Parts Localization Oriented Practical Method for Point Projection on Model Surfaces Based on Subdivision

Jinting Xu[1], Jianhuang Wu[2] and Wenbin Hou[1]

[1]Dalian University of Technology, xujt@dlut.edu.cn (J.T. Xu), houwb@dlut.edu.cn (W.B. Hou)
[2]Shenzhen Institutes of Advanced Technology, jh.wu@siat.ac.cn

ABSTRACT

In algorithm of part localization, the distances between the measured points and their projections on the model surface is evaluated as the machining allowance or machining error of the parts with free-form surfaces. Point projection of a measured point on the model surface is the key to process the localization of the measured data points to the CAD model. At present, iteration-type methods are the most popular solving strategy. However, it cannot provide full assurance that all roots have been found. This paper presents a subdivision-based method for computing point projection on the model surface. Different from the previous methods, our method is to use the position relationships of the graph of the first derivative of the squared distance function and the u-v parametric plane to eliminate the invalid surface segments. A simple formula is derived based on AMBP (arithmetic of multivariate Bernstein-form polynomial) to facilitate the use of this criterion. Compared with the iteration-type methods, one advantage of our method is that it can avoid providing any initial value for achieving the proper result and guarantee the roots for all conditions. Finally, some examples are given to demonstrate the effectiveness and robustness of the proposed method.

Keywords: parts localization, point projection, model surfaces, subdivision.

1. INTRODUCTION

Point projection is an essential tool in quality evaluation of the machined parts with free-form surface and localization for machining allowance optimization for evaluating the error between the machined parts or blank parts and CAD models [11,20,21]. Much of the research work on point projection is concentrated on Newton iteration and its variations because of their quadratic convergence rate close to the simple root [7,8,12,23]. The present work is to compute the point projection on the model surface by using the subdivision based strategy rather than iteration processing. Out of the various applications of point projection, localization for machining allowance optimization and inspection of machined parts based on nominal CAD model are the main motivation for the present work. In such applications, the focus of interest is the correctness of point projection rather than the computation speed of the algorithm itself.

Although the predominant iteration-type methods exhibits quadratic convergence rate that is very attractive for many applications, it is still a error prone process that fails quite often especially for the boundary points [17], and some tests by Ma and Hewitt [14] have indicated that widely used Newton-Raphson method gives occasionally some wrong results even with a quite good initial iteration point when applying it on the whole surface. Perhaps, occasional errors is trivial for the graphics processing, however, it is fatal for the practical industrial applications. For example, in inspection of the machined part, if the error on a point shows bigger than that it should be because of miscalculation of its closest point, the qualified parts may be considered to be defective ones that need to be returned back to the factory and is reworked [10]. Also, for machining allowance optimization, the distance between a point and its projection on the model surface is evaluated as the machining allowance at this point, if wrong projection is calculated, the blank parts may be reworked due to the material shortage of some areas resulted from the wrong projection, even although the nominal model can be actually enclosed within the blank parts [22]. For these industrial applications, the robustness of calculation of the closest point may be found to be more important and economical than the savings of the computing time.

The lack of robustness promotes the development of effective and reliable techniques. A different solving strategy, the methods based on subdivision, has been used for computing the closest point or providing a good initial value for Newton-type methods. For example, Piegl and Tiller [17] decomposed the model surface into quadrilaterals, projected the test point onto the closest quadrilateral, and then recovered the parameters of the closest point from the quadrilateral. Ma and Hewitt [14] eliminated the invalid surface segments by checking the relationship of the test point and the control points of Bézier surface, and then subdivided the valid surface segments, approaching to the closest point or providing a good initial value for Newton-Raphson method. However, their elimination criterion might fail in case of some 3D curves that have been given by Chen et al. [3,4]. Zhou et al. [25] converted the point projection problem into a polynomial equation system and the solutions were obtained by PP (Projected-Polyhedron) algorithm. Dyllong and Luther [6] gave a diffident exclusion criterion, however in some cases their method might also lead to wrong results. Selimoic [19] improved the subdivision-based method by a stricter elimination criterion which can increases considerably the robustness of the algorithm.

When subdividing the model surface, it is necessary to determine which segments contain possibly the projection. In other words, the segments not including the solutions must be eliminated as much as possible; it is also the crux of the subdivision-based methods. The criterions of Ma [14] and Selimoic [19] are both to utilize the relationship of the test point and the control polygons of Bézier surfaces to eliminate the invalid surface segments. Different from theirs, the elimination criterion proposed in the paper is to utilize the position relationships of the graph of the first derivative of the squared distance function and the u-v parametric plane to eliminate those invalid surface segments. A simple formula is derived using AMBP to facilitate the use of this new criterion. Together with the variation diminishing property of the Bernstein-form polynomials, it can eliminate effectively the segments not including the solutions. With a slight modification, the proposed method can be applied well to NURBS surfaces. For the sake of simplicity, its basic principles are explained by using B-spline surface in the following.

2. OUTLINE OF THE PROPOSED METHOD

In the proposed method, a B-spline surface is first decomposed into a set of Bézier surfaces. For each Bézier patches, the first derivative surface of the squared distance function is modeled by using AMBP, whose relative position to the u-v parametric plane is then used to determine whether the derivative surface is tangential to the u-v parametric plane or not and the tangential point is the closest point. The candidate surface patch will be iteratively subdivided, approaching to the tangential point, until a given tolerance is reached, as shown in Fig. 1. The proposed method is summarized as following and the detailed steps are discussed in the subsequent sections.

- Subdividing a B-spline surface into a set of Bézier surfaces;
- Creating the first derivative surface of the squared distance function;
- Narrowing down the surface including possibly the solutions;
- Comparing the distances between the test point and the candidate points to find the solutions.

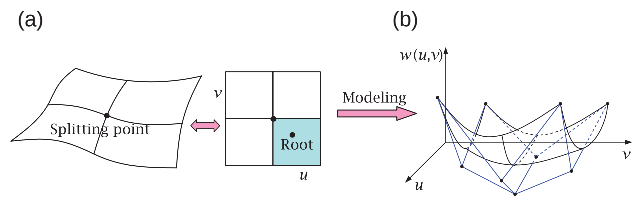

Fig. 1: Surface subdivision: (a) Bézier surface subdivision. Left: Bézier surfaces; Right: Parameter domain. (b) The first derivative surface of distance function

3. SUBDIVISION OF MODEL SURFACES

In the work, model surface is represented in B-spline form. The details of the mathematical description for such surfaces can be found in the literature [16]. For explanatory convenience, B-spline surfaces are reviewed briefly as follows. A B-spline surface of degree k in $u$direction and l in v direction is defined by

$$\boldsymbol{r}(u,v)=\sum_{i=0}^{m}\sum_{j=0}^{n}\boldsymbol{b}_{i,j}N_{i,k}(u)N_{j,l}(v) \tag{1}$$

where $\boldsymbol{b}_{i,j}$ are the control points of B-spline surface, and $N_{i,k}(u)$ and $N_{j,l}(v)$ are the normalized B-spline basis defined over the knot vectors U and V, respectively

$$U=\left[\underbrace{0,\cdots,0}_{k+1},u_{k+1},\cdots,u_m,\underbrace{1,\cdots,1}_{k+1}\right]$$

$$V=\left[\underbrace{0,\cdots,0}_{l+1},v_{l+1},\cdots,v_n,\underbrace{1,\cdots,1}_{l+1}\right]$$

So far, many efforts have been focused on subdivision of B-spline surface into Bézier surfaces. The most popular method is Oslo algorithm based on knot insertion developed by Boehm [2] and Cohen et al. [5] and proved by Prautzsch [18]. For practical applications, a more efficient way of using the Oslo algorithm is given by Lyche and Morken [13]. More detailed review is also referred to the literature [2,5,13,15,18]. The essential subdivision steps are briefly summarized as follows: Given a B-spline surface of degree k and l, defined by Eq. (1), a set of Bézier surface can be derived by inserting interior knot in U until its multiplicity is k and then inserting interior knot in V until its multiplicity is l. An example of decomposing a B-spline surface into its piecewise Bézier form is shown in Fig. 2. In addition, a Bézier surface can also be split into four Bézier sub-surfaces at arbitrary parameters (u, v) by applying de Casteljau algorithm on the control points of the surface in u and v direction.

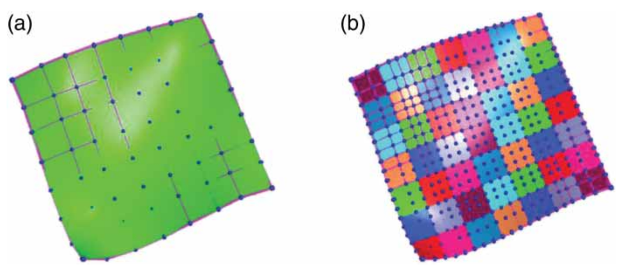

Fig. 2: Subdivision of B-spline surface into Bézier surfaces: (a) A cubic B-spline surface with 10 × 10 control points, (b) Bézier surfaces after decomposition.

4. FIRST DERIVATIVE SURFACE

In this section, the first derivative of the squared distance function is formulated as a Bernstein-form polynomial by AMBP and its graph can be described as a Bézier surface over the u-v parameter plane, such that the necessary condition for the closest point, $\nabla d_p^s(u,v) = 0$, can be re-explained as the first derivative surface is tangential to the u-v parametric plane and the tangential point is the closest point.

For convenience of the following processing, model surface has been subdivided into its piecewise Bézier form in Section 3. Here, the basic principles of the proposed method are explained using Bézier surface. Mathematically, the point projection can be described as to find a corresponding point of a given point $\boldsymbol{p}$ on a model surface $s(u,v)$ such that the distance between $\boldsymbol{p}$ and its corresponding point is minimal. The function to be minimized was

$$\min_{u,v}\left(d_p^s(u,v)\right) = \min_{u,v}\left(\|\boldsymbol{p} - \boldsymbol{r}(u,v)\|^2\right) \tag{2}$$

If the closest point is not a point on the surface boundary, the following condition is necessary, i.e. $\nabla d_p^s(u,v) = 0$. Thus, calculation of closest point is turned into a problem of solving the roots of $\nabla d_p^s(u,v) = 0$. In this paper, instead of using traditional numerical methods, a quadtree decomposition based method is given to solve it. An equivalent equation to $\nabla d_p^s(u,v) = 0$ is given by

$$w(u,v) = \left(\frac{\partial d_p^s(u,v)}{\partial u}\right)^2 + \left(\frac{\partial d_p^s(u,v)}{\partial v}\right)^2 = 0 \tag{3}$$

The partial derivatives of the squared distance function $d_p^s(u,v)$ with respect to the parameters u and v is given by

$$\frac{\partial d_p^s(u,v)}{\partial u} = 2\frac{\partial \boldsymbol{r}(u,v)}{\partial u}(\boldsymbol{p} - \boldsymbol{r}(u,v)) \tag{4}$$

$$\frac{\partial d_p^s(u,v)}{\partial v} = 2\frac{\partial \boldsymbol{r}(u,v)}{\partial v}(\boldsymbol{p} - \boldsymbol{r}(u,v)) \tag{5}$$

where

$$\frac{\partial r(u,v)}{\partial u} = m\sum_{i=0}^{m-1}\sum_{j=0}^{n} b_{i,j}^{1,0} B_{i,m-1}(u)B_{j,n}(v) \tag{6}$$

$$\frac{\partial r(u,v)}{\partial v} = n\sum_{i=0}^{m}\sum_{j=0}^{n-1} b_{i,j}^{0,1} B_{i,m}(u)B_{j,n-1}(v) \tag{7}$$

$b_{i,j}^{1,0}$ and $b_{i,j}^{0,1}$ are the first forward difference vector of the control points $\boldsymbol{b}_{\mathrm{i,j}}$ of Bézier surface $\boldsymbol{r}(u,v)$. And the point $\boldsymbol{p}$ can be described as a degenerative Bézier surface by using the normality of Bernstein basis as follows

$$\boldsymbol{s}_P(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \boldsymbol{d}_{i,j} B_{i,m}(u)B_{j,n}(v) \tag{8}$$

where $\boldsymbol{d}_{i,j} = \boldsymbol{p}, i = 0, 1, \cdots, m; j = 0, 1, \cdots n$. By using the arithmetic operation of subtraction of two bivariate Bernstein-form polynomials referred to [1], $\boldsymbol{p} - \boldsymbol{r}(u,v)$ can be formulated as

$$\boldsymbol{p} - \boldsymbol{r}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \boldsymbol{e}_{i,j} B_{i,m}(u)B_{j,n}(v) \tag{9}$$

where $\mathrm{e}_{\mathrm{i,j}} = \mathrm{d}_{\mathrm{i,j}} - \boldsymbol{b}_{\mathrm{i,j}}, i = 0, 1, \cdots, m; j = 0, 1, \cdots, n$. Then, substituting Eq. (6) and (9) into Eq. (4), $\partial d_p^s(u,v)/\partial u$ can be rewritten as

$$\begin{cases} \dfrac{\partial d_p^s(u,v)}{\partial u} = \displaystyle\sum_{i=0}^{2m}\sum_{j=0}^{2n} f_{i,j} B_{i,2m}(u)B_{j,2n}(v) \\ f_{i,j} = \displaystyle\sum_{k=\max(0,i-r)}^{\min(2m-1,i)} \sum_{l=\max(0,j-s)}^{\min(2n,j)} \frac{C_m^k C_1^{i-k} C_l^n C_0^{j-1}}{C_{2m}^i C_{2n}^j} F_{k,l} \\ F_{i,j}^{(2m-1,2n)} = \displaystyle\sum_{k=\max(0,i-p)}^{\min(s,i)} \sum_{l=\max(0,j-q)}^{\min(t,j)} \\ \quad \times \dfrac{C_s^l C_p^{i-1} C_t^k C_q^{j-k}}{C_{s+p}^i C_{t+q}^j} (\boldsymbol{b}_{l,k}^{1,0} \cdot \boldsymbol{e}_{i-1,j-k}) \end{cases} \tag{10}$$

where $f_{i,j}$ are the Bernstein coefficients of $\partial d_p^s(u,v)/\partial u$, and s, t, p and q are, respectively, the maximum degrees of the polynomials $\partial r(u,v)/\partial u$ and $\boldsymbol{p}-\boldsymbol{r}(u,v)$. Similarly, $\partial d_p^s(u,v)/\partial v$ can be rewritten as

$$\begin{cases} \dfrac{\partial d_p^s(u,v)}{\partial u} = \displaystyle\sum_{i=0}^{2m}\sum_{j=0}^{2n} h_{i,j}B_{i,2m}(u)B_{j,2n}(v) \\ h_{i,j} = \displaystyle\sum_{k=\max(0,i-r)}^{\min(2m,i)}\ \sum_{l=\max(0,j-s)}^{\min(2n-1,j)} \dfrac{C_m^k C_1^{i-k} C_l^n C_0^{j-1}}{C_{2m}^i C_{2n}^j} H_{k,l} \\ H_{i,j}^{(2m,2n-1)} = \displaystyle\sum_{k=\max(0,i-p)}^{\min(s,i)}\ \sum_{l=\max(0,j-q)}^{\min(t,j)} \\ \quad \times \dfrac{C_s^l C_p^{i-1} C_t^k C_q^{j-k}}{C_{s+p}^i C_{t+q}^j} (\boldsymbol{b}_{l,k}^{1,0} \cdot \boldsymbol{e}_{i-1,j-k}) \end{cases} \tag{11}$$

After substituting Eq. (10) and (6) into Eq. (3), by using the AMBP, $w(u,v)$ can be rewritten as a bivariate Bernstein-form polynomial as shown in Eq. (12)

$$\begin{cases} w(u,v) = \displaystyle\sum_{i=0}^{4m}\sum_{j=0}^{4n} g_{i,j}B_{i,4m}(u)B_{j,4n}(v) \\ g_{i,j} = x_{i,j} + y_{i,j} \\ x_{i,j} = \displaystyle\sum_{k=\max(0,i-p)}^{\min(s,i)}\ \sum_{l=\max(0,j-q)}^{\min(t,j)} \\ \quad \times \dfrac{C_s^l C_p^{i-l} C_t^k C_q^{j-k}}{C_{s+p}^i C_{t+q}^j} (f_{l,k} \cdot f_{i-1,j-k}) \\ y_{i,j} = \displaystyle\sum_{k=\max(0,i-p)}^{\min(s,i)}\ \sum_{l=\max(0,j-q)}^{\min(t,j)} \\ \quad \times \dfrac{C_s^l C_p^{i-1} C_t^k C_q^{j-k}}{C_{s+p}^i C_{t+q}^j} (h_{l,k} \cdot h_{i-1,j-k}) \end{cases} \tag{12}$$

where $g_{i,j}$ are the Bernstein coefficients, s and p is $2m$, t and q is $2n$. The graph of $w(u,v)$ can be descri-bed by a Bézier surface over the u-v parameter plane, and it is called as the first derivative surface in this paper and is modeled by the following parametric equation

$$\begin{cases} \boldsymbol{s}_w : \boldsymbol{w}(u,v) = \displaystyle\sum_{i=0}^{4m}\sum_{j=0}^{4n} \boldsymbol{g}_{i,j}B_{i,4m}(u)B_{j,4n}(v) \\ \boldsymbol{g}_{i,j} = \left[\dfrac{i}{4m}, \dfrac{j}{4n}, g_{i,j}\right]^{\mathrm{T}} \end{cases} \tag{13}$$

where $\boldsymbol{g}_{i,j}$ are the control points of the first derivative surface $\boldsymbol{s}_{\mathrm{w}}$. From Eq. (3), it is seen that $w(u,v)$ is non-negative, namely, no portion of $\boldsymbol{s}_{\mathrm{w}}$ lies below the u-v parametric plane, which implies that if $w(u,v)$ is equal to zero, namely $\nabla d_p^s(u,v)=0$, $\boldsymbol{s}_{\mathrm{w}}$ has to be tangential to the u-v parametric plane, and the tangent point is the closest point. In other words, if the first derivative surface $\boldsymbol{s}_{\mathrm{w}}$ is completely above the u-v parametric plane, the surface $\boldsymbol{s}(u,v)$ will be regarded as an invalid surface to be discarded.

5. METHODS FOR FINDING THE CLOSEST POINT

5.1. Elimination Criterion for Invalid Surface

To narrow the domains including the potential solutions, the invalid Bézier surfaces not containing the solutions must been eliminated as much as possible. For this purpose, a simple and efficient criterion for eliminating the invalid surface is given in this section.

Criterion 1. For surface, the signs of Bernstein coefficients $g_{i,j}$ in Eq. (12) are checked, if all signs are positive, the first derivative surface must be not tangential to the u-v parametric plane such that the test surface $\boldsymbol{s}(u,v)$ is regarded as an invalid surface and must be eliminated.

This criterion derives from the variation diminishing property of Bézier surface. As shown in Fig. 3, if the first derivative surface is tangential to the u-v parametric plane, its control points are distributed on both sides of the plane, and if the convex hull or the control points of the first derivative surface are completely above the u-v parametric plane, the first derivative surface must not be tangential to the u-v parametric plane and this means that the closest point is not on the tested model surface. For all Bézier surfaces obtained by subdivision in Section 3, the above checking procedure is repeated. The surface patches passing this test are selected as candidate surfaces to be further subdivided.

5.2. Algorithm for Finding Closest Point on the Model Surface

Once a Bézier surface is selected as the candidate surface, its corresponding derivative surface will be subdivided recursively and at the same time the signs of the Bernstein coefficients $g_{i,j}$ of $w(u,v)$ also be checked in the subdivided parameter domain. An adaptive quadtree decomposition on the u-v domain is adopted to narrow the domain possibly containing the tangent point, and de Casteljau algorithm is used to subdivide the u-v domain into four sub-domains at the midpoint of u and v. In searching the closest point, $\boldsymbol{s}_w$ is subdivided recursively and control points of sub-$\boldsymbol{s}_{w,i}$ are checked simultaneously. If all control points are completely above the u-v parametric plane, the node of the corresponding parameter region is marked as one excluding the roots. The searching process stops a depth d_t where the size of the domain is less than a given threshold ϵ_0, namely $2^{-d_t} \le \epsilon_0$. Then the quadtree is traversed and all unmarked nodes are collected at d_t from which the intervals $[u_l,u_h]\times[v_l,v_h]$ containing possibly the closest point is produced and the closest point is calculated by $\boldsymbol{r}((u_l+u_h)/2,(v_l+v_h)/2)$. Figure 4 shows an example of the adaptive quadtree decomposition.

For each candidate Bézier surfaces, the procedure is repeated and the midpoints of all marked domains

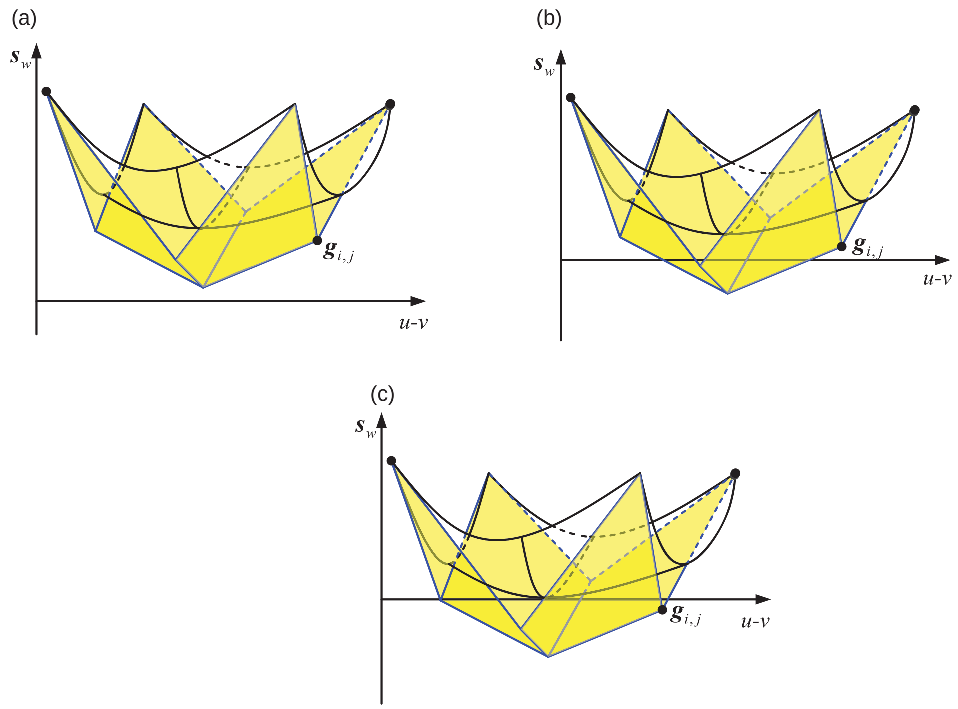

Fig. 3: Relationship between the control points of the first derivative surface and the $u - v$ parametric plane.

are regarded as the candidate points of the closest point. If all Bézier surfaces are considered to be invalid surfaces, this means that the closest point is not on the model surface but its boundaries, the closest point of the test point can be calculated on the surface boundaries by our previous method that addresses the problem of the closest point on a parametric curve [24]. Thus, the closest points of a test point on the surface boundaries should be also added into the collection of the candidate points. Then, by comparing the distances between the test point and the candidate points, the closest point can be found on the model surface of the test point.

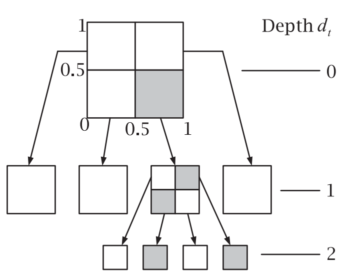

Fig. 4: Adaptive quadtree decomposition. The marked gray domains indicate those which possibly contain the solutions.

As mentioned previously, using knot insertion technique, B-spline surface can be easily subdivided into its piecewise Bézier form. For each Bézier surface, the above algorithm is implemented to judge whether $w(u, v) = 0$ hold or not. For Bézier surfaces satisfying this condition, quadtree decomposition is implemented to find the closest point. In such a way, the proposed method is generalized nicely to B-spline surface. Since the proposed method does not involve any iteration, it avoids the requirement of providing a good initial value for achieving the proper result and can also guarantee the root for all conditions.

6. EXAMPLES

The proposed algorithm has been implemented on a PC in C++, and in the following some examples will be given to demonstrate its effectiveness and robustness. As Selimonic [19] has ever pointed out that CPU performance time strongly depends on hardware and programming strategy, however the robustness is only associated with the algorithm itself, and also as mentioned earlier, for the practical industrial applications such as localization and inspection of the machined parts based on CAD model, the robustness is the main focus of interest. Thus, here the emphasis is the ability that the proposed method deals correctly with the point projection in any situations.

6.1. Example 1: Point Projection onto B-spline Surface

Figure 5 shows an example that a point is projected onto a 3D bicubic B-spline surface used earlier in Fig. 2. The size of the control net of the surface is 10×10 and the test point is (50.00, 50.00, 50.00).

Fig. 2(b) shows the obtained Bézier surface patches by subdividing this B-spline surface, and for all Bézier patches the first derivative surfaces are constructed by the formulae given in Section 4 and are used to eliminate the invalid surface patches by *Criterion* 1 of Section 5.1. After implementing the above test, among 49 Bézier patches, only one surface is valid surface which contains possibly the closest point of the test point and 98% of Bézier surface patches (48 invalid surfaces) are eliminated, as shown in Fig. 5(a). And then this one valid surface is further subdivided until the size of its parameter domain is less than a user-specified tolerance. Fig. 5(b) shows the closet point obtained by the proposed method.

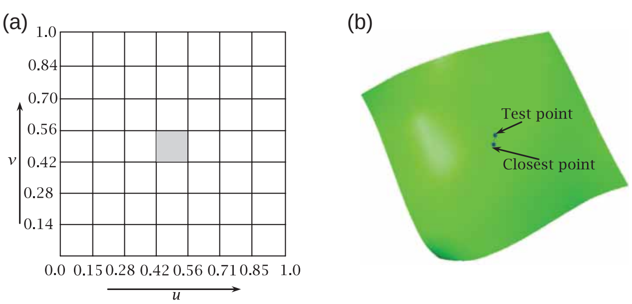

Fig. 5: Point projection onto a bicubic B-spline surface and the remarked domain indicates one which possibly contains solution: (a) Elimination of invalid surface patches, (b) test point and closest point on the surface.

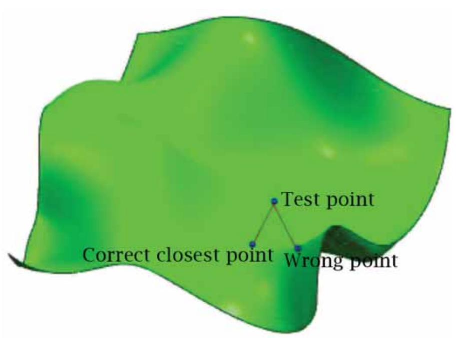

Fig. 6: Comparison between the proposed method and Newton-Raphson method.

6.2. Example 2: Comparison of our Method and Newton-Raphson Method

When applying iteration-type methods on the whole CAD surface, Ma and Hewitt [14] have pointed out that widely used iteration methods such as Newton-Raphson method possibly led to a wrong result. In this our test, for a point shown in Fig. 6, Newton-Raphson method produces a wrong answer when the subdivision interval is set as 10^{-3}. The reason is under this subdivision interval the obtained initial point for iteration process is closer to that wrong projection. If a smaller subdivision interval such as 10^{-5} or 10^{-6} is used, Newton-Raphson method can lead to the proper projection that is same as that of the proposed method and is shown in Fig. 6. The dependence on a good initial value makes iteration-type methods not providing full assurance that all solutions have been found in any situations and such good initial value is also hard to obtain due to the complexity of the shape of B-spline surface.

6.3. Example 3: Point Projection on the Surface Boundary

Piegl [17] and Ma [14] both point out that iteration-type methods are an error prone process that fails often especially for some boundary points. In this situation, the proposed method always gives right results. Fig. 7 illustrates the result of projecting the points onto the surface boundary.

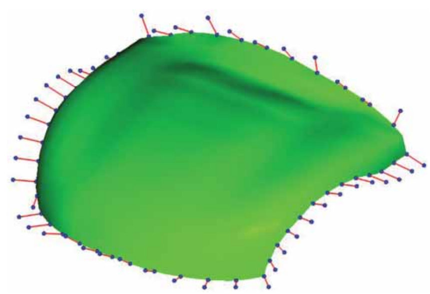

Fig. 7: Point projection on the surface boundary.

6.4. Example 4: Application to Localization Algorithm for Machining Allowance Optimization

Optimal localization for the machining allowance optimization refers to a process of determining the position and orientation of the design frame relative to the machine frame when a casting or forging blank part is arbitrarily fixed to a machine table, and then the optimal transformation matrix is used to process the initially generated tool paths which acts on the blank as if it is accurately fixed to the machine table. This process involves a set of measured data and a nominal CAD model and amounts to find a rigid body motion, transforming the measurement points to coincide with the design surface as close as possible. It is highly preferable to push each measured point outside the model surface in order to ensure that the surfaces of interest can be machined with sufficient machining allowances. In this case, the machining allowance is represented by the distances between the points measured from the machined parts and its projection on the nominal CAD surface.

As mentioned earlier, if wrong calculation result of the closest point occurs, the blank part could be considered to be substandard and has to be reworked due to the material shortage of some areas resulted

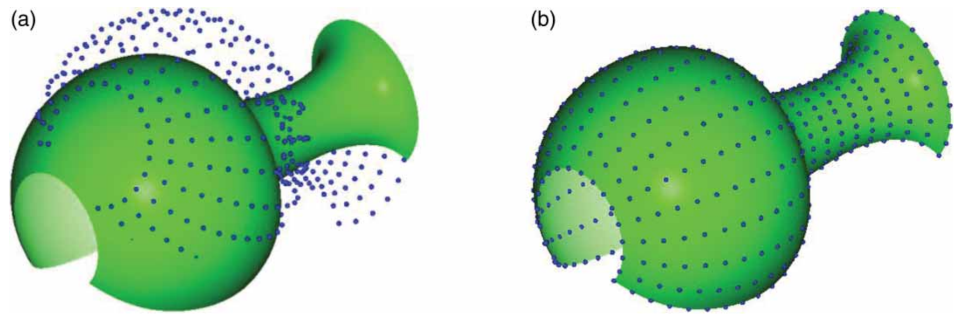

Fig. 8: Application of the proposed method to localization algorithm: (a) Initial position and orientation of measurement points to model surface, (b) localization result after matching with oriented distance constraints by the method [22]

from the calculation error, even though the nominal model actually can be enclosed within the blank part. From a practical point of view, iteration-type methods may not be preferred although they are faster than that based on subdivision strategy. For industrial applications, the robustness of calculation of the closest point may be found to be more important and economical than the savings of the computing time. The proposed method is to adaptively narrow down the surface to approach to the closest point by using surface subdivision rather than iteration processing, and it has been demon-strated by the test results that the proposed method can provide full assurance for all solutions and is thus nicely applicable to localization processing compared with iteration-type methods. Fig. 8 shows an example that the proposed method is used in localization of the measurement data to the nominal model surface, and the calculated closest distances are used as constraints of the machining allowance optimization to ensure that the blank parts can be machined with sufficient machining allowances.

7. ALGORITHM ANALYSIS

In the proposed method, the strategy of adaptive quadtree decomposition is used to narrow down the regions possibly including the projection. For a candidate surface, its parameter domain is subdivided into four sub-domains at the midpoint of u and v. At a depth d_t, there are at most 4^{d_t} nodes and each node has an interval of size $s2^{-d_t} \le u \le (s+1)2^{-d_t}$ and $t2^{-d_t} \le v \le (t+1)2^{-d_t}$ where s and t are integers of $0 \le s \le 4^{(d_t-1)}$ and $0 \le t \le 4^{(d_t-1)}$. Obviously, surface subdivision is the most time-consuming step of the proposed algorithm and it has significant influence on the CPU performance. Generally, how many subdivisions are implemented is not known in advance, but the maximum depth required to satisfy a user-specified tolerance ϵ_0 can be estimated. For a user-specified tolerance ϵ_0, our algorithm stops when the size of the subdivided domain is less than ϵ_0, namely $2^{-d_t} \le \epsilon_0$. In the worst case, subdivision happens twice at every node up to depth $d_t - 1$, such that the total number of applications of the de Casteljau algorithm is $2\sum_{i=0}^{d_t} 4^i$ [9]. For a Bézier surface of degree m in u direction and n in v direction, the application of de Casteljau algorithm in u and v direction is $O(m^2 + n^2)$. Thus, the time complexity in the worst case of the algorithm is $O(4^{d_t}(m^2 + n^2))$. As an example, when a sphere and a point on the center of the sphere are provided as input to the algorithm, the worst case will happen. However, generally time complexity could be much better than the worst case because for much of the model surface in industrial applications, the number of the closest point of a given point is limited and generally is one. In this situation, the subdivision on the invalid surface can be considerably reduced. As shown in Example 6.1, 98% of Bézier patches need not be subdivided after implementing Criterion 1 once. Despite this, its CPU performance is still slower than Newton-Raphson method in practical implementation. But, as mentioned earlier, the original intention of our method and the subdivision-based methods is to compensate for the deficiencies of robustness of iteration-type methods and give full assurance that all solutions have been properly found, which is very important for those practical industrial applications.

8. CONCLUSION

In this paper, a method for calculating the closest point on model surface is proposed. Different from the previous subdivision-based methods, the proposed method employs AMBP to formulize the first derivative of squared distance function into a Bernstein-from polynomial, and then by using the linear precision property of Bernstein polynomial, the graph of the first derivative is modeled by a Bézier surface. Using the position relationships of obtained Bézier surface and the u-v parametric plane, the domain which contains the potential solutions is iteratively subdivided until the closest point is found. Experimental results have demonstrated that the proposed criterion can eliminate efficiently those invalid surface patches, and in Example 6.1, only implementing Criterion 1 once, 98% of Bézier patches is determined to be invalid surfaces that need not be further subdivided. To some extent, this shows that the method is very effective for discarding the invalid surfaces, but CPU time performance is slower than Newton-Raphson method in the practical implementation of our examples. Programming strategy of recursive subdivision is considered to be an important reason, but as mentioned before, the computation time of the algorithm is not the focus of this paper, and effectiveness and robustness of algorithm are the most important for the industrial applications from a practical point of view. As demonstrated by the experimental results, the proposed method can provides full assurance that all solutions can be found for any conditions. Moreover, since it does not involve any iteration processes, our method also avoids the requirements of providing a good initial value for achieving the proper result. Thus, this method can be nicely applicable to the localization for machining allowance optimization and inspection of the machined parts with free-form surfaces.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (grant no. 51105058), China Postdoctoral Science Foundation (grant no. 2013M540221, 2014T70246) and the Fundamental Research Funds for the Central Universities (grant no. DUT14QY35).

REFERENCES

[1] Berchtold, J.; Bowyer, A.: Robust arithmetic for multivariate Bernstein-form polynomials, Computer-Aided Design, 32(11), 2000, 681–689.

[2] Boehm, W.: Inserting new knots into B-spline curves, Computer-Aided Design, 12(4), 1980, 199–201.

[3] Chen, X. D.; Su, H.; Yong, J. H.; Paul, J. C.; Sun, J. G.: A counterexample on point inversion and projection for NURBS curve, Computer Aided Geometric Design, 24(5), 2007, 302.

[4] Chen, X.; Yong, J. H.; Wang, G. Z.; Pail, J. C.; Xu, G.: Computing the minimum distance between a point and a NURBS curve, Computer-Aided Design, 40(10–11), 2008, 1051–1054.

[5] Cohen, E.; Lyche, T.; Riesebfeld, R.: Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics, Computer Graphics and Image Process, 14(2), 1980, 87–111.

[6] Dyllong, E.; Luther, W.: Distance calculation between a point and a NURBS surface, Proceedings of the 4th International Conference on Curves and Surfaces, Saint-Malo, France, July 1–7, 1999, 55–62.

[7] Hartmann, E.: On the curvature of curves and surfaces defined by normalforms, Computer Aided Geometric Design, 16(5), 1999, 355–376.

[8] Hu, S. M.; Wallner, J.: A second algorithm for orthogonal projection onto curves and surfaces, Computer Aided Geometric Design, 22(3), 2005, 251–260.

[9] Ko, K. H.; Maekawa, T.; Patrikalakis N. M.; Masuda H.; Wolter, F.-E.: Shape intrinsic properties for free-form object matching, Journal of Computing and Information Science in Engineering, ASME, 3(4), 2003, 325–333.

[10] Li, Y. D.; Gu, P. H.: Free-form surface inspection techniques state of the art review, Computer-Aided Design, 36(13), 2004, 1395–1417.

[11] Li, Y. D.; Gu, P. H.: Inspection of free-form shaped parts, Robotics and Computer-Integrated Manufacturing, 21(4–5), 2005, 421–430.

[12] Liu, X. M.; Lei, Y.; Yong, J. H.; Gu, H. J.; Sun, J. G.: A torus patch approximation approach for point projection on surfaces, Computer Aided Geometric Design, 26(5), 2009, 593–598.

[13] Lyche, T.; Morken, K.: Making the Oslo algorithm more efficient, Journal of Numerical Analysis, 23(3), 1986, 663–675.

[14] Ma, Y. L.; Hewitt, W. T.: Point inversion and projection for NURBS curve and surface: control polygon approach, Computer Aided Geometric Design, 20(2), 2003, 79–99.

[15] Mallón, P. N.; Bóo, M.; Bruguera, J. D.: Parallel architecture for conversion of NURBS curves to Bézier curves, Proceedings of the 26th EUROMICRO conference, Maastricht, 2000, 1324–1331.

[16] Piegl, L. A.; Tiller, W.: The NURBS Book, Springer-Verlag, New York, Second Edition, 1997.

[17] Piegl, L. A.; Tiller, W.: Parameterization for surface fitting in reverse engineering, Computer-Aided Design, 33(8), 2001, 593–603.

[18] Prautasch, H.: A short proof of the Oslo algorithm, Computer Aided Geometric Design, 1(1), 1984, 95–96.

[19] Selimovic, I.: Improved algorithm for the projection of points on NURBS curves and surfaces, Computer Aided Geometric Design, 23(5), 2006, 439–445.

[20] Shen, B.; Huang, G. Q.; Mak K. L.; Wang, X. C.: A best-fitting algorithm for optimal location of large-scale blanks with free-form surfaces, Journal of Materials Processing Technology, 139(1–3), 2003, 310–314.

[21] Sun, Y. W.; Wang, X. M.; Guo, D. M.; Liu, J.: Machining localization and quality evaluation of parts with sculptured surfaces using SQP method, International Journal of Advanced Manufacturing Technology, 42(11–12), 2009, 1131–1139.

[22] Sun, Y. W.; Xu, J. T.; Guo, D. M.; Jia, Z. Y.: A unified localization approach for machining allowance optimization of complex curved surfaces, Precision Engineering, 33(4), 2009, 635–641.

[23] Wang, H.; Kearney, J.; Atkinson, K.: Robust and efficient computation of the closest point on a spline curve, Proceeding of the 5th International Conference on Curves and Surfaces, 2002, 397–406.

[24] Xu, J. T.; Liu, W. J.; Bian, H. Y.; Li, L.: Accurate and efficient algorithm for the closest point on a parametric curve, International Conference on Computer Science and Software Engineering, Wuhan, China, Dec 22–24, 2, 2008, 1000–1002.

[25] Zhou, J. M.; Sherbrooke, E. C.; Patrikalakis, N.: Computation of stationary points of distance functions, Engineering with Computers, 9(4), 1993, 231–246.