

# An Efficient Sensing Localization Algorithm for Free-Form Surface Digitization

Yunbao Huang

Xiaoping Qian

e-mail: qian@iit.edu

Mechanical, Materials and Aerospace  
Engineering,  
Illinois Institute of Technology,  
Chicago, IL 60616

We present a divide-and-conquer method that efficiently finds a near-optimal distribution of sensing locations for free-form surface digitization. We formulate a next-best-point problem and transform the uncertainty of a B-spline surface into a higher-dimensional B-spline surface. This technique allows the use of the convex hull and subdivision properties of B-spline surfaces in the divide-and-conquer algorithm. It thus greatly reduces the search time for determining the next best sensing location.

[DOI: 10.1115/1.2904931]

**Keywords:** sensing localization, next best point, B-spline surface, uncertainty, surface reconstruction

## 1 Introduction

Free-form surfaces appear widely in products from various manufacturing industries. Reverse engineering, the process of reconstructing the computer representation of physical surfaces, has become indispensable in the development of these products. Geometric shape acquisition is a first step in reverse engineering, and it provides 3D coordinates of physical objects for the reconstruction process.

The reconstructed surface has uncertainty due to sensor noise. *Sensing localization* is a process aiming to determine the sensing locations to reduce the uncertainty of the reconstructed surface. Inappropriate choices of sensing locations can lead to unnecessarily large uncertainty in the resulting surface. Ineffective sensing localization may require lengthier time for sensing.

Existing methods for sensing localization in free-form surface digitization are either gradient based or exchange based [1–5]. These methods tend to either produce locally optimal sensing locations or converge slowly and work in a nondeterministic manner.

We present a sensing localization method for point sensors that acquire data on a point-by-point basis, such as tactile probes in coordinate measurement machines (CMM) or point laser scanners. The key insight in our method is the transformation of the uncertainty distribution of a B-spline free-form surface into a higher-dimensional B-spline surface. The surface point of maximal uncertainty can then be efficiently located based on the convex hull and subdivision properties of B-spline surfaces. Based on this technique, we incrementally obtain near-optimal sensing locations through a divide-and-conquer strategy. The steps in this approach are as follows.

1. Formulate the sensing localization problem as a *next-best-point* (NBP) problem.
2. Transform the uncertainty of the reconstructed surface into a higher-dimensional B-spline surface. The NBP is thus naturally interpreted as the highest point on the new surface.
3. Use a divide-and-conquer strategy to locate the NBP based on the subdivision and the convex hull properties of a B-spline surface.

4. Repeat the above process to obtain multiple sensing locations.

Due to its computational efficiency, this approach has been used in online sensing localization, where the next sensing location is dynamically determined and the sensed data are used to steer the next sensing location.

The remainder of this paper is organized as follows. Section 2 reviews prior work in sensing localization. Section 3 introduces surface reconstruction and uncertainty computation. Section 4 describes the principle of uncertainty-based sensing localization and our divide-and-conquer based sensing localization algorithm. Section 5 presents experimental validation results. Section 6 analyzes the algorithm efficiency. This paper concludes in Sec. 7.

## 2 Literature Review

Sensor planning for point sensors has been an active research topic for over a decade [6–11]. Uniform distribution of sensing locations is often adopted [5,12] and surface features such as patch size and mean curvature are also considered in distributing the sensing locations [13]. However, in these cases, the uncertainty of the resulting surface is not minimized.

High-level control of sensing locations is demonstrated in Ref. [14] based on the uncertainty of a regression model. However, only a simple superellipsoid model is employed to represent the object and a gradient method is used to search for locally optimal sensing locations. Closed B-spline section curves are utilized to model the 3D free-form object and uncertainty-based sensing localization is proposed in Ref. [15], but there is no discussion on how to search the optimal sensing location space. Sensing localization considering reconstructed surface uncertainty has been attempted in Refs. [2,5], and simulated annealing and conjugate gradient methods were introduced to reduce the surface uncertainty. The results based on gradient methods are locally optimal and the simulated annealing approach is slow and nondeterministic.

Sensing localization is also an important research topic in many other applications such as fixture design [3,16], and workpiece localization [4,17], where interchange methods are used to optimize sensing locations on the discretized point-set domain. In the design of experiment community, sensing localization is recognized as a general measurement design problem, and Fedorov exchange [18,19] is often adopted to determine sensing locations. However, it converges slowly and is nondeterministic. Acceler-

Contributed by the Computer Aided Product Development (CAPD) Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received May 16, 2006; final manuscript received September 29, 2007; published online May 16, 2008. Review conducted by J. Corney.

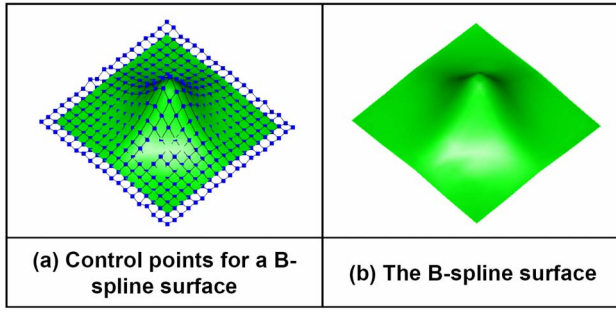


Fig. 1 B-spline surface representation

ated algorithms have been proposed, for example, by increasing the number of exchange pairs and deleting the candidate points that are less desirable [20].

### 3 B-Spline Surface Reconstruction, Uncertainty, and Sensing Localization

This section first describes the  $B$ -spline representation for free-form surfaces and its uncertainty computation. We then proceed to analyze the uncertainty-based sensing localization.

**3.1 B-Spline Surface.** The  $B$ -spline surface representation is a common form for modeling free-form shapes [21]. We use this representation for the surface reconstructed from the sensed data.

A bicubic  $B$ -spline surface such as the one seen in Fig. 1 can be represented as

$$S(u, v) = \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} B_i(u) B_j(v) \mathbf{P}_{ij} \quad (1)$$

where  $B$  is a  $B$ -spline shape function,  $n_u$  is the number of control points along  $u$  direction,  $n_v$  is the number of control points along  $v$  direction, and  $\mathbf{P}_{ij}$  is the  $ij$ th control point. The equation can also be expressed as the inner product of two vectors as follows:

$$S(u, v) = \mathbf{A}^T \mathbf{P} \quad (2)$$

where  $\mathbf{A} = \mathbf{A}(u, v)$  is the  $B$ -spline shape function vector ( $\mathbf{A} \in \mathcal{R}^n$ ,  $n = n_u \cdot n_v$ ), and  $\mathbf{P}$  is the collection of control points ( $\mathbf{P} \in \mathcal{R}^n$ ,  $n = n_u \cdot n_v$ ).

**3.2 B-Spline Surface and Uncertainty From Measurements.** Given  $k$  measurements  $z_1, z_2, \dots, z_k$  with uncertainty  $\Lambda_{z_1}, \Lambda_{z_2}, \dots, \Lambda_{z_k}$  and a  $B$ -spline surface structure (the number of control points, knot vector, and degree), we can reconstruct a  $B$ -spline surface from those measurements via a weighted least squares method. The objective function  $f(\mathbf{P}_k)$  will be defined as the weighted distance between the surface and the measurements as follows:

$$f(\mathbf{P}_k) = \sum_{i=1}^k \frac{(z_i - \mathbf{A}_i^T \mathbf{P}_k)^2}{\Lambda_{z_i}} \quad (3)$$

The vector  $\mathbf{P}_k$  contains the control points for the  $B$ -spline surface from  $k$  measurements, and  $\mathbf{A}_i$  is the  $B$ -spline shape function vector for the  $i$ th measurement  $z_i$ .

The gradient of  $f(\mathbf{P}_k)$  is

$$\begin{aligned} \nabla f(\mathbf{P}_k) &= \nabla \left( \sum_{i=1}^k \frac{(z_i - \mathbf{A}_i^T \mathbf{P}_k)^2}{\Lambda_{z_i}} \right) = -2 \sum_{i=1}^k \frac{\mathbf{A}_i (z_i - \mathbf{A}_i^T \mathbf{P}_k)}{\Lambda_{z_i}} \\ &= -2 \left( \sum_{i=1}^k \mathbf{A}_i (\Lambda_{z_i})^{-1} z_i - \sum_{i=1}^k \mathbf{A}_i (\Lambda_{z_i})^{-1} \mathbf{A}_i^T \mathbf{P}_k \right) \end{aligned} \quad (4)$$

The zero of the gradient is given by

$$\mathbf{P}_k = (\hat{\mathbf{A}}_k \Lambda_{z_k}^{-1} \hat{\mathbf{A}}_k^T)^{-1} \hat{\mathbf{A}}_k \Lambda_{z_k}^{-1} \mathbf{Z}_k \quad (5)$$

where  $\hat{\mathbf{A}}_k = [\mathbf{A}_1 \mathbf{A}_2 \dots \mathbf{A}_k]$ ,  $\Lambda_{z_k} = \text{diag}(\Lambda_{z_1}, \Lambda_{z_2}, \dots, \Lambda_{z_k})$ , and  $\mathbf{Z}_k = [z_1 z_2 \dots z_k]^T$ .

It is reasonable to assume that the measurements in  $\mathbf{Z}_k$  are samples from a Gaussian probability distribution with zero mean. From Eq. (5), we can get the distribution of control points  $\mathbf{P}_k$ , would then likewise satisfy a Gaussian distribution.

Let  $\mathbf{M}_k = \hat{\mathbf{A}}_k \Lambda_{z_k}^{-1} \hat{\mathbf{A}}_k^T$ . The uncertainty covariance matrix is given by

$$\Lambda_{\mathbf{P}_k} = (\mathbf{M}_k^{-1} \hat{\mathbf{A}}_k \Lambda_{z_k}^{-1}) \Lambda_{z_k} (\mathbf{M}_k^{-1} \hat{\mathbf{A}}_k \Lambda_{z_k}^{-1})^T = \mathbf{M}_k^{-1} \hat{\mathbf{A}}_k \Lambda_{z_k}^{-1} \hat{\mathbf{A}}_k^T (\mathbf{M}_k^{-1})^T = \mathbf{M}_k^{-1} \quad (6)$$

The matrix  $\mathbf{M}_k$  is known as the Fisher information matrix [22].

The uncertainty,  $\Lambda_p$ , of a surface point  $p$  at the surface's parametric domain  $(u, v)$  can be defined by

$$\Lambda_p = \mathbf{A}^T \Lambda_{\mathbf{P}_k} \mathbf{A} \quad (7)$$

When all measurements are taken by the same sensor and have identical variances, i.e.,  $\Lambda_z \equiv \Lambda_{z_i}$  ( $1 \leq i \leq k$ ), the uncertainty covariance and Fisher information matrices can be simplified to

$$\Lambda_{\mathbf{P}_k} = \Lambda_z (\hat{\mathbf{A}}_k \hat{\mathbf{A}}_k^T)^{-1} \quad \text{and} \quad \mathbf{M}_k = (\Lambda_z)^{-1} \hat{\mathbf{A}}_k \hat{\mathbf{A}}_k^T \quad (8)$$

**3.3 Uncertainty Based Sensing Localization.** Since the control points  $\mathbf{P}_k$  of the reconstructed surface satisfy a Gaussian probability distribution, the uncertainty  $\Lambda_{\mathbf{P}_k}$  constitutes a measure of information contained in the Gaussian probability distribution  $\mathcal{Q}(\mathbf{P}_k) \in \mathcal{R}$  describing the parameter errors

$$\mathcal{Q}(\mathbf{P}_k) = (\mathbf{P}_k - \hat{\mathbf{P}}_k)^T \Lambda_{\mathbf{P}_k} (\mathbf{P}_k - \hat{\mathbf{P}}_k) \quad (9)$$

where  $\hat{\mathbf{P}}_k$  is the maximum likelihood estimate of control points  $\mathbf{P}_k$ . Given a confidence level  $\gamma$  (a distribution probability of  $\mathbf{P}_k$ ), we can find a  $\mathcal{Q}_\gamma$  from the distribution. Moreover, the true model must lie in the hyperellipsoid defined by  $\mathcal{Q}_\gamma$ . In Eq. (9), it can be seen that a hyperellipsoid of smaller volume implies that the reconstructed surface has higher confidence and better approximates the underlying shape. The size of the hyperellipsoid is dependent only on  $\Lambda_{\mathbf{P}_k}$  (or  $\mathbf{M}_k$ ). There are various criteria to characterize  $\Lambda_{\mathbf{P}_k}$  (or  $\mathbf{M}_k$ ) such as determinant, eigenvalue, and trace [19], and among those criteria,  $D$ -optimality has been widely used due to its simple updating formulas [19,20]. These formulas are as follows.

- (1) Equations for updating  $\Lambda_{\mathbf{P}_k}$ ,  $\mathbf{M}_k$ , and their determinants when adding or withdrawing a single data point  $z$  with  $B$ -spline shape function matrix  $\mathbf{A}_z$  and uncertainty  $\Lambda_z$  as follows:

$$\mathbf{M}'_k = \mathbf{M}_k \pm \mathbf{A}_z (\Lambda_z)^{-1} \mathbf{A}_z^T \quad (10)$$

$$\Lambda_{\mathbf{P}'_k} = (\mathbf{I} \mp \Lambda_{\mathbf{P}_k} \mathbf{A}_z (\Lambda_z \pm \mathbf{A}_z^T \Lambda_{\mathbf{P}_k} \mathbf{A}_z)^{-1} \mathbf{A}_z^T) \Lambda_{\mathbf{P}_k} \quad (11)$$

$$\begin{aligned} \det(\mathbf{M}'_k) &= \det(\mathbf{M}_k \pm \mathbf{A}_z (\Lambda_z)^{-1} \mathbf{A}_z^T) \\ &= \det(\mathbf{M}_k) \cdot \det(\mathbf{I} \pm \mathbf{A}_z \mathbf{A}_z^T (\mathbf{M}_k)^{-1} (\Lambda_z)^{-1}) \end{aligned} \quad (12)$$

Since the rank of  $\mathbf{A}_z \mathbf{A}_z^T (\mathbf{M}_k)^{-1} (\Lambda_z)^{-1}$  is one and has only one nonzero eigenvalue  $\mathbf{A}_z^T (\mathbf{M}_k)^{-1} \mathbf{A}_z (\Lambda_z)^{-1}$ ; therefore,  $\det(\mathbf{I} \pm \mathbf{A}_z \mathbf{A}_z^T (\mathbf{M}_k)^{-1} (\Lambda_z)^{-1}) = (1 \pm \mathbf{A}_z^T \Lambda_{\mathbf{P}_k} \mathbf{A}_z (\Lambda_z)^{-1})$ ,  $\det(\mathbf{M}'_k) = \det(\mathbf{M}_k) (1 \pm \mathbf{A}_z^T \Lambda_{\mathbf{P}_k} \mathbf{A}_z (\Lambda_z)^{-1})$ , and

$$\det(\Lambda_{\mathbf{P}'_k}) = \frac{1}{\det(\mathbf{M}'_k)} = \frac{\det(\Lambda_{\mathbf{P}_k})}{1 \pm \mathbf{A}_z^T \Lambda_{\mathbf{P}_k} \mathbf{A}_z (\Lambda_z)^{-1}} \quad (13)$$

where  $\mathbf{M}'_k$  and  $\Lambda_{\mathbf{P}'_k}$  are the updated uncertainty matrices of  $\mathbf{M}_k$  and  $\Lambda_{\mathbf{P}_k}$ .

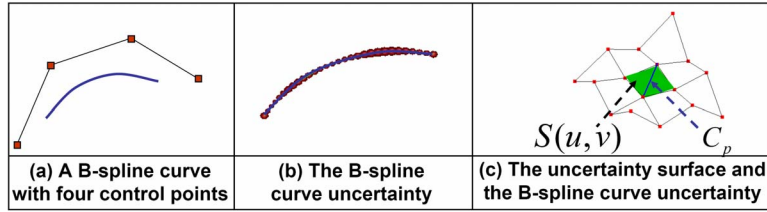


Fig. 2 Geometric interpretation of the uncertainty of a  $B$ -spline curve

(2) The computation of the change in the determinant,  $\Delta \det(\mathbf{M}_k)$ , when exchanging the point  $z$  with  $z'$  [19],

$$\begin{aligned} \Delta \det(\mathbf{M}_k) &= \det(\mathbf{M}_k - \mathbf{A}_z(\Lambda_z)^{-1}\mathbf{A}_z^T + \mathbf{A}_{z'}(\Lambda_{z'})^{-1}\mathbf{A}_{z'}^T) \\ &= (\Lambda_{z'}^{-1}(\mathbf{A}_{z'}^T\Lambda_{\mathbf{p}}\mathbf{A}_{z'} - \mathbf{A}_z^T\Lambda_{\mathbf{p}}\mathbf{A}_z(1 \\ &\quad + \mathbf{A}_{z'}^T\Lambda_{\mathbf{p}}\mathbf{A}_{z'}(\Lambda_z)^{-1}) + (\mathbf{A}_{z'}^T\Lambda_{\mathbf{p}}\mathbf{A}_z)^2(\Lambda_z)^{-1}) \end{aligned} \quad (14)$$

where  $\mathbf{A}_{z'}$  is the  $B$ -spline shape function matrix for  $z'$ , and  $\Lambda_{z'}$  is the variance of  $z'$ .

Using a  $B$ -spline surface representation and the  $D$ -optimality criterion, the sensor location problem can also be defined as follows: Given a free-form surface,  $B$ -spline surface structure, and a number of measurements  $k$  determine the sensing locations of  $k$  measurements to maximize  $\det(\mathbf{M}_k)$  or minimize  $\det(\Lambda_{\mathbf{p}_k})$ .

Equations (12) and (13) are highly nonlinear since the  $B$ -spline blending function  $A_z$  is a bicubic function of the parameter pair  $(u, v)$  and may have multiple peaks and valleys. When the number  $k$  is large, these equations are also of high dimension. Consequently, it is difficult to find the globally optimal distribution of multiple sensing locations. It is therefore beneficial to develop a method that can obtain near-optimal solutions in an efficient manner.

#### 4 NBP Based Sensing Localization Algorithm

Our next-best-point (NBP) method based on the  $D$ -optimal criterion obtains near-optimal sensing locations by iteratively choosing a sensing location that minimizes the surface uncertainty in the acquired data. We use a divide-and-conquer approach based on the subdivision and convex hull properties of the  $B$ -spline surface to rapidly find the NBP in the parametric domain.

**4.1 Problem Formulation.** At step  $i$  in the sensing process, we have an estimated surface and uncertainty. We will choose the next sensing location as the point that minimizes the determinant of the covariance matrix.

Let  $\Lambda_{\mathbf{p}_i}$  be the covariance matrix of current estimated control points, and  $z_{i+1}$  be the next measurement on the free-form surface. From Eqs. (11) and (13), the updated uncertainty covariance matrix is found to be

$$\Lambda_{\mathbf{p}_{i+1}} = (\mathbf{I} - \Lambda_{\mathbf{p}_i}\mathbf{A}_{z_{i+1}}(\Lambda_{z_{i+1}} + \mathbf{A}_{z_{i+1}}^T\Lambda_{\mathbf{p}_i}\mathbf{A}_{z_{i+1}})^{-1}\mathbf{A}_{z_{i+1}}^T)\Lambda_{\mathbf{p}_i} \quad (15)$$

and its determinant as

$$\det(\Lambda_{\mathbf{p}_{i+1}}) = \det(\Lambda_{\mathbf{p}_i}) / (1 + \mathbf{A}_{z_{i+1}}^T\Lambda_{\mathbf{p}_i}\mathbf{A}_{z_{i+1}}(\Lambda_{z_{i+1}})^{-1}) \quad (16)$$

where  $\mathbf{A}_{z_{i+1}}$  is the  $B$ -spline shape function related to  $z_{i+1}$ .

From Eq. (16), we can see that minimizing  $\det(\Lambda_{\mathbf{p}_{i+1}})$  is equivalent to maximizing  $\mathbf{A}_{z_{i+1}}^T\Lambda_{\mathbf{p}_i}\mathbf{A}_{z_{i+1}}$ . Thus, the NBP problem can be cast as finding the point on the reconstructed  $B$ -spline surface that has the maximal uncertainty.

**4.2 NBP Search on a Free-Form Curve.** Before describing the NBP search on a free-form surface, we will describe the process on a free-form curve.

**4.2.1 Uncertainty Surface for a Reconstructed  $B$ -Spline Curve.** Assume as given a reconstructed  $B$ -spline curve with control points  $\mathbf{P}$  and the uncertainty covariance matrix  $\Lambda_{\mathbf{p}}$ . The point  $p$  with parameter  $u$  the  $B$ -spline curve has uncertainty

$$\Lambda_p = \mathbf{A}^T(u)\Lambda_{\mathbf{p}}\mathbf{A}(u) \quad (17)$$

where  $\mathbf{A}(u)$  is the  $B$ -spline shape function vector for a  $B$ -spline curve. Equation (17) can also be written as

$$\Lambda p = \mathbf{A}^T(u)\Lambda_{\mathbf{p}}\mathbf{A}(v) \quad (v = u) \quad (18)$$

Let  $S(u, v) = \mathbf{A}(u)\Lambda_{\mathbf{p}}\mathbf{A}^T(v)$ . We can see the following.

- $S(u, v)$  is a  $B$ -spline height surface defined by the uncertainty matrix  $\Lambda_{\mathbf{p}}$  on the domain  $\Omega = \{(u, v) | 0 \leq u, v \leq 1\}$ . We call  $S(u, v)$  an *uncertainty surface*.
- The distribution of the uncertainty of the  $B$ -spline curve  $p(u)$  over  $u$  forms a space curve  $C_p$  on  $S(u, v)$  at  $\{v = u, u \in [0, 1]\} \subset \Omega$  (see Fig. 2). The uncertainty  $p(u)$  is  $\Lambda_p = S(u, u)$ .

The problem of finding the NBP with maximal uncertainty on a  $B$ -spline curve is equivalent to finding the highest point on the space curve  $C_p$  on  $S(u, v)$ , which we find using the following two properties of  $B$ -splines [21].

- The *convex hull* property, meaning that a  $B$ -spline curve is completely enclosed in the convex hull defined by its control points  $\mathbf{P}$ .
- The *subdivision* property, meaning that a  $B$ -spline curve can be subdivided into pieces, each of which retains the  $B$ -spline form.

**4.2.2 NBP Search Through a Divide-and-Conquer Strategy.** The core ideas of our divide-and-conquer method are (1) to eliminate the curve segments of lower uncertainty through the convex hull property of the  $B$ -spline uncertainty surface, and (2) to further subdivide each remaining  $B$ -spline curve segment into smaller segments.

For example, the uncertainty surface shown in Fig. 3(a) has one surface patch, and the blue diagonal line is the curve ( $v = u$ ) on the uncertainty surface that corresponds to the  $B$ -spline curve's uncertainty. To find the point on the  $B$ -spline curve with the maximal uncertainty, subdivision is performed on the uncertainty surface

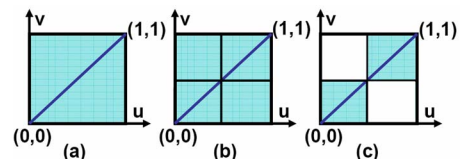


Fig. 3 Subdivision and extraction of uncertainty surface

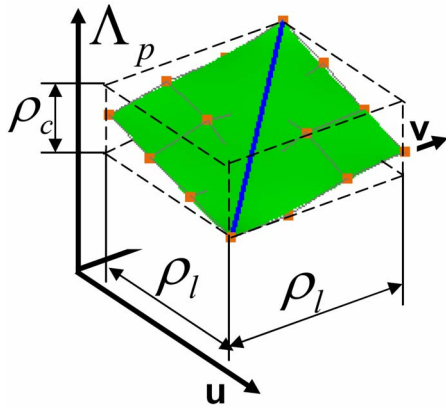


Fig. 4 Tolerance for the NBP

patch and the corresponding  $B$ -spline curve segment. Thus, four patches in Fig. 3(b) and two  $B$ -spline curve segments can be obtained. Among the four patches, only the two diagonal surface patches containing the curve ( $v=u$ ) need to be extracted. Among the extracted patches, any patch whose maximum is smaller than the minimum of another patch is eliminated.

Repeating the subdivision, extraction, and elimination process, we will find that the final subdivided curve segment converges to the optimal point within a user-specified tolerance  $\rho$ . The  $\rho$  includes two elements. One is the height  $\rho_c$  of the uncertainty surface's convex hull along its uncertainty height axis, and the other is the maximal interval length  $\rho_l$  of subdivided patches in the parametric domain as in Fig. 4.

Figure 5 details an example of the NBP computing process on a  $B$ -spline curve. In the upper-level pictures, the gray areas represent, in the parametric domain, the remaining subdivided patches after the elimination. The lower-level pictures show that the distribution of uncertainty for remaining patches after each elimination step converges to the NBP with the color corresponding to the uncertainty value.

**4.3 NBP Search on a Free-Form Surface.** Corresponding to the uncertainty equation for free-form curves, the uncertainty of a  $B$ -spline surface point is

$$\Lambda_p = \mathbf{A}^T(u, v) \Lambda_p \mathbf{A}(u, v) \quad (19)$$

which is on a  $B$ -spline surface in  $\mathfrak{R}^4$  defined by

$$S(u_1, v_1, u_2, v_2) = \mathbf{A}^T(u_1, v_1) \Lambda_p \mathbf{A}(u_2, v_2) \quad (20)$$

where  $S(u_1, v_1, u_2, v_2)$  is a higher-dimensional  $B$ -spline uncertainty manifold defined on the domain  $\Omega = \{(u_1, v_1, u_2, v_2) | 0$

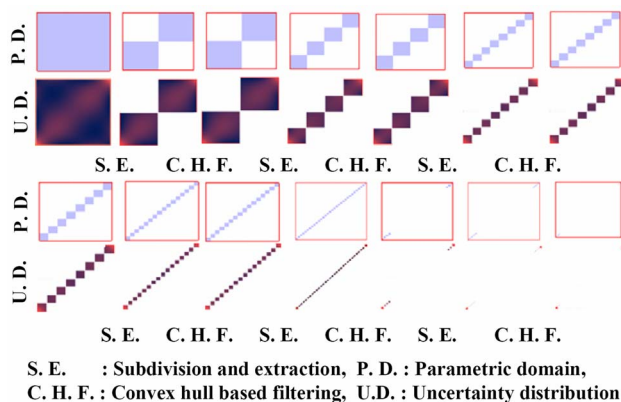


Fig. 5 An example of a NBP computing process

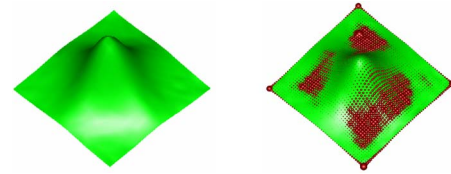


Fig. 6 A reconstructed surface and its uncertainty

$\leq u_1, v_1, u_2, v_2 \leq 1$ }, and  $\mathbf{A}(u_1, v_1)$  and  $\mathbf{A}(u_2, v_2)$  are the  $B$ -spline shape function vectors. The distribution of the uncertainty of the  $B$ -spline surface point  $p(u, v)$  forms a space surface on the  $\mathfrak{R}^4$  uncertainty surface  $S$  with  $u_1=u_2, v_1=v_2$ .

This higher-dimensional  $B$ -spline uncertainty surface also has the strong convex hull and subdivision properties. As such, the NBP search method on a curve can be directly extended to the NBP search on a reconstructed  $B$ -spline surface.

Consider the reconstructed surface over the parametric domain  $[0, 1] \times [0, 1]$  and its uncertainty shown in Fig. 6, the divide-and-conquer process is demonstrated in Fig. 7. Since the uncertainty surface for a  $B$ -spline surface is defined in  $\mathfrak{R}^4$ , only the patches with larger uncertainty that are not eliminated are shown.

**4.4 Summary of NBP Search.** The process of the NBP search method is summarized in Fig. 8. The input of an NBP search is a  $B$ -spline surface representation and its uncertainty covariance matrix from prior measurements. The output is the NBP. The search process involves several steps.

1. Transform the  $B$ -spline surface's uncertainty into a  $B$ -spline tensor form, referred to as a higher-dimensional *uncertainty surface*.
2. Subdivide the uncertainty surface into smaller patches and extract the individual subdivided  $B$ -spline patches.
3. Compare the convex hull of these patches and eliminate patches of which the largest uncertainty value is smaller than the smallest uncertainty value of another patch.
4. Terminate the search process if both the parametric interval length and the height of the uncertainty patches are smaller than some given tolerance values. If not, keep iterating Steps 2–4.

**4.5 Properties of the Individual NBP Search.** This section analyzes the global optimality and monotonous convergence of the individual NBP search.

*Property 1 (global optimality).* The point found by the NBP method is less than  $\rho_l$  (any prior given tolerance) away from the

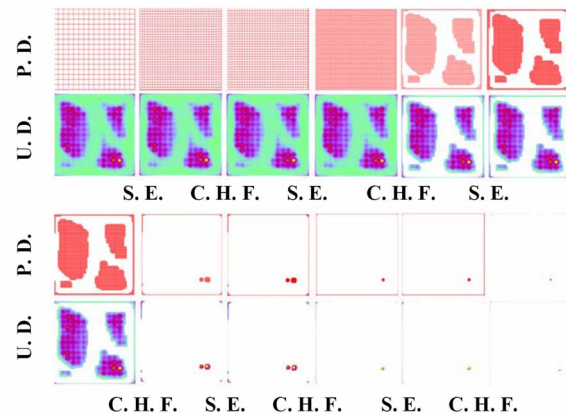


Fig. 7 NBP search process in a  $B$ -spline surface

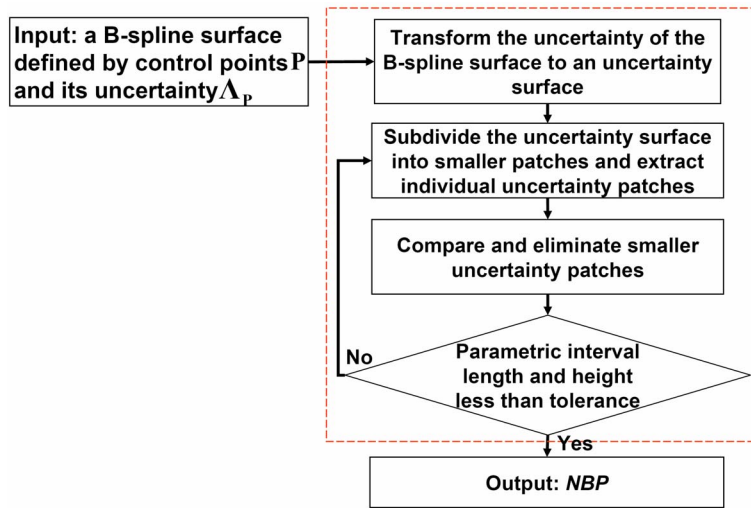


Fig. 8 Flowchart of NBP search

globally optimal point in the parametric domain, and less than  $\rho_c$  (any prior given turbulence) from the uncertainty value of the globally optimal point.

*Proof.* Denote the true globally optimal point as  $\Phi^*$  with uncertainty value  $\Lambda_{\Phi^*}$ , and the point found by the NBP method as  $\Phi$  with uncertainty value  $\Lambda_{\Phi}$ . We claim

$$|\Phi - \Phi^*| < \rho_l \text{ and } |\Lambda_{\Phi} - \Lambda_{\Phi^*}| < \rho_c \quad (21)$$

From the NBP search introduced in Secs. 4.2 and 4.3, we see that (1) the NBP method searches the entire parametric domain, and (2) the maximum patch size where the next point is located is below the user specified tolerance  $\rho_l$  and  $\rho_c$ . It means that the distance between the obtained NBP  $\Phi$  and the true point  $\Phi^*$  satisfies  $|\Phi - \Phi^*| < \rho_l$ , and the deviation between the uncertainties  $\Lambda_{\Phi}$  and  $\Lambda_{\Phi^*}$  of the true point  $\Phi^*$  satisfies  $|\Lambda_{\Phi} - \Lambda_{\Phi^*}| < \rho_c$ . Therefore, the found point is globally optimal both on the parametric domain and in the uncertainty value.

Such global optimality enables the maximal increase of  $\det(\mathbf{M}_k)$  for each NBP search and leads to fast convergence of localization of multiple sensing positions.

*Property II (guaranteed monotonic convergence).* The gradually decreasing parametric search interval and the subdivided uncertainty surface will converge to the true globally optimal point monotonically.

*Proof.* Let  $m$  be the number of subdivisions,  $l_m$  be the maximal interval length after  $m$  subdivisions, and  $h_m$  be the height between maximal uncertainty value and that of the found point along the uncertainty height axis.

The parametric domain for the NBP search is monotonically convergent and the interval length can be represented as  $l_m = 2^{-m}/(n_u - 3)$  where  $n_u$  is the number of control points in the  $u$  direction. A similar equation can be derived in the  $v$  direction. It is clear that  $\lim_{m \rightarrow \infty} l_m = 0$ .

Let  $\varphi_m$  be the convex hull of an uncertainty surface patch after the  $m$ th subdivision. From the  $B$ -spline subdivision property, we know that  $\varphi_m \subset \varphi_{m-1}$ . As such, we have  $0 < h_m < h_{m-1}$ , and  $\lim_{m \rightarrow \infty} h_m/h_0 = 0$ .

These properties hold true for surface patches because the control points of the subdivided patches are convex (and positive) combination of a subset of the original control points [21].

**4.6 NBP Based Sensing Localization.** The NBP method is immediately applicable to dynamic sensing localization where a single point needs to be determined to guide the subsequent sensing. This has led to a new data acquisition and surface reconstruction

methodology, called *dynamic sensing-and-modeling* [23].

This section thus focuses on the NBP method in offline sensing planning where multiple sensing locations need to be determined.

Given a  $B$ -spline surface structure and the number of data points  $k$  to be measured, we can determine multiple sensing locations as follows.

*Step 1.* Generate  $k$  design points on the domain and compute  $\Lambda_{\mathbf{p}_k}$  and  $\mathbf{M}_k$ .

*Step 2.* Withdraw the  $i$ th point ( $i=1, 2, \dots, k$ ) and update the surface uncertainty covariance and Fisher information matrices with Eqs. (10) and (11).

*Step 3.* Search for the next point with the NBP method.

*Step 4.* Update the surface uncertainty covariance and Fisher information matrices by adding the found point obtained from Step 3 with Eqs. (10) and (11).

*Step 5.* Compute the increase of determinant with Eq. (14).

*Step 6.* Repeat Steps 2–5 and record the maximal determinant increase of  $\mathbf{M}_k$  until all  $k$  design points are replaced.

*Step 7.* Repeat Steps 2–6 until the maximal determinant increase of  $\mathbf{M}_k$  is below a given tolerance.

The stopping criterion in Step 7 checks the maximal determinant increase of  $\mathbf{M}_k$  rather than its increase for each individual sensing location. This has been found to be effective in bypassing configurations of sensing locations in which individual sensing locations are close to the optimal locations and lead to very minor increases in  $\det(\mathbf{M}_k)$  while other sensing locations are far from their optimal locations.

## 5 Experimental Validation

Experimental examples are presented below to demonstrate the *efficiency* of the sensing localization method and the *accuracy* of the resulting sensing locations.

To validate the efficiency of the developed method, we compare it with other methods such as basic Fedorov exchange [18,19], simulated annealing [2], interchange [16], accelerated interchange [20], and a conjugate gradient method [5]. The accuracy comparison requires true global optimal locations. Since it is not easy to find the globally optimal solution, we use a hybrid method to produce the “true” optimal solution to validate the accuracy of our algorithm. We combine the developed divide-and-conquer and basic Fedorov exchange algorithm. All four examples are tested on a Dell Dimension DIM4700 with the dual Intel Pentium IV 2.8 GHz processors and 512 Mbyte RAM.

**5.1 Example 1: A Benchmark Example.** In Ref. [2], the

**Table 1 Experimental conditions for various methods**

Localization method	Experimental conditions
Basic Fedorov exchange	$N_c=1000, \rho_\Delta=0.01$
Simulate annealing	$q=1000, \rho_\Delta=0.01$
Interchange	$N_c=10,000, \rho_\Delta=0.0001$
Accelerated interchange	$N_c=10,000, \rho_\Delta=0.0001, q=180, a=0.9$
Conjugate gradient	$\rho_\Delta=0.0001$
Iterative NBP (INBP)	$\rho_l, \rho_c=0.0001, \rho_\Delta=0.0001$
INBP and then basic Fedorov exchange	$\rho_l, \rho_c=0.0001, \rho_\Delta=0.0001, N_c=1000$

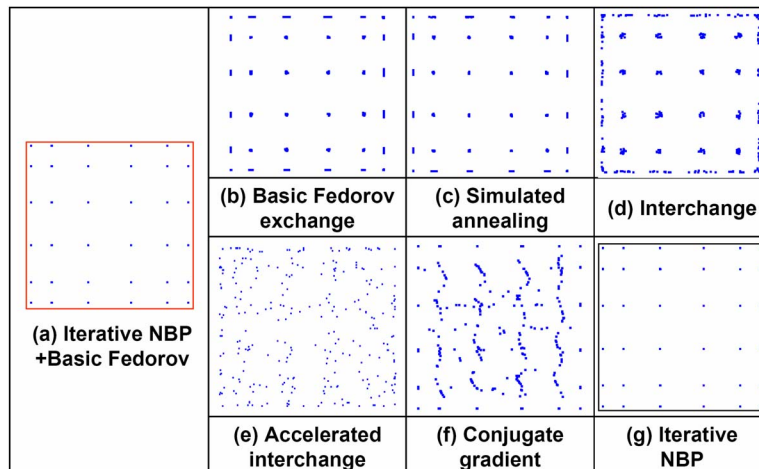
$B$ -spline surface model for a free-form surface is given as the uniform bicubic  $B$ -spline surface ( $6 \times 6$  control points) with  $k=360$  data points to be taken. We define the experimental conditions for various methods in Table 1. Note that the looser convergence condition  $\Delta \det(\mathbf{M}_k) \leq \rho_\Delta = 0.01$  is selected for the random search methods such as basic Fedorov exchange and simulated annealing methods because they converge very slowly. Their computational time is found to be still larger than the divide-and-conquer based method, which uses  $\Delta \det(\mathbf{M}_k) \leq \rho_\Delta = 0.0001$ .

In Table 1,  $N_c$  is the size of the candidate points from which each of the  $k$  sensing locations will be exchanged to identify one good sensing location,  $q$  is the number of exchanged points for each individual sensing location,  $a$  is the ratio of the remaining points after comparison, and  $\rho_l$  and  $\rho_c$  are the domain and uncertainty value tolerance. The resulting sensing locations of the 360 measurements are given in the parametric domain and they are shown in Fig. 9.

The computational time and the determinant of the Fisher information matrix  $\det(\mathbf{M}_k)$  are also compared in Table 2.

From Fig. 9 and Table 2, we can see the following.

- The basic Fedorov exchange and simulated annealing methods require longer time to converge and the results are not as good as the NBP method.
- As with the application of basic Fedorov exchange method on a discrete point-set domain, the interchange method also converges slowly, and deviates much from the best solution. However, the accelerate interchange converges faster than the basic method, but as not fast as the NBP method.
- Even though the conjugate gradient converges very fast, its solution is far from the optimal solution. This confirms that its result is seriously biased by the initial values.

**Fig. 9 Sensing locations on the parametric domain****Table 2  $\det(\mathbf{M}_k)$  and computational time**

Localization method	$\det(\mathbf{M}_k)$	Time (s)
Basic Fedorov exchange	-82.104	19,519.549
Simulated annealing	-81.891	1,695.609
Interchange	-86.160	9,561.078
Accelerated interchange	-92.2491	525.453
Conjugate gradient	-88.137	510.141
Iterative NBP	-81.498	344.954
Iterative NBP and then basic Fedorov exchange	-81.491	344.954+3.734

- The divide-and-conquer based NBP not only converges faster than all other methods but also gives the sensing distribution closest to the true optimal solution.
- The basic Fedorov exchange method converges rapidly if it starts from the distribution generated by our divide-and-conquer algorithm. It also confirms that the divide-and-conquer method produces the near-optimal result. Moreover, the benefits of the hybrid method do not justify the programming effort required to implement it.

Therefore, the convex hull based divide-and-conquer method is a very efficient and accurate method for sensing localization.

**5.2 Example 2: Sensing Localization for Turbine Blade Digitization.** In this example, we use the CMM touch probe to digitize the blade surface. The  $B$ -spline surface model is obtained based on the initial surface reconstruction from the data acquired through the Minolta Vivid 910 (Fig. 10).

In Fig. 10, a blade surface is first scanned with the Minolta scanner. We reconstructed a uniform  $B$ -spline surface, and finally obtained the  $B$ -spline surface with  $12 \times 11$  control points. Based on the obtained  $B$ -spline model structure, we determined  $12 \times 11$  sensing locations and took the  $12 \times 11$  measurements on the blade surface using a high-accuracy tactile point sensor. The results are shown in Table 3 and Fig. 11.

Again, the NBP method runs much faster than all other methods other than the gradient method. It also achieves better sensing locations than all other methods.

**5.3 Example 3: Sensing Localization in Dynamic Sensing.** This NBP based divide-and-conquer method can also facilitate online sensing localization as in dynamic sensing-and-modeling applications [23]. Since our method can deterministically find the

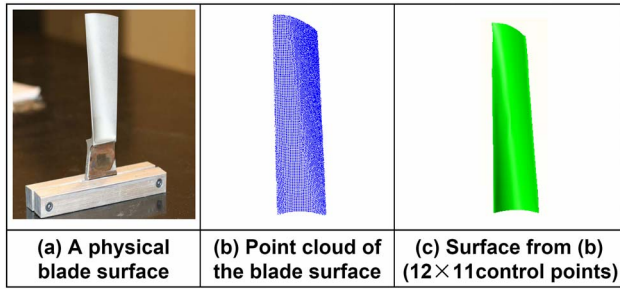


Fig. 10 Blade surface and estimated  $B$ -spline surface structure

Table 3  $\det(\mathbf{M}_k)$  and computational time

Localization method	$\det(\mathbf{M}_k)$	Time (s)
Basic Fedorov exchange	-435.40	4,549.36
Simulated annealing	-434.62	2,267.14
Interchange	-439.508	19,712.92
Accelerated interchange	-443.09	772.27
Conjugate gradient	-532.73	20.94
Iterative NBP	-433.64	685.39
Iterative NBP and then basic Fedorov exchange	-433.63	685.39+10.42

Table 4  $\det(\mathbf{M}_k)$  and computational time

Localization methods	$\det(\mathbf{M}_k)$	Computational time (s)
The Fedorov exchange	2366.8	48.04
The NBP	2368.3	19.28

individual optimal sensing location required in the dynamic sensing process, we only compare below our method against the basic Fedorov method.

In a simulated example in Fig. 12, dense but incomplete (the white area is the missing data area in Fig. 12(a) point-cloud data (102,063 points) can be rapidly scanned with data uncertainty  $\Lambda_x=\Lambda_y=\Lambda_z=0.01$ , and then the initial surface can be reconstructed. The reconstructed surface (Fig. 12(b)) and its uncertainty is shown (the red ellipsoids representing the surface uncertainty in (Fig. 12(c)) on a uniform bicubic  $B$ -spline surface ( $19 \times 19$  control points). The areas of larger uncertainty can also be seen, corresponding to the incomplete data areas where additional measurements are required.

With the developed sensing localization algorithm, additional 31 NBP points are sensed and the reconstructed surface is updated. With the same experiment conditions as shown in Table 1, the Fedorov exchange method and our NBP search method are compared in Table 4.

As seen in Table 4, the dynamic procedure for the NBP algorithm can lead to the optimized sensing locations more efficiently and achieve a lower uncertainty surface, when compared to the Fedorov exchange method.

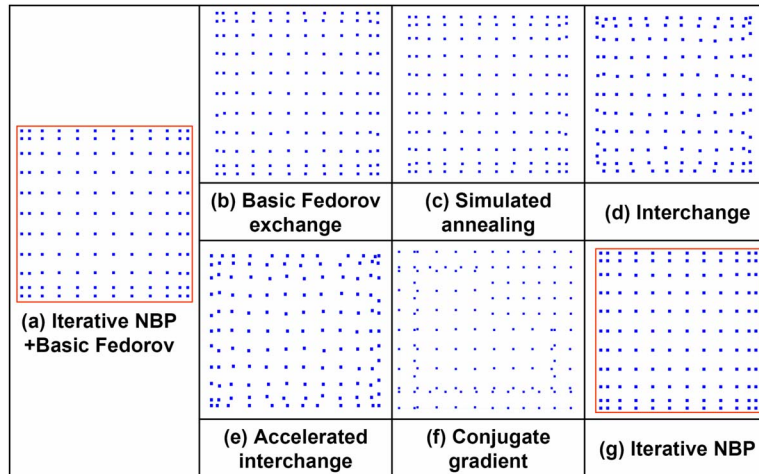


Fig. 11 Sensing locations on the parametric domain

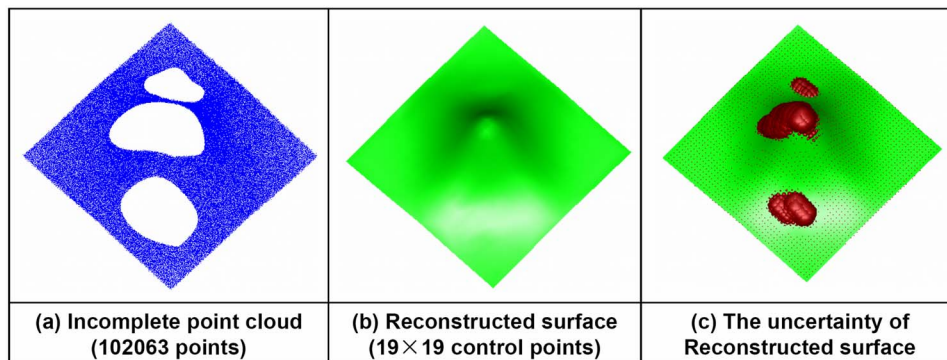


Fig. 12 Initial surface and uncertainty from incomplete points

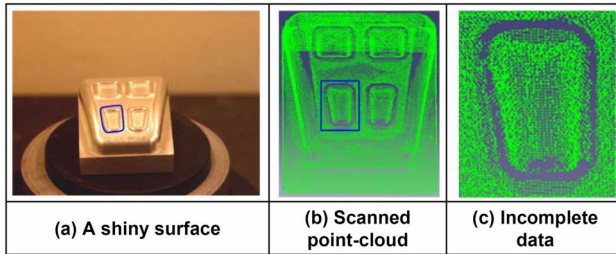


Fig. 13 Incomplete data in shiny surface through laser scanners

**5.4 Example 4: Sensing Localization for Shiny Surface Digitization.** A further example of using a touch probe to complement the optical scanner (Minolta) in a specular reflection area is shown in Fig. 13. Due to the specular reflection from the shiny surface part in Fig. 13(a) there are data missing in Fig. 13(c). A dynamic-sensing-and-modeling method was therefore developed in Ref. [23] to dynamically determine the NBP, then drive the touch probe to the computed NBP and acquire the data, and to update the surface model based on the acquired data. The process is repeated until it converges.

In the dynamic sensing-and-modeling approach, the divide-and-conquer based sensing localization yielded smaller surface uncertainty and a more accurate surface than that through basic Fedorov exchange with the same experimental condition shown in Table 1. Note that before the fusion of touch probe points, the initial surface from the scanner has  $\log \det(\mathbf{M}_k) = -6.8846 \times 10^3$ .

From Table 5 and Fig. 14, we can see that the uncertainty of the reconstructed surface through the NBP method is also lower than that through the basic Fedorov exchange method.

Table 5  $\log \det(\mathbf{M}_k)$  and computational time

Localization method	Time (s)	$\log \det(\mathbf{M}_k)$
Fedorov exchange	193.97	$-7.0547E+003$
The NBP	70.78	$-7.0717E+003$

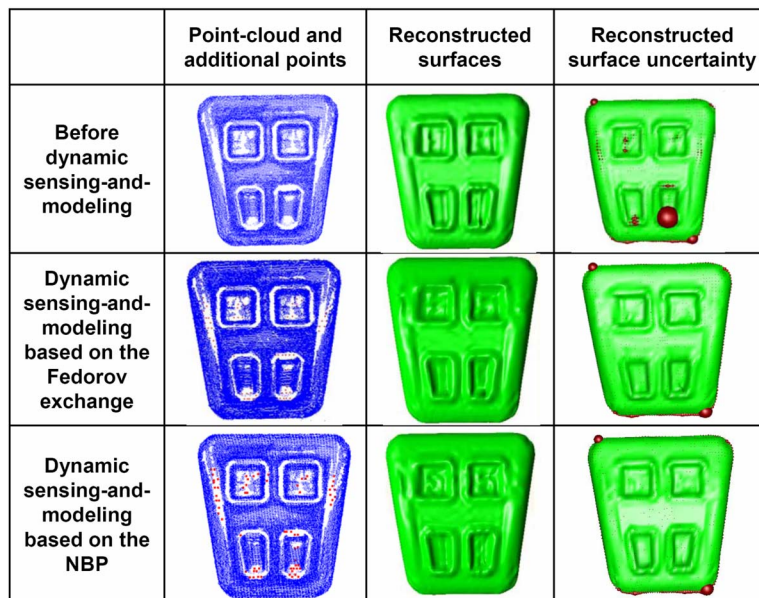


Fig. 14 Reconstructed surfaces through dynamic sensing-and-modeling

## 6 Algorithm Efficiency

In this section, for the above examples, we plot the ratio of the number of patches that cannot be discarded versus the total number of patches created after subdivisions in the best and worst NBP search in Fig. 15. Therefore, this chart characterizes the efficiency of the divide-and-conquer method at each subdivision. Note that in these four examples, each case involves many times of NBP search for multiple locations. The best case refers to the NBP search that converges faster than all other searches, i.e., one that has the fewest number of cumulative subdivided patches. The worse case refers to the NBP search with the longest search time.

As shown in Fig. 15, approximately more than 65% of the subdivided patches can be eliminated in the subdivision process for all 2198 ( $=360 \times 4 + 31 + 132 \times 5 + 67$ ) NBPs in four examples with control points ranging from  $6 \times 6$  to  $35 \times 35$ , which explains why the divide-and-conquer based sensing localization algorithm is *very efficient*.

To obtain NBP with the same level accuracy as the divide-and-conquer approach, we can also divide the whole domain into identical subareas on the parametric domain and find the same NBP by exhaustively comparing their uncertainty of each subarea (here, we refer to it as the exhaustive search method).

In Fig. 16,  $w$  represents the number of cumulative not-eliminated patches,  $s$  expresses the number of subdivided subareas on the domain, and then  $\log(w/s)$  shows the computation ratio of the divide-and-conquer method to the exhaustive search method in all four examples. From the computation ratio chart, we can see that  $(w/s)$  exponentially decreases with the increase of subdivision times in all four examples. It means that the convex hull of the uncertainty surface can be used to *effectively eliminate* low uncertainty surface patches on the parametric domain and is *exponentially more efficient* than the simple exhaustive search.

## 7 Conclusion

This paper presents an efficient sensing localization method for free-form surface digitization. The basic idea is to produce a desirable distribution of sensing locations by iteratively finding the next best sensing location. In each step, a divide-and-conquer strategy is introduced to efficiently find the NBP.

The theoretical contribution of this approach lies in the transformation of the  $B$ -spline surface's uncertainty distribution into a



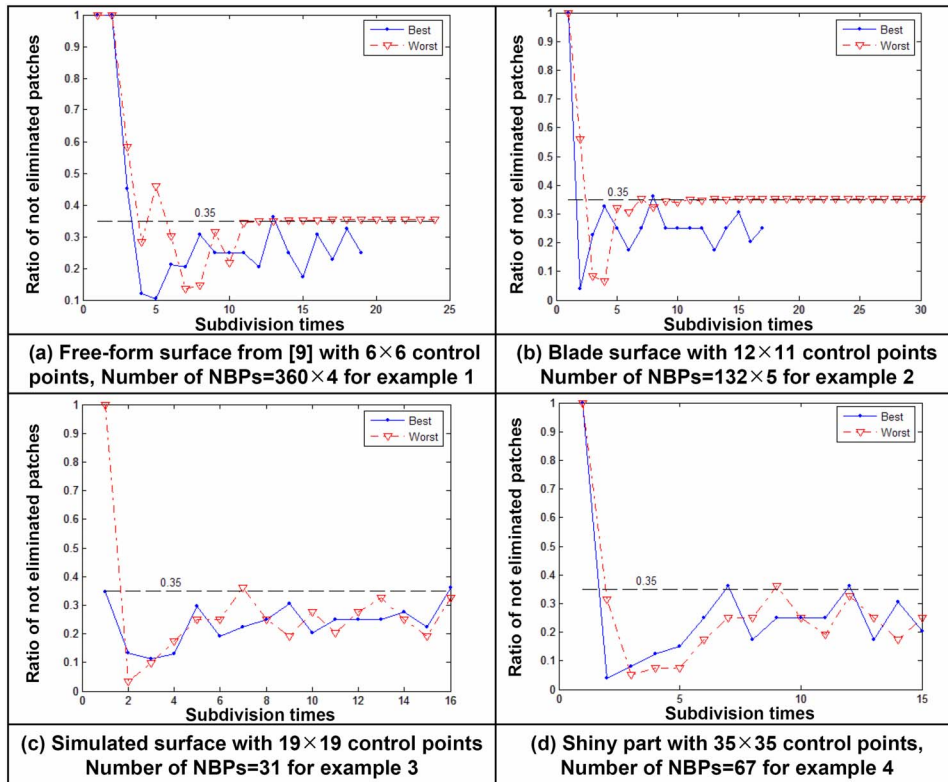


Fig. 15 Ratio of not-eliminated patches to the total subdivided patches

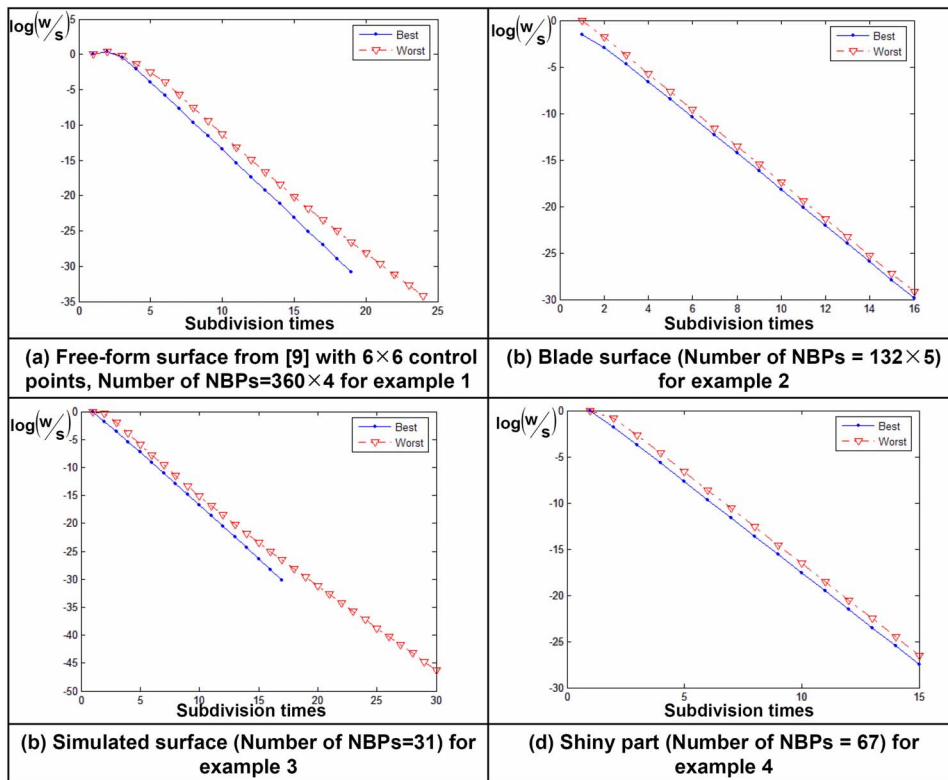


Fig. 16 Cumulative patches not eliminated at each subdivision level versus subdivision times

higher-dimensional uncertainty surface, expressed as a tensor  $B$ -spline form. As such, the convex hull and the subdivision properties of  $B$ -spline can be used for efficiently finding the NBP. Iterative use of NBP leads to a good distribution of overall sensing locations.

Experimental study demonstrates that the NBP method finds better sensing locations faster than many existing methods tested in this paper. The divide-and-conquer method for the NBP search converges exponentially faster than the simple exhaustive search. Due to the efficiency of the NBP search, it has been successfully used in both offline sensing planning and online dynamic sensing planning.

## Acknowledgment

This work was supported in part by the NSF Grant No. 0529165 and AFOSR Grant No. FA 9550-07-1-0241.

## References

- [1] Kim, P., and Ding, Y., 2004, "Optimal Design of Fixture Layout in Multistation Assembly Processes," *IEEE Trans. Autom. Sci. Eng.*, **1**(2), pp. 133–145.
- [2] Li, Y. F., and Liu, Z. G., 2003, "Method for Determining the Probing Points for Efficient Measurement and Reconstruction of Free-Form Surfaces," *Meas. Sci. Technol.*, **14**(8), pp. 1280–1288.
- [3] Wang, Y., and Pelinescu, D. M., 2001, "Optimizing Fixture Layout in a Point-Set Domain," *IEEE Trans. Rob. Autom.*, **17**(3), pp. 312–323.
- [4] Xiong, Z., Wang, M. Y., and Li, Z., 2004, "A Near-Optimal Probing Strategy for Workpiece Localization," *IEEE Trans. Rob. Autom.*, **20**(4), pp. 668–676.
- [5] Yau, H., 1998, "Uncertainty Analysis in Geometric Best Fit," *Int. J. Mach. Tools Manuf.*, **38**, pp. 1323–1342.
- [6] Konstantinos, A. T., Peter, K. A., and Roger, T., 1995, "A Survey of Sensor Planning in Computer Vision," *IEEE Trans. Rob. Autom.*, **11**(1), pp. 86–104.
- [7] Lee, K. H., and Park, H. P., 2000, "Automated Inspection Planning of Free-Form Shape Parts by Laser Scanning," *Rob. Comput.-Integr. Manufact.*, **16**(4), pp. 201–210.
- [8] Pito, R., 1999, "A Solution to the Next Nest View Problem for Automated Surface Acquisition," *IEEE Trans. Pattern Anal. Mach. Intell.*, **21**(10), pp. 1016–1030.
- [9] Qian, X., and Harding, K. G., 2003, "A Computational Approach for Optimal Sensor Setup," *Opt. Eng. (Bellingham)*, **42**(5), pp. 1238–1248.
- [10] Richard, P., and Ruzena, B., 1995, "A Solution to the Next Best View Problem for Automated CAD Model Acquisition of Free-Form Objects Using Range Cameras," *Proc. SPIE* **2598**, pp. 78–89.
- [11] Sheng, W., Xi, N., Song, M., Chen, Y., and Rankin, J. S., 2000, "Automated CAD-Guided Automobile Part Dimensional Inspection," *Proc. IEEE International Conference of Robotics and Automation*, Vol. 2, pp. 1157–1162.
- [12] Spyridi, A. J., 1994, "Automatic Generation of High Level Inspection Plans for Coordinate Measuring Machines," Ph.D. dissertation, University of Southern California, Los Angeles, CA.
- [13] Elkott, D. F., Elmaraghy, H. A., and Elmaraghy, W. H., 2002, "Automatic Sampling for CMM Inspection Planning of Free-Form Surfaces," *Int. J. Prod. Res.*, **40**(11), pp. 2653–2676.
- [14] Whaite, P., and Ferrie, F. P., 1997, "Autonomous Exploration: Driven by Uncertainty," *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**(3), pp. 193–205.
- [15] Li, Y. F., and Liu, Z. G., 2003, "Uncertainty-Driven Viewpoint Planning for 3D Object Measurements," *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 127–132.
- [16] Wang, Y., and Nagarkar, S., 1997, "Optimal Sensor Location Design in Automated Coordinate Checking Fixtures," in the *Proceedings of IEEE International Symposium on Assembly and Task Planning (ISATP'97)*, pp. 140–145.
- [17] Xiong, Z., Wang, M. Y., and Li, Z., 2003, "A Computer-Aided Probing Strategy for Workpiece Localization," in *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, Vol. 3, pp. 3941–3946.
- [18] Fedorov, V. V., 1972, *Theory of Optimal Experiments*, W. Studden and E. M. Klimbo, eds., Academic, New York.
- [19] Nguyen, N. K., 1992, "A Review of Some Exchange Algorithms for Constructing Discrete D-optimal Designs," *Comput. Stat. Data Anal.*, **14**(4), pp. 489–498.
- [20] Liu, C., Ding, Y., and Chen, Y., 2005, "Optimal Coordinate Sensor Placements for Estimating Mean and Variance Components of Variation Sources," *IIE Trans.*, **37**, pp. 877–889.
- [21] Piegl, L., and Tiller, W., 1995, *The NURBS Book*, Springer, New York.
- [22] Kullback, S., 1968, *Information Theory and Statistics*, Dover, New York.
- [23] Huang, Y., and Qian, X., 2007, "A Dynamic Sensing-and-Modeling Approach to 3D Point- and Area-Sensor Integration," *ASME J. Manuf. Sci. Eng.*, **129**, pp. 623–635.