\mathbf{N} = \begin{bmatrix} -2B & A - C & x_pB - y_pA - E \\ A - C & 2B & x_pC - y_pB + D \\ x_pB - y_pA - E & x_pC - y_pB + D & 2(x_pE - y_pD) \end{bmatrix}$$

and $\mathbf{z} = [x\ y\ 1]^T$.

When the given conic is a circle, orthogonal projection can be analytically computed by elementary geometry. Except for such a case, orthogonal projection points are the intersections of the two conics in Eq. (9). Intersection computation of Eq. (9) is obtained by considering a family of conics given by

$$\mathbf{z}^T(\alpha\mathbf{M} + \beta\mathbf{N})\mathbf{z} = 0, \quad (10)$$

where α and β are real numbers. With det($\mathbf{M}$) = 0 and β = 0, the given conic Eq. (5) is reduced to a single or a pair of lines. Then, orthogonal projection can be easily performed. When det $\mathbf{M} \neq 0$ and β = 1, then we have

$$\mathbf{z}^T\mathbf{E}\mathbf{z} = 0, \ \mathbf{E} = \alpha\mathbf{M} + \mathbf{N}. \quad (11)$$

Here, α satisfies the equation det($\mathbf{E}$) = 0. This means that the conic of Eq. (11) is degenerate and consists of two lines or a single line. Then, the foot-points of orthogonal projection of the given point are reduced to the intersection of the conic (5) with the lines. The point loser to the given point is chosen as the final orthogonal projection point.

Eberly's projection method focuses on finding the correct initial point so that Newton's iteration method can converge to correct solutions all the time [19]. The method was developed for orthogonal projection onto an ellipse, and later was extended to other conics and 3D quadratic surfaces [15]. The choice of such points starts with translating and rotating the coordinate system of the ellipse as given in Eq. (12) to obtain a representation in canonical coordinates.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0, \qquad a \geq b > 0. \quad (12)$$

Next, the orthogonality conditions are obtained similarly to Eq. (6), which are substituted in Eq. (12) to produce an equation of variable t_p.

$$P(t_p) = \frac{a^2x_p^2}{(t_p + a^2)^2} + \frac{b^2y_p^2}{(t_p + b^2)^2} - 1 = 0. \quad (13)$$

Then, the roots of Eq. (13) correspond to the orthogonal projection points. In order to solve Eq. (13), a starting point is chosen as

$$t_{pi} = max\{ax_p - a^2, by_p - b^2\}, \quad (14)$$

from which Newton's method can be employed to obtain the accurate root. An improved version of Eberly's method was proposed by [18]. They followed the similar process to Eberly's with modified coordinate transformation and proposed a process that can be applied to all conic types.

Orthogonal projection onto general curves and surfaces is more involved than orthogonal projection onto conics because their shapes are defined in a more complicated way. Therefore, analytical computation of orthogonal projection onto such curves and surfaces may not be possible in most cases. Instead, various numerical methods have been introduced. The methods can be classified into two types in terms of algorithmic approaches: iteration and subdivision. The former is to compute orthogonal projection by solving a governing equation such as Eq. (1) in an iterative manner. This method requires a starting value that is supposed to be a good approximation to the orthogonal projection point. Next, it iteratively finds the accurate solution. The latter is to find the root of the equation by subdividing the domain of interest into smaller ones. The subdivided regions that are assumed to contain roots are further subdivided. This recursive subdivision continues until the sizes of intervals in each axis of the subdivided region are less than the user-defined tolerance, or until the flatness of the subdivided region is satisfied.

3.2 Iteration-based

3.2.1 Algebraic root finding

Computing the orthogonal projection of a point is equivalent to finding the solution of Eq. (1). The solution can be obtained using algebraic or geometric methods, which essentially search for the orthogonal projection point in an iterative manner.

Newton-Raphson method is a typical choice for computing the root of Eq. (1). Because the derivatives of the equation are analytically computed, the method can find the roots of the equation without further geometric information. In the context of computing the minimum distance, Mortenson [26] presented equations for surface distance measures, which can be solved using Newton's iteration method. Hartmann [27] used the normalform of a curve for the computation of the foot-point on the curve of a point. Consider a smooth implicit curve $c(\mathbf{x})$ in $\mathbf{R}^2$ and $\mathbf{x} \in \mathbf{R}^n$. If there exists a continuously differentiable function h with $|\nabla h|=1$ on and in the vicinity of the curve, the equation $h = 0$ is called the normalform of $c(\mathbf{x})$. The value of the normalform is equivalent to the suitably oriented distance between a test point and its foot-point, and the gradient of h is the unit normal at the foot-point. The foot-point corresponds to the orthogonal projection point with the gradient of h, the unit normal vector. The computation algorithm is based on combining the foot-point on the tangent of the curve and the approximate foot-point on the tangent parabola. The accurate foot-point on the curve is computed iteratively based on those foot-points at each iteration step. Essentially, this approach is similar to the Newton type iteration method.

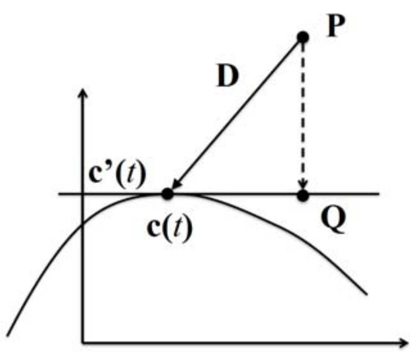

Figure 1. Illustration of the first order approximation of the curve and the orthogonal projection of **P** on the tangent line.

A good initial value is required to find a solution iteratively using a Newton's iteration method. This requirement is critical in this scheme because the success of finding a correct root heavily depends on the initial value. However, providing such an initial value is not an easy problem because of the complexity of the equation caused by the complicated shapes of a curve or a surface [28].

3.2.2 Geometric iteration

The orthogonal projection point can be obtained by iteratively searching for a point on a curve or a surface satisfying the orthogonality condition. Searching is performed based on geometric properties such as tangent vectors or planes and curvature. At each point, the governing equation of orthogonal projection such as Eq. (1) is tested to check if the point is close to the true orthogonal projection point.

Limaien and Trochu [8] proposed a method of computing the orthogonal projection of a point onto a parametric curve or a parametric surface. They used the concept of 'dual kriging' that had been formulated by Matheron [29]. With the dual kriging, the equations of smooth parametric curves from a set of points can be automatically constructed [29].

Kriging is a method of interpolation based on Gaussian process regression. It is considered as an estimator giving a local estimation value using a linear combination of data points. Values for interpolation are modeled by a Gaussian process with prior covariances, which leads to the best linear unbiased prediction. An alternative formulation to kriging is called dual kriging, which provides an estimated value using a linear combination of covariance functions. Smooth parametric curves and surfaces can be represented using dual kriging from a set of points [8]. For example, a kriged curve can be defined in parametric form consisting of a linear part for the average shape of the curve and the correction part to the average shape as follows.

$$c(t) = a_0 + a_1 t + \sum_{j=1}^{N} b_j \left|t - t_j\right|^3 \tag{15}$$

where t is the arc-length of the curve $c(t)$ from $t = t_0$, and a_0 and a_1 determine the average shape of the curve, and the summation with b_j adjusts the average shape. Here, the average shape is denoted as the *drift*, and the cubic term is denoted as the *generalized covariance*. These correction terms allow the curve to fit the given data points. The orthogonal projection for the curve case is computed as follows. Suppose that **P** be a test point, $\mathbf{Q}_i$ denotes the current point on a curve, and $\mathbf{T}_i$ is the tangent vector at $\mathbf{Q}_i$. At the orthogonal projection point, $(\mathbf{P} - \mathbf{Q}_i)\cdot\mathbf{T}_i = 0$ should be satisfied. Considering the sign of $(\mathbf{P} - \mathbf{Q}_i)\cdot\mathbf{T}_i$, the span of the curve with two consecutive points $\mathbf{Q}_i$ and $\mathbf{Q}_{i+1}$ is located, which contains the orthogonal projection of **P**. Next, the orthogonal projection of **P** is computed by the *inverse kriging* between the two points as follows. Intermediate points on the curve are obtained for parameters t_i. Here, the number of intermediate points for the inverse kriging is determined depending on the desired accuracy. Next, a function $t = f((\mathbf{P} - \mathbf{Q})\cdot\mathbf{T})$ is constructed by interpolating the parameters t_i using their scalar values $(\mathbf{P} - \mathbf{Q}_i)\cdot\mathbf{T}_i$. The parametric value of orthogonal projection on the curve of $\mathbf{P}$ is then obtained by computing the parameter t satisfying $t = f(0)$. For the surface case, isoparametric curves on the surface are considered, and the procedure for the orthogonal projection on the curve is iteratively applied.

Hoschek and Lasser [30] presented a method of iterative parameter adjustment for parametrization. They did not explicitly discuss the orthogonal point projection onto a curve or a surface. However, their approach can be directly applied to finding orthogonal projection. This process for orthogonal projection onto a curve is summarized as follows. Consider a point $\mathbf{P}$ and a parametric curve $\mathbf{c}(t)$. Then, the parameter t is iteratively adjusted to make the error vector $\mathbf{D} = \mathbf{c}(t) - \mathbf{P}$ perpendicular to $\mathbf{c}'(t)$. The amount of adjustment dt at each iteration is obtained from the distance between the projection of $\mathbf{P}$ onto the tangent line at $\mathbf{c}(t)$ and the point $\mathbf{c}(t)$. As the iteration continues, the amount of adjustment dt converges to zero, leading to the parametric value t for the orthogonal projection of $\mathbf{P}$ onto $\mathbf{c}(t)$. This scheme can be extended to the surface case. In this process the first derivative of the curve is employed to approximate the local shape of the curve as shown in Figure 1.

Two different versions of Hoschek and Lasser's method are also proposed. Using the Taylor expansion, the terms up to the first derivative of the expansion are taken for formulation of the parameter adjustment [31]. In [32], the function $\mathbf{D}\cdot\mathbf{D}$ is used to derive the amount of correction. Namely, Hoscheck and Lasser's method is based on the first derivative of a curve or a surface for the iteration.

On the other hand, Hu and Wallner [33] proposed a method based on the second derivative properties of a curve or a surface for the orthogonal point projection. The core idea of the point projection onto a curve is to approximate the local shape of the curve at a point using a curvature circle. Next, the given point is projected onto the circle to produce $\mathbf{Q}$ as shown in Figure 2. Using $\mathbf{Q}$ the amount of parametric value of t is estimated to obtain the orthogonal projection point on the curve. The illustration of this method is given Figure 2, where $\mathbf{C}$ is the center of the curvature circle. This process continues until the amount of adjustment is less than the user-defined tolerance. This scheme is directly applied to the problem of the orthogonal point projection onto a surface. The schematic illustration of the surface case is given in Figure 3, where $\mathbf{s}$ is the surface. A normal curvature in the direction $\mathbf{P}$ from $\mathbf{s}_0$ is computed. Then, a circle is defined using the normal curvature with the center of curvature $\mathbf{C}$ and the value of the normal curvature. Next, the point $\mathbf{P}$ is projected onto the circle to produce the approximate projection point $\mathbf{Q}$ as shown in the figure. Using the projected point $\mathbf{Q}$, the amounts of updates for the parameter u and v of the surface are estimated. This procedure is repeated until the desired accuracy is obtained. This method shows improved stability compared to and performance similar to Newton-Raphson method. However, the method cannot overcome the dependency of initial values for convergence to the right solution.

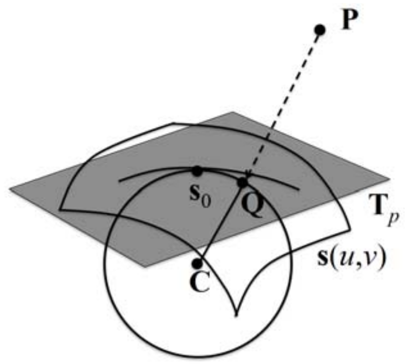

Figure 3. Hu and Wallner's second order method for surfaces

The extension of Hu and Wallner's method is proposed by Liu *et al.* [34]. Unlike Hu and Wallner's method their method can handle orthogonal point projection onto a surface only. The local shape of the surface is approximated by a torus patch as shown in Figure 4. The major and minor circles of the torus are constructed using the maximum and minimum

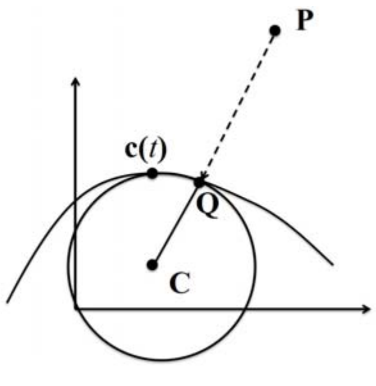

Figure 2. Hu and Wallner' second order method for curves.

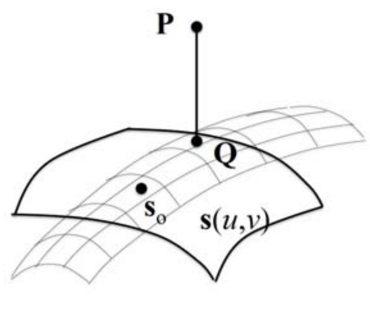

Figure 4. A torus patch is created to approximate the local shape of the surface near $\mathbf{s}_0$

principal curvatures. Next, the test point is projected onto the patch. Using the projected point **Q** on the patch, the parametric values of the original surface corresponding to the orthogonal projection on the torus patch are estimated. For this parameter estimation, the surface is Taylor expanded up to the second order. Then, a Newton-type iteration method is employed to find the parameter changes of the surface corresponding to the projection point on the torus patch. This method is demonstrated to show better performance than Hu and Wallner's method in terms of speed and stability. However, because it requires the Newton-type iteration method to find the parameter values on the surface, the stability of the method is not guaranteed.

Ko [35] presented a method of orthogonal projection of a point onto a 2D planar curve. He approximated the local geometric shape of a curve at a point using a quadratic polynomial based on the Taylor expansion of the curve at the point. Then the approximated orthogonal projection point is computed using the polynomial analytically, which is then provided as input to the next iteration. This process continues until the orthogonal projection point is obtained with a user-defined accuracy. In his method, the possibility of using the third derivatives of the curve is presented to improve the overall performance of the point projection.

Song *et al.* [36] used a biarc for approximating the local geometric structure of a parametric curve near the point of interest. The projection onto the biarc approximation is computed, giving the update for the parameter. This process is repeated until the termination condition is satisfied. Here, the concept of biarc approximation is of main interest. A biarc consists of two circular arcs connected at a common end point. The biarc is designed to approximate the local shape of the curve over a certain interval containing the projection point with the boundary positions and the first order derivatives.

The core idea of the geometric iteration approaches is to approximate the shape of a curve or a surface in the neighborhood of a point on the curve or the surface as closely as possible using a simple analytical expression, and to compute the orthogonal projection point on the approximate shape efficiently. From the approximated projection point, the amount of adjustment for the parameter(s) is computed to produce a new parametric value(s). This process is repeated until the computed orthogonal projection point converges to the accurate orthogonal projection point. This means that if the shape can be approximated as accurately as possible to the true shape, and the orthogonal point on the approximated shape can be obtained efficiently, then the convergence to the true orthogonal projection point can be accelerated, and the sensitivity to the initial point can be reduced.

So far, the first and the second order approximation approaches have been presented. Moreover, there has been an attempt to use the third order derivative properties for the local shape approximation. Considering general curves or surfaces except the line segments or the planes, the second order approximation methods have demonstrated better performance than the first order methods.

When a curve is defined implicitly, Aigner and Juttler [20] proposed a robust method for computing orthogonal projection, called the circle shrinking method. The core idea of this approach is as follows. Consider a point **P** and a curve $c(x,y) = 0$ that is implicitly defined. When **Q** is the orthogonal projection point, a circle can be defined, whose center and radius are **P** and $|\mathbf{P} - \mathbf{Q}|$, respectively. Because **Q** is the closest point from **P**, the circle intersects the curve only at **Q**, meaning that no other point on the curve is inside the circle, and the implicit function values are either zero or have the same sign as **P**. As shown in Figure 5, $\mathbf{Q}_i$ on the curve is provided as the initial point of iteration. Then, a circle passing $\mathbf{Q}_i$ is created with **P** as its center. The function $(\mathbf{P} - \mathbf{Q}_i)\cdot\nabla c$ is tested to check if $\mathbf{Q}_i$ is the orthogonal projection point or not. If not, a point $\mathbf{Q}^+$ in the neighborhood of $\mathbf{Q}_i$ is selected on the circle. $\mathbf{Q}^+$ can be the first local maximum of $c(x,y)$ along the arc from $\mathbf{Q}_i$. Next, a line segment connecting **P** and $\mathbf{Q}^+$ is obtained. The intersection between the line and the curve is calculated, which is denoted as $\mathbf{Q}_{i+1}$. This process is repeated until the radii of the successive circles do not change by more than a certain tolerance.

The geometric iteration methods have a close relation with geometric modeling and processing. Therefore, they can be naturally embedded in the whole application pipeline. More research on solving the two issues will be performed in the near future.

3.3 Subdivision based method

3.3.1 Subdivision of curve or surface

Piegl and Tiller [7] discussed the point projection problem in the context of parametrization for surface fitting from a set of random points. They used a base surface for estimating parameter values of given points by projecting them onto the base surface and taking the parameter values of the projected points. They claimed that the standard way of projection requires a good initial guess, which is expensive and error prone. They proposed a method to do away with Newton-Raphson method to find the projection points on the surface. The base surface is subdivided into quadrilaterals, and each point is projected onto the closest quadrilateral. Then, the parameter value for the point is computed from the parameter values of the corner points of the closest quadrilateral. The dependency of the initial guess no longer exists. However, the accuracy of the estimated parametrization is dependent on the quality of decomposition. The decomposition should be performed in such a way that the subdivided quadrilaterals are uniform in size and are created based on the geometry and the curvature of the base surface. Ma and Hewitt [37] proposed a method for computing point projection and inversion. Their concept is somewhat similar to that of Piegl and Tiller's. But the difference lies in the subdivision strategy that a curve or a surface is subdivided into Bezier curves or Bezier surfaces.

Then, the control polygons are used to find the candidates for orthogonal projection. The candidates are recursively subdivided into sub-curves until the control polygons of the sub-curves become “flat” or reach a limit for recursion. Here, the control polygons being “flat” means that the control points of a curve are nearly collinear or the control polygon of a surface is nearly planar. Then, the approximated projection points are obtained by projecting the points on the straight line or the plane. In order to improve accuracy, Newton-Raphson method can be employed. Ma and Hewitt showed that their method outperforms Piegl and Tiller’s method in terms of computation time and stability. However, the subdivision of a curve or a surface is still an expensive operation. The Ma and Hewitt’s algorithm for computing the nearest point on a Bezier curve in $\mathbf{R}^3$ space to the test point, however, produces incorrect results as demonstrated by Chen et al. [38]. They provided an example that Ma and Hewitt’s algorithm fails to find the correct answer. This defect may contribute to the performance degradation. An improved version of Ma and Hewitt’s method is proposed by Selimovic [39]. The new method subdivides the curve or the surface recursively as Ma and Hewitt’s approach. However, the computational performance could be improved by introducing new elimination criteria, with which a large number of subdivided parts are excluded in the computation. Endpoint interpolation, the convex hull property and tangent cones are used to develop the criteria for curves and surfaces. They showed that their approach could reduce the number of subdivided parts for testing of projection substantially compared with Ma and Hewitt’s method. The performance improvement is more pronounced for surfaces than for curves, which is demonstrated in their paper. Chen et al. [40] proposed the circular clipping algorithm, which is more efficient than those in [37] and [39]. They used a circle for a planar curve or a sphere for a 3D curve as an elimination region. The curve segments outside the circle or a sphere are eliminated after subdivision. This process is repeated until the termination condition is satisfied. Oh et al. [41] employed the concept of the circle/sphere clipping method by [40] but introduced a more efficient technique using the separating axis and k-DOP type of bounding scheme for culling or clipping unnecessary part of curves or surfaces during the iteration. Moreover, through testing the uniqueness of the projection point in the subdivided regions the computational efficiency can be improved. The method by Oh et al. [41] was extended to compute projection of a moving query point onto a freeform curve [42].

3.3.2 Subdivision based root finding

When Eq. (1) is formulated as a polynomial equation, then the root of the equation can be obtained using a subdivision-based method. Zhou et al. [43] presented a method to compute the stationary points of a distance function. The subset of such points corresponds to the orthogonal projection points on a curve or a surface. The equation is represented in Bernstein form, which is then given as input to the Projected Polyhedron (PP) algorithm [44] that finds the solution of the equation by narrowing down the domain containing roots through subdivision. This approach is not restricted to the problems of point projection or point inversion. This approach is robust in that it always finds the projection point as long as it exists. However, this approach is complicated in terms of implementation and is so expensive that it may not be used for the applications that require processing a large number of points.

These methods are robust because they do not use any initial values, and the whole domain of interest is considered in the computation of orthogonal projection points. However, they show a drawback in terms of performance because they require subdivision of a curve or a surface, which is an expensive operation. The subdivision methods are not used to compute the accurate orthogonal projection points in general because of the long computation time. Instead, they are usually employed to find the initial points, which are then used as input to various numerical computation methods for accurate orthogonal projection points such as Newton-Raphson method. Therefore, these methods must be used after considering the trade-off between the robustness and performance.

3.4 Extension of orthogonal projection of points

Orthogonal point projection onto a surface is extended to the orthogonal projection of a curve onto a surface. Here, the curve is given in 3D, and its projection on the surface is also a curve in 3D, but lies on the surface. Therefore, it can be considered as a method of curve design on the surface.

The problem of orthogonal projection of a curve onto a surface was addressed by Pegna and Wolter [17]. They derived a set of differential equation for tracing the projected curve on a surface. For this development, a few assumptions are necessary. First, the surface should be regular and second order continuous. Next, the curve itself should be positioned close enough to the surface, and the projected curve should lie in the interior of the surface. These assumptions are somewhat strict. However, the notion of orthogonal projection of a curve onto a surface is more than point projection in that the governing equation is given by a set of differential equations. Consider a parametric surface $\mathbf{s}(u,v)$ $(0 \le u, v \le 1)$ and a space curve $\mathbf{c}(t)$ with t as a parameter $(0 \le t \le 1)$. Then the projected curve on $\mathbf{s}$, denoted by $\boldsymbol{\gamma}(t)$ is given by $\gamma(t) = \mathbf{s}(u(t),v(t))$ satisfying

$$\begin{aligned}(\boldsymbol{\gamma}(t) - \mathbf{c}(t)) \cdot \frac{\partial \mathbf{s}}{\partial u} &= 0,\\ (\boldsymbol{\gamma}(t) - \mathbf{c}(t)) \cdot \frac{\partial \mathbf{s}}{\partial v} &= 0.\end{aligned} \tag{16}$$

After taking the derivative of Eq. (16) with respect to t, and using the chain rule, we obtain

$$\begin{bmatrix} \frac{du}{dt} \\ \frac{dv}{dt} \end{bmatrix} \mathbf{K}^{-1} \begin{bmatrix} \frac{d\mathbf{c}}{dt} \cdot \frac{\partial \mathbf{s}}{\partial u} \\ \frac{d\mathbf{c}}{dt} \cdot \frac{\partial \mathbf{s}}{\partial v} \end{bmatrix}, \tag{17}$$

where

$$\mathbf{K} = \begin{bmatrix} E + \rho L & F + \rho M \\ F + \rho M & G + \rho N \end{bmatrix}, \rho = |\boldsymbol{\gamma}(t) - \mathbf{c}(t)|. \tag{18}$$

E, F, and G are the first fundamental form coefficients, and L, M, and N are the second fundamental form coefficients of **s**, respectively. Given the parametric values of u and v of an initial point on the surface, the system Eq. (17) is numerically solved to trace the u and v values corresponding to the projected curve on the surface. Here, Runge-Kutta method can be employed. The initial point is obtained by computing the orthogonal projection point onto the surface. Ko [45] extended Eq. (17) to cover the orthogonal projection of lines of curvature and geodesic curves onto a surface. Song *et al.* [46] proposed a second order tracing method for orthogonal projection of a curve onto a parametric surface. The successive tracing points from the starting point are computed using the second order tracing formula defined in the parametric domain of the surface. The computed points are then approximated in a polyline based on the Hausdorff measure in the parametric domain, which is then mapped on the surface to produce the orthogonal projection curve on the surface. Although the related theory is complicated compared to the first order method, this second order method is claimed to be faster than the first order method. Wang et al. in [47] and [48] proposed algorithms for G^1 and G^2 continuous curve construction on a free-form surface using normal projection, respectively. The condition of normal projection in [17] was employed with different formulation of a set of differential equations for the normal projection curves. Xu et al. [49] proposed a method of orthogonal projection of a curve on a parametric surface using a second order approximation scheme. The central difference of others is that the projected curve on the surface is parameterized with the parameter of the curve. A set of differential equations is derived based on second order Taylor approximation, which is solved numerically with error adjustment.

4. Orthogonal projection of point onto point sets

Popularity of 3D scanning devices is growing, and point clouds are widely used for model representation. Therefore, more and more shape and rendering processes are developed to handle point clouds for various operations such as object fitting.

Orthogonal projection of a point onto point clouds requires a different approach from that of a point onto a curve or a surface because of uncertainty of accurate shape representation of the point clouds. Alexa and Adamson [50] analyzed Levin's MLS method [51] for computing normals from surfaces by point sets. The method based on Levin's approach would not result in orthogonal projection because the normal to the estimated tangent frame is not the actual surface normal. They provided an enhanced algorithm for computing orthogonal projections using the revised normal computation scheme. They assumed that the points implicitly define a smooth surface and introduced an implicit function f as follows.

$$f(\mathbf{x}) = \mathbf{n}(\mathbf{x})^T(\mathbf{x} - \mathbf{a}(\mathbf{x})) \tag{19}$$

The approximating surface is represented by the points **x** satisfying $f(\mathbf{x}) = 0$. Here, $\mathbf{n}(\mathbf{x})$ and $\mathbf{a}(\mathbf{x})$ are the weighted averages of normal and points at a location **x**. Orthogonal projection of **x** can then be estimated as follows [50] and [52]:

1) Set $\mathbf{x}'=\mathbf{x}$
2) Compute $\mathbf{a}(\mathbf{x}')$, $\mathbf{n}(\mathbf{x}')$, and $\nabla f(\mathbf{x}')$.
3) $\mathbf{x}' = \mathbf{x} - g\nabla f(\mathbf{x}')$,
   $g = (\mathbf{n}(\mathbf{x}')^T(\mathbf{a}(\mathbf{x}')-\mathbf{x}))/(\mathbf{n}(\mathbf{x}')^T\nabla f(\mathbf{x}'))$.
4) If $\|(\mathbf{n}(\mathbf{x}')^T(\mathbf{a}(\mathbf{x}')-\mathbf{x}'))\| > \mathcal{E}$, go back to step 2.

The point $\mathbf{x}'$ is repeatedly adjusted such that the vector ($\mathbf{x}'-\mathbf{x}$) becomes in the direction of $\nabla f(\mathbf{x}')$. Using the implicit representation of the surface from the point sets, exact normals can be estimated, from which accurate orthogonal projection can be computed.

Orthogonal projection of a point onto a set of points is addressed in a more general way. Namely, given a cloud of points $\mathbf{C}_n$, an arbitrary point **P** and the projection direction $\mathbf{n}_p$, the projection in direction $\mathbf{n}_p$, called *the directed projection*, is the point on $\mathbf{C}_n$ of **P** on the straight line $\mathbf{Q}^*(t) = \mathbf{P} + t\mathbf{n}_p$, where t is a parameter. This is intuitively equivalent to finding the intersection between $\mathbf{Q}^*(t)$ and $\mathbf{C}_n$. Azariadis and Sapidis [53] used the directed projection as its main component for solving the problem of generating smooth curves on the surface defined by point clouds. They proposed an iterative algorithm for finding the directed projection point. A weight function was proposed in a heuristic manner using the distance and the axis defined by **P** and **n**. With the weight, $\mathbf{Q}^*$ is obtained as a root of the following minimization problem:

$$Minimize\ E(\mathbf{Q}^*) = \sum_{i=0}^{N-1} a_i \|\mathbf{Q}^* - \mathbf{P}\|^2$$

These steps are repeated until the iteration converges to the projection point. This problem can be extended for computing orthogonal projection. Liu et al. [54] discussed a method of handling the case that the projection direction is not given. They proposed a method for estimating the projection direction, and an iterative algorithm to compute the projection of a 3D point onto a set of points. It turns out that their method inherently computes the orthogonal projection or normal projection of the given point onto the point clouds. Du and Liu [55] addressed the problem of lack of robustness of the method by [53] with respect to outliers and provided a way to overcome the issue using the concept of least median of squares. Zhang and Ge [56] proposed an improved MLS algorithm for the directed projection problem. Unlike MLS projection approach that the solution always lies along the normal direction, their proposed method computes directed projection by searching the solution along the projection direction. Moreover, they avoided the problem of incorrect choice of points for evaluating the weighting values. The existing MLS approach selects the neighboring points based on the shortest distance from the given point, which is prone to error when the geometric shape of the surface is complicated. For the criterion, the shortest distance from the projection vector was used for improved convergence.

5. Conclusions

Orthogonal projection is an important process in geometric modeling, computer aided design and computer graphics. The problem itself can be formulated in a mathematically rigorous manner. However, the actual solution method to the problem still experiences the stability and performance issues because orthogonal projection is usually given as roots of nonlinear equations, and solving them is usually an unstable and time consuming operation. Therefore, various applications from registration, curve and surface fitting, and similarity assessment that require accurate orthogonal projection are inevitably suffering from robustness and performance degradation.

In this review, a mathematical discussion on orthogonal projection is provided, and methods of computing orthogonal projection are reviewed. The solution methods are mainly classified into two groups: iteration-based and subdivision based methods. Extension to curve projection and projection onto a point cloud is explored. Various solution methods belonging to each category are presented and compared.

Mathematically, orthogonal projection requires that there should exist a tangent vector at the projected point on a curve or a surface. This means that the curve or the surface should be regular, and a tangent vector or a tangent plane should exist. When there are points on a surface or a curve where such a condition is not satisfied, orthogonal projection cannot be defined mathematically, which is a degenerate case. If computing orthogonal projection is a primary goal, the degenerate case needs to be handled with care to avoid a crash case of the computation method.

When the first derivative vanishes at a point on a curve or a surface, the curve or the surface is no longer *regular*, and no tangent line or tangent plane is considered at that point. Such a point is called the singular point, where differential geometry cannot be considered, and no unique normal vector can be obtained. Therefore, the mathematical definition of orthogonal projection is no longer applied in this case. However, such a degenerate case should be "nicely" handled for robustness of orthogonal projection computation. Computationally, the degenerate cases can be detected by checking the first derivative value during computation. If the derivative becomes less than a user-defined tolerance, the computation method stops, the type of singularity is analyzed, and a routine to handle such a type is called for to take care of the situation. In this paper, we focus on a smooth curve or a smooth surface, which are regular. When there is a singular point, the curve or the surface is subdivided into regions, each of which becomes regular. Next, orthogonal projection is performed. If computing the minimum distance is a main task, such a degenerate case can be handled using a minimization routine.

The recommended choice for orthogonal projection can be made depending on the representation methods of a curve or a surface. If a curve is given in implicit form, the method by [20] can be used because of its robustness in computation. For a curve and a surface in parametric form, two aspects should be taken into consideration: robustness and computation time. In order to guarantee the robustness of the orthogonal projection computation, a subdivision based global scheme should be employed, followed by Newton type iteration to improve accuracy. Under the condition that points are located closely enough near the target curve or surface, a simple Newton type iteration method is the best choice because of its speed and simplicity. However, satisfying such a condition is quite arbitrary. Therefore, either of the methods by [33] or [34] can be employed. Here, [33] is simpler than [34] in terms of implementation but is inferior to [34] in terms of robustness.

It is concluded that there is no perfect algorithm to solve the problem of orthogonal projection that satisfies high accuracy, robustness and efficient computation time simultaneously. Instead, much of the literature has focused on robustness of the solution methods although a certain amount of time loss is expected, which is a reasonable direction of research. Methods with increased robustness in computation of orthogonal projection will benefit in the actual applications more than unstable methods with less computation time, because the computation time can be handled by optimized hardware and software. Because the robustness of the solution process is the most critical component in the computation of orthogonal projection and the most influential to the other applications, it is recommended that more research efforts should be directed to improving the robustness of orthogonal projection computation.

Acknowledgments

The research is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0010099).

References

[1] Besl PJ, McKay ND. A method for registration of 3-D shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1992; 14(2):239-256.

[2] Sun YW, Xu JT. An optimal matching algorithm based on rough localization and exact adjustment, Key Engineering Materials. 2005; 291-292:661-666.

[3] Zhang ZY. Iterative point matching for registration of freeform curves and surfaces. International Journal of Computer Vision. 1994; 13(2):119-152.

[4] Sharp GC, Lee SW, Wehe DK. ICP registration using invariant features. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002; 24(1):90-102.

[5] Song Y, Vergeest SM, Matching geometric shapes in 4D space incorporating curvature information. In: ASME 2007 International Design Engineering Technical conferences and Computers and Information in Engineering Conference, 33rd Design Automation Conference, Parts A and B; 2007 Sep 4-7; Las Vegas, Nevada, USA; p. 515-523.

[6] Ma W, Kruth JP. Parametrization of randomly measured points for least squares fitting of B-spline curves and surfaces. Computer-Aided Design. 1995; 27(9):663-675.

[7] Piegl LA, Tiller W. Parametrization for surface fitting in reverse engineering. Computer-Aided Design. 2001; 33:593-603.

[8] Limaiem A, Trochu F. Geometric algorithms for the intersection of curves and surfaces. Computers and Graphics. 1995; 19(3):391-403.

[9] Escobar JM, Montenegro R, Rodriguez E and Montero G. Simultaneous aligning and smoothing of surface triangulations. Engineering with Computers. 2011; 27(1):17-29.

[10] Flory S, Hofer M, Constrained curve fitting on manifolds. Computer-Aided Design. 2008; 40(1):25-34.

[11] Zheng ZT and Chen SG. The general orthogonal projection on a regular surface. In: Proceedings of the 5th European Computing Conference (ECC'11); 2011 Apr 28-30; Paris, France; p. 116-119.

[12] Ahn SJ, Rauh W, Warnecke HJ. Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. Pattern Recognition. 2001; 34:2283-2303.

[13] Ahn SJ, Rauh W, Recknagel M. Least squares orthogonal distance fitting of implicit curves and surfaces. LNCS. 2001; 2191:398-405.

[14] Ahn SJ, Rauh W, Cho HS, Warnecke HJ. Orthogonal distance fitting of implicit curves and surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002; 42(5):620-638.

[15] Chernov N, Ma H. Least squares fitting of quadratic curves and surfaces. Nova Science Publisher: Computer Vision; 2011.

[16] Wijewickrema S, Esson C, Paplinski A. Orthogonal distance least squares fitting: a novel approach. Computer Vision, Image and Computer Graphics. Theory and Applications. 2010; 68:222-268.

[17] Pegna J, Wolter FE. Surface curve design by orthogonal projection of space curves onto free-form surfaces. Journal of Mechanical Design, ASME Transaction. 1996; 118(1):45-52.

[18] Chernov N, Wijewickrema S. Algorithms for projecting points onto conics. Journal of Computational and Applied Mathematics. 2013; 251:8-21.

[19] Eberly DH. 3D game engine design: a practical approach to real-time computer graphics. 2nd ed. San Francisco (CA): Morgan Kaufmann Publishers; 2007.

[20] Aigner M, Juttler B. Robust computation of foot points on implicitly defined curves. In: Daehlen M, Morken K, Schumaker L, editors, Mathematical methods for curves and surfaces. Brentwood (LA): Nashboro Press; 2005. p. 1-10.

[21] Christianson B. Solving quartics using palindromes. Mathematical Gazette. 1991; 75:327-328.

[22] Mochimaru Y. Reciprocal solution of a quartic equation. International Journal of Pure and Applied Mathematics. 2004; 14(2):207-210.

[23] Neumark S. Solution of cubic and quartic equations. Oxford: Pergamon Press; 1965.

[24] Yacoub MD, Fraidenraich G. A new simple solution of the general quartic equation. Mathematical Gazette; 2011.

[25] Shmakov SL. A universal method of solving quartic equation. International Journal of Pure and Applied Mathematics. 2011; 71(2):251-259.

[26] Mortenson ME. Geometric modeling. New York (NY): Wiley; 1985.

[27] Hartmann E. On the curvature of curves and surfaces defined by normalforms. Computer Aided Geometric Design. 1999; 16:355-376.

[28] Piegl LA, Tiller W. The NURBS book. 2nd ed. New York (NY): Springer-Verlag; 1995.

[29] Matheron G. Splines et Krigeage: leur equivalence formelle, Rapport N-667. Center de Geostatistique, Fontainebleau: Ecole des Mines de Paris; 1980.

[30] Hoschek J, Lasser D, Fundamentals of computer aided geometric design. Natick (MA): A K Peters; 1993.

[31] Rogers DF, Fog NG. Constrained B-spline curve and surface fitting. Computer-Aided Design. 1989; 21:641-648.

[32] Hoschek J, Schneider FJ, Wassum P. Optimal approximate conversion of spline surfaces. Computer Aided Geometric Design. 1989; 6:293-306.

[33] Hu SM, Wallner J. A second order algorithm for orthogonal projection onto curves and surfaces. Computer Aided Geometric Design. 2005; 22(3):251-260.

[34] Liu XM, Yang L, Yong JH, Gu HJ, Sun JG. A torus patch approximation approach for point projection on surfaces. Computer Aided Geometric Design. 2009; 26:593-598.

[35] Ko KH. Analysis on point projection onto curves. Transactions of the Society of CAD/CAM Engineers. 2013; 18(1):49-57. In Korean.

[36] Song HC, Xu X, Shi KL, Yong JH. Projecting points onto planar parametric curves by local biarc approximation. Computers & Graphics. 2014; 38:183-190.

[37] Ma YL, Hewitt WT, Point inversion and projection for NURBS curve and surface: control polygon approach. Com-

puter Aided Geometric Design. 2003; 20:79-99.

[38] Chen XD, Su H, Yong JH, Paul JC, Sun JG. A counterexample on point inversion and projection for NURBS curve. Computer Aided Geometric Design. 2007; 24(5): 302.

[39] Selimovic I. Improved algorithms for the projection of points on NURBS curves and surfaces. Computer Aided Geometric Design. 2006; 23:439-445.

[40] Chen XD, Yong JH, Wang G, Paul JC, Xu G. Computing the minimum distance between a point and a NURBS curve. Computer-Aided Design. 2008; 40:1051-1054.

[41] Oh YT, Kim YJ, Lee J, Kim MS, Elber G. Efficient point-projection to freeform curves and surfaces. Computer Aided Geometric Design. 2012; 29:242-254.

[42] Oh YT, Kim YJ, Lee J, Kim MS, Elber G. Continuous point projection to planar freeform curves using spiral curves. The Visual Computer. 2012; 28:111-123.

[43] Zhou J, Sherbrooke EC, Patrikalakis NM. Computation of stationary points of distance functions. Engineering with Computers. 1993; 9:231-246.

[44] Sherbrooke EC, Patrikalakis, NM. Computation of the solutions of nonlinear polynomial systems. Computer Aided Geometric Design. 1993; 10(5):379-405.

[45] Ko K. Algorithms for three-dimensional free-form object matching. PhD Thesis. Cambridge (MA): Massachusetts Institute of Technology; 2003.

[46] Song HC, Yong JH, Yang YJ and Liu XM. Algorithm for orthogonal projection of parametric curves onto B-spline surfaces. Computer-Aided Design. 2011; 43:381-393.

[47] Wang X, Zhang W, Zhou L, Zhang L. Constructing G1 continuous curve on a free-form surface with normal projection. International Journal of Computer Mathematics. 2010; 87(10):2291-2302.

[48] Wang X, An L, Zhou L, Zhang L. Constructing G2 continuous curve on freeform surface with normal projection. Chinese Journal of Aeronautics. 2010; 23:137-144.

[49] Xu HY, Fang X, Tam HY, Wu X, Hu L. A second-order algorithm for curve orthogonal projection onto parametric surface. International Journal of Computer Mathematics. 2012; 89(1):98-111.

[50] Alex M, Adamson A. On normals and projection operators for surfaces defined by point sets. In: Eurographics Symposium on Point-Based Graphics; 2004; ETH Zurich, Switzerland.

[51] Levin D. Mesh-independent surface interpolation. Berlin: Springer; 2003. p. 37-49.

[52] Kobbelt L, Botsch M. A survey of point-based techniques in computer graphics. Computers & Graphics. 2004; 28(6):801-814.

[53] Azariadis PN, Sapidis NS. Drawing curves onto a cloud of points for point-based modeling. Computer-Aided Design. 2005; 37:109-122.

[54] Liu YS, Paul JC, Yong JH, Yu PQ, Zhang H, Sun JG, Ramani K. Automatic least-squares projection of points onto point cloudes with applications in reverse engineering. Computer-Aided Design. 2006; 38:1251-1263.

[55] Du MC, Liu YS. An extension on robust directed projection of points onto point clouds. Computer-Aided Design. 2008; 40:537-553.

[56] Zhang Y, Ge L. Improved moving least squares algorithm for directed projecting onto point clouds. Measurement. 2011; 44:2008-2019.