



Designing B-Spline Surfaces with Haptics Based on Variational Technique

Xiaodong Liu

Hebei University of Technology, xdg.liu@gmail.com

ABSTRACT

Due to their 3D nature, many CAD B-Spline surfaces are difficult to design and edit with traditional 2D interfaces, e.g., a 2D mouse. Based on an energy minimization method or variational B-Spline technique, the paper presents the technique of using a haptic device to design B-Spline surfaces. Under complex constraints and without any pre-calculation, the variational B-Spline technique used in the system can solve virtually all practical B-Spline surfaces in a matter of milliseconds. Such results are much more efficient and powerful than previous work in the variational B-Spline field. By using a haptic interface, the system allows a user in a natural 3D environment to directly manipulate/design constraints, e.g., points and curves, and in real time, the system generates energy-minimization B-Spline surfaces in response to haptic operations. The discussed technique delivers all the performance and capacity required for using haptics to design high-quality B-Spline surfaces, and haptic-based surface operations provide a powerful method for designing B-Spline surfaces.

Keywords: haptics, B-Spline, VR, variational B-spline technique, energy minimization.

1. INTRODUCTION

B-Spline is the de-facto industry standard for curves and surfaces in CAD. With traditional 2D interfaces, e.g., a 2D mouse, many CAD B-Spline Surfaces are difficult to design and edit due to their 3D nature. Haptics, which provides designers with a tactile feedback to interact with digital models in 3D, has great promise for use in CAD. Based on an energy minimization method or variational B-Spline technique, we have developed a haptic-based CAD system - DesignWorks, which uses a haptic device to design B-Spline surfaces. Photos in Fig. 1 show DesignWorks in operation. In these photos, the haptic cursor is shown as a green sphere.

1.1. Main Contributions

1) The paper presents the technique for using a haptic device to design B-Spline surfaces based on a variational B-Spline technique. Compared with the previous work, the haptic-based operations discussed in this paper can generate more plausible surfaces, and handle more complex B-Spline models. They provide a powerful method for designing B-Spline surfaces with haptics.

2) The paper presents the framework and addresses the key issues on the variational B-Spline technique, including: I) New forms of constraints for B-Spline surfaces. For example, the normal direction curve constraints, which are superior to the traditional C1 constraints and basically replace the traditional C1 constraints for tangent continuities. Both point-based and curve-based constraints will be discussed; II) New energy form for B-Spline surfaces. It is primarily based on the rate of change in bending and generates noticeably smoother shapes compared to other quadratic B-Spline energy forms. The default shape can be specified so that the generated B-Spline surface has the minimum energy while conforming as closely as possible to the specified default shape; III) The complete elimination of constraint equations by energy-based items. This method is accurate, efficient and robust. Our work shows that hard constraints (or the "Lagrange" method) should not be used for variational B-Spline technique in practice. ("accurate" means: In CAD systems, numerical tolerances or deviations are inevitable, and all CAD calculations are performed with specified tolerances. CAD systems have high-precision requirements. Simply speaking, for B-Spline, the tolerance of 1.0E-6 meter is considered to be a good or accurate result to build waterproof models in CAD systems.) IV) By bringing





Fig. 1: DesignWorks in operation.

these together, we demonstrate that, without any pre-calculation, our work can solve a B-Spline surface with up to 20,000 control points in real time (refer to Section 3.5 for the detailed performance) and, at the same time, produce high quality B-Spline surfaces satisfying an arbitrary combination of point and curve constraints. To our knowledge, it is the only system that could achieve such a performance and capacity, and such a performance and capacity is literally hundreds (even thousands) of times more powerful and efficient than previous work in the variational B-Spline field.

The discussions in this paper are in the context of B-Spline surfaces, and may not be applicable to other surface representations. The remainder of the paper is organized as follows: Section 2 reviews the research in the related fields. Section 3 discusses the technique for designing B-Spline surfaces with haptic interactions based on the variational B-Spline technique. Section 4 conveys our conclusions.

2. REVIEW OF RELATED RESEARCH

The work discussed in this paper mainly involves two areas: the variational B-Spline technique (or sometimes called the energy minimization method) and haptics. This section reviews related research in these two areas.

2.1. Haptics

In the conceptual design field, Virtual Reality (VR) devices can provide a faster sketching method [6,7]. Recent progress in haptic devices has promoted the use of force feedback in VR applications [1,14]. Evans et al. [3] and Volkov et al. [18] perform studies to evaluate how a haptic device affects the ability of a person for designing or making design decisions. The results indicate that the addition of a haptic interface can expedite the design process and allows users to complete a task in significantly less time.

In the design related fields, haptic technique has been primarily used in three major categories: virtual assembly simulations [11], the creation of polygon/voxel-based models [4,17], B-Spline based manipulation [5,8,9]. In our opinion, directly manipulating/designing B-Spline surfaces is needed when using haptics to design surfaces in CAD because: (1) B-Spline is the standard representation for curves/surfaces in CAD; (2) as stated in [5,8], changes made to other representations, e.g., a polygon/voxel-based representation, are very difficult, if not impossible, to be converted back into a B-Rep model in a CAD system.

Compared with the previous work of using haptics for designing B-Spline [9], which adopted a totally different method, i.e., geometrically-based manipulation, the haptic-based operations discussed in this paper can obviously generate more plausible curves/surfaces, and handle more complex B-Spline models. The method discussed in this paper produces fair or plausible B-Spline surfaces that are analogous to a physical model: as shown in Fig. 2, when a wooden batten bends, it always assumes a shape that minimizes its bending energy to form a smooth curve. Similarly, the energy-minimization technique generates a B-Spline curve or surface that is the one among all possible solutions that has the minimum energy (or the smoothest curve or surface) and satisfies all the specified constraints. The method can solve virtually any B-Spline models regardless of the complexity of the constraints because the energy-minimization technique has the inherent capacity to find the “best” solution under extremely complex constraints. While variational B-Spline techniques can produce plausible 3D shapes and are a superior method for B-Spline modeling, due to its performance issue discussed in Section 2.2, few papers have explored the potential for using haptics to design CAD surfaces based on the variational B-Spline technique. Our research enables us to explore the use of the variational B-Spline technique for haptic-based surfacing design, and the technique provides all the capacity and performance needed for using haptics in designing B-Spline.



Fig. 2: Bending a batten.

2.2. Variational B-spline Technique

Variational techniques generate fair curves/surfaces by minimizing curve/surface energy. The literature in the field is vast, and covers almost every surface representation, e.g., subdivision, implicit surfaces, B-Spline and triangle meshes [2,13,20]. In this paper, we limit our discussions to the variational technique for B-Spline, in which we claim our contributions. Typically, variational B-Spline technique is expressed as [2,13,20], which pioneered the research in the variational B-Spline field:

$$\text{Minimize Energy } E(\text{Surf}) \quad (1)$$

$$\text{Which is subject to Constraints } CP = G \quad (2)$$

For linear-based methods, the surface energy $E(\text{Surf})$ is generally simplified by [2]:

$$E(\text{Surf}) = \iint (S_{uu}^2 + 2S_{uv}^2 + S_{vv}^2) dudv$$

where S represents a surface, u and v parameterize the surface, S_{uu} is a shorthand for $\partial^2 S / \partial u^2$, and S_{uv} is a shorthand for $\partial^2 S / \partial u \partial v$. As discussed in [2,16], soft constraints or penalty method only weakly encourage meeting the constraints, so to enforce constraints to meet high-precision requirements, the constraint equations in Eqn. (2) are used. This method is also

called the “hard-constraint” or “Lagrange” method (since it is typically solved by the Lagrange multiplier). While the “hard-constraint” or “Lagrange” technique is a superior technique for B-Spline modeling, its calculation is time-consuming and numerically unstable: As Welch et al. [20] state, “a u, w curve with very slight high-degree oscillations will give rise to nearly-dependent constraint rows for an interpolation constraint”, so the constraint equations in Eqn. (2) are numerically very close to singular and its calculation is time-consuming. Michalik et al. [12] state that “for a surface with 500 DOF (i.e., about 500 control points) and more than a thousand constraint equations (i.e., typically, several curve constraints), the SVD step alone (for handling Eqn. (2)) already requires 15–20 seconds”, and “the run-time grows rapidly for larger examples”.

As highlighted in Section 1.1, next section will discuss the framework and the key issues of the variational B-Spline technique used in DesignWorks. The variational B-Spline technique discussed in this paper has been implemented in VBS kernel [10,19]. The technique can generate energy-minimization B-Spline surfaces in response to haptic operations in real time. To confirm the claimed performance, we produce a live demo, and both the demo file and the generated B-Spline surface can be downloaded from [19]. The demo “deforms” (more precisely speaking, it solves the B-Spline surface without any pre-calculations) a B-Spline surface with $140 \times 140 = 19,600$ control points in real-time under one-point and six-curve constraints, and it is generated from a computer with an Intel i5-760 2.8 GHz processor and 4 GB RAM (Section 3.5 will discuss the performance benchmarks). Our work also shows that, to enforce constraints to meet high-precision requirements, the widely used “hard-constraint” or “Lagrange” method is not needed, and the “Lagrange” method should not be used for variational B-Spline technique in practice. Fig. 3 illustrates some waterproof surfaces created by

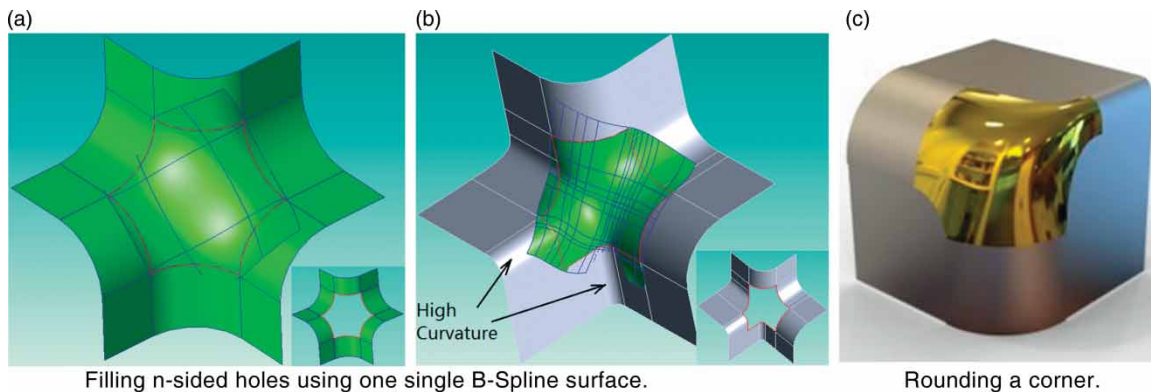


Fig. 3: Waterproof surfaces created by VBS kernel: (a)/(b) each boundary consists of n curves. VBS kernel uses one single rectangular B-Spline surface to fill the hole. The tolerance requirement is less than $1.0E-6$ meter. (The hole is shown as a small picture at the bottom right of (a) and (b). To demonstrate that only one single B-Spline surface is needed, the entire un-trimmed rectangular B-Spline surface is shown.); (c) the golden surface is a B-Spline surface created by VSB kernel, i.e., after patching the hole, the generated B-Spline surface is trimmed.

VBS kernel: high-quality B-Spline surfaces are generated by satisfying complex boundary constraints, and a 1.0E-6 meter is used as the tolerance requirement.

3. HAPTIC OPERATIONS FOR DESIGNING B-SPLINE BASED ON VARIATIONAL B-SPLINE TECHNIQUE

3.1. Haptic Operations for Designing B-Spline

In our system, a haptic interface is used to directly manipulate/design constraints, e.g., points or curves, and the system uses the variational B-Spline technique to generate the “smoothest” B-Spline surface that satisfies the specified constraints in real time. Design-Works supports two types of operations for using haptics to design B-Spline surfaces: a point-based manipulation and a curve-based manipulation. From a designer’s perspective, the two types of haptic-based operations are similar: a designer touches a position, e.g., an arbitrary position on a surface or on a curve, clicks the haptic button down, moves the haptic cursor, and then releases the haptic button. During the operations, the system generates the energy minimization B-Spline surface in real time in response to haptic operations and the specified constraints, and the surface is solved fast enough in response to the movement of the haptic stylus as if the surface is being “deformed”. The tactile and force feedbacks are

used in these haptic-based operations, which will be further discussed in Section 3.5.

Fig. 4 (a) and (b) demonstrate the point-based manipulation using haptics. The cyan sphere, i.e., the haptic cursor, is the tip position of the haptic stylus, and is the point constraint being designed/manipulated by the haptic operation. In Fig. 4 (a), the surface is required to interpolate a point constraint (shown as a cyan sphere) and two short straight edges. In Fig. 4 (b), the surface is created by interpolating a point constraint (shown as a cyan sphere) and 6 curve constraints (i.e., the four edges of the top face and two green curves). The system generates the energy minimization B-Spline surface in real time in response to the movement of the haptic cursor. Fig. 4 (c) and (d) demonstrate the curve-based manipulation using haptics, and the surface is created by interpolating the green curve and the four edges of the top face. The green curve is the curve constraint being designed/manipulated by the haptic operation. Based on the movement of the haptic cursor (shown as a cyan sphere), a new energy minimization B-Spline curve is generated (shown as a green line), and the system generates the energy minimization B-Spline surface satisfying these curve constraints in real time as if the surface is being “deformed”. The energy minimization or variational B-Spline curves are very similar to the B-Spline surface discussed in this paper, but they are much simpler.

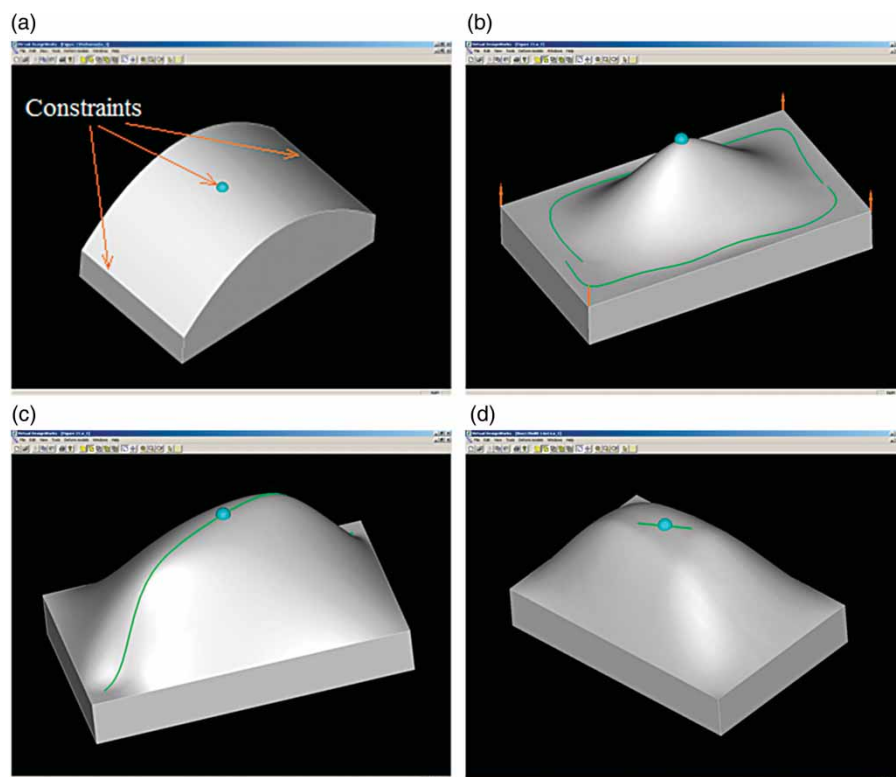


Fig. 4: (a)/(b) Surfaces created by the point-based manipulation using haptics; (c)/(d) Surfaces created by the curve-based manipulation using haptics.

Variational B-Spline curves are useful for the construction of curve constraints. Because of its simplicity and the space limitation of the paper, we will focus on the haptic-based operations for B-Spline surfaces.

DesignWorks adopts a dual model representation similar to [8], i.e., it maintains two independent models: a CAD geometry model (B-Rep) and a haptic geometry model (polygon-based). In our opinion, to use a haptic interface for the design of CAD models, a dual model representation is necessary to integrate CAD with haptics. With the dual model representation, operating and updating a CAD model with a haptic-based interface is similar to operating and updating a CAD model with a mouse-based interface. With a mouse interface, a user picks and works on graphics (a polygon model) on a screen and performs the actual operations on the CAD model. Then the graphics are updated according to the modified CAD model in real time. With a haptic interface, a user touches and feels a haptic geometric model (a polygon model), and performs the actual operations on the native CAD model. The haptic model is then updated according to the modified CAD model.

The algorithms used in haptic operations are involved three steps:

1. OnHapticButtonDown: This function intercepts and handles the event when the haptic button is pressed down. During this even, the system sets up a B-Spline surface and solves it on the fly by using the technique discussed in Section 3.2–3.4. The performance benchmarks are discussed in Section 3.5.
2. OnHapticMove: This function intercepts and handles the event when the haptic button is pressed down and the haptic stylus is moving. During this event, the system updates the B-Spline surface in response to haptic movements.
3. OnHapticButtonUp: This function intercepts and handles the event when the haptic button is released. During this event, by using an iterative method similar to [10], DesignWorks refines the generated surface to meet specified requirements, e.g., a tolerance of 1.0E-6 meter.

The technique discussed in this paper is fast enough to process the haptic operations in real time, and there is no need for any pre-calculations. The remainder of this section presents the framework and addresses the key issues on the variational B-Spline technique used for the haptic operations above, i.e., the point-based manipulation and curve-based manipulation. It is organized as follows: Section 3.2 discusses the framework of the variational B-Spline technique. Section 3.3 presents the B-Spline surface energy functional and the default shape will be considered. Section 3.4 addresses the energy-based constraints, and both point-based and

curve-based constraints will be discussed. Section 3.5 discusses the performance benchmarks, the mathematical details of minimizing the energy functional, and the advantages of using haptics for surface design. Finally, Section 3.6 discusses future work.

3.2. Framework of the Variational B-spline Technique

A B-Spline surface $S(u, v)$ is represented as follows [15]:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{ij}$$

Here, p and q are the degrees of the surface. \mathbf{P}_{ij} s are the $(n+1) \times (m+1)$ grid of control points.

The functional framework of a variational B-Spline surface E_v can be expressed as:

$$E_v = E_{Surf} + E_{Cons} \quad (3)$$

Where E_{Surf} is the B-Spline surface energy, and E_{Cons} is the constraint-based energy. By minimizing the energy functional in Eqn. 3, the solution, i.e., the optimized control points of a B-Spline surface, is generated. It should be noted that, different from the widely used “hard-constraint” or “Lagrange” method for enforcing constraints to meet high-precision requirements, Eqn. 3 expresses all constraints as some sort of energy E_{Cons} , and there is no constraint equations as shown in Eqn. (2). Our research shows the method is very accurate and can easily satisfy CAD systems’ high-precision requirements [10]: VBS kernel reveals that, when there are enough degrees of freedom to precisely satisfy a constraint (this is generally the case for a point constraint), the tolerance or deviation between the generated surface and the constraint is less than 1.0E-12 meter, and when there is not enough degrees of freedom to precisely satisfy a constraint (this is a typical case for a curve constraint), a B-Spline surface with a couple of hundreds of control points is generally enough to satisfy complex constraints with a less than 1.0E-6 meter tolerance. As stated in Section 1, CAD systems have high-precision requirements, and for B-Spline, the tolerance of 1.0E-6 meter is considered to be accurate to build waterproof models in CAD systems. The results mean that, to enforce constraints to meet high-precision requirements, the widely used “hard-constraint” or “Lagrange” method is not needed. In Section 3.5, we will discuss the performance issue, which shows that the system in Eqn. (3) is more efficient and robust to solve. Based on these discussions, it is clear that the “hard-constraint” or “Lagrange” method should not be used in practice for variational B-Spline technique.

3.3. B-spline Surface Energy Functional E_{Surf}

Our system uses a new energy form in the context of B-Spline surfaces, which is primarily based on

the rate of change in bending. It generates noticeably smoother shapes compared to other quadratic B-Spline energy forms, e.g., the widely used quadratic bending energy for B-Spline surfaces. The default shape can be specified as the energy minimization surface for haptic operations.

The B-Spline surface energy functional E_{Surf} in Eqn. 3 is defined as follows:

$$E_{Surf} = \alpha E_{roc\ in\ bending} + \beta E_{bending} + \gamma E_{stretching} \quad (4)$$

Where $E_{roc\ in\ bending}$ defines a B-Spline surface's resistance to the rate of change in bending. $E_{bending}$ and $E_{stretching}$ represent a B-Spline surface's resistance to bending and stretching. α, β and γ are the weights used to model the internal energy of the B-Spline surface. $E_{roc\ in\ bending}$, $E_{bending}$ and $E_{stretching}$ can be expressed as

$$E_{roc\ in\ bending} = \iint (\Delta S_{uuu}^2 + 3\Delta S_{uuv}^2 + 3\Delta S_{uvv}^2 + \Delta S_{vvv}^2) dudv \quad (5)$$

$$E_{bending} = \iint (\Delta S_{uu}^2 + 2\Delta S_{uv}^2 + \Delta S_{vv}^2) dudv \quad (6)$$

$$E_{stretching} = \iint (\Delta S_u^2 + \Delta S_v^2) dudv \quad (7)$$

Here, ΔS_u , ΔS_v , ΔS_{uu} , ΔS_{vv} and ΔS_{uuu} etc. are the 1st, 2nd, and 3rd partial derivative differences of a B-Spline surface between the generated surface and the original surface, e.g., $\Delta S_u = S_{u_generated} - S_{u_original}$. The delta energy, i.e. ΔS_{vv} and ΔS_{uuu} etc., can generate fair surfaces that have the minimum energy while conforming as closely as possible to the specified default shape. Naturally, a user expects this behavior when touching and deforming a surface with haptics.

These B-Spline energy items have quadratic forms, thus are numerically efficient to solve. The new B-Spline energy form used in our system is mainly based on $E_{roc\ in\ bending}$, i.e., $\alpha = 0.7$, $\beta = 0.3$, $\gamma = 0$ (refer to the following discussions). $E_{bending}$ and $E_{stretching}$ are the B-Spline energy functional widely used previously, e.g., in [2] and [20]. Generally speaking, $E_{roc\ in\ bending}$ generates surfaces with noticeably smoother B-Spline shapes compared to the surfaces created by other B-Spline energy items, i.e., $E_{bending}$ and $E_{stretching}$. B-Spline surfaces with large α values (i.e., $E_{roc\ in\ bending}$) attempt to distribute the curvature changes over large areas and thus generate very fair shapes. The B-Spline surface energy functional $E_{stretching}$, on the other hand, tends to minimize the surface area, and generally produces B-Spline surfaces with a sharp peak or turn. Thus, we typically set the $E_{stretching}$ weight as zero.

Fig. 5 illustrates comparisons between B-Spline surfaces created with different energy functions. Again, the haptic cursor is shown as a cyan sphere. The constraints in Fig. 5 (a) and (b) are similar to the

constraints in Fig. 4 (a). In Fig. 5 (a), the weights in Eqn. (4) are set as $\alpha = 1$, $\beta = 0$, $\gamma = 0$. In Fig. 5 (b), the weights are set as $\alpha = 0$, $\beta = 1$, $\gamma = 0$. The B-Spline surface energy functional $E_{roc\ in\ bending}$ generates a smooth surface (a cylinder surface), whereas $E_{bending}$ generates a surface with a bulge. Fig. 5 (c) and (d) compares the surfaces created by a curve constraint (shown as a green curve) and four boundary edges. The weights in Fig. 5 (c) are set as $\alpha = 1$, $\beta = 0$, $\gamma = 0$, whereas the weights in Fig. 5 (d) are set as $\alpha = 0$, $\beta = 0.4$, $\gamma = 0.6$. These examples illustrate that the B-Spline surface energy functional $E_{roc\ in\ bending}$ generates a surface with noticeably smoother shapes compared to other B-Spline quadratic energy items, i.e., $E_{bending}$ and $E_{stretching}$. It tends to distribute the change of bending over a large area. (The surface in Fig. 4 (a) is also created with $E_{roc\ in\ bending}$.) However, under rare cases for some applications, $E_{roc\ in\ bending}$ alone could generate undesired B-Spline shapes. Combining a small portion of $E_{bending}$ with $E_{roc\ in\ bending}$ works reliably. In general, the following weights are used for haptic operations: $\alpha = 0.7$, $\beta = 0.3$, $\gamma = 0$, i.e., our system uses this energy combination for the surfaces generated by haptic operations (i.e., OnHapticButtonDown, OnHapticMove and OnHapticButtonUp). We find it is a good combination of B-Spline surface energies. Except for Fig. 4 (a) and Fig. 5, all other surfaces in this paper are generated using this combination of B-Spline surface energies.

3.4. Energy-based Constraints

The energy-based constraints are used to completely replace the widely used constraints in Eqn. (2), and as discussed in Section 3.2, they can enforce constraints to meet high-precision requirements. These constraints include curve and point constraints, which correspond to the point-based and curve-based haptic manipulations. This subsection discusses the energy-based constraints, and it also presents new forms of constraint energies for B-Spline surfaces, e.g., energy-based normal direction curve constraints.

Energy-based constraints E_{Cons} in Eqn. (3) can be expressed as

$$E_{Cons} = E_{Position}^{Pt} + E_{C1}^{Pt} + E_{C2}^{Pt} + E_{Normal}^{Pt} + E_{Position}^{Cur} + E_{Normal}^{Cur} \quad (8)$$

where $E_{Position}^{Pt}$ and $E_{Position}^{Cur}$ etc. are the corresponding energy items for point and curve constraints. For example, $E_{Position}^{Cur}$ is the energy item for a positional curve constraint, E_{Normal}^{Cur} specifies the normal direction of a surface along the target curve, and $E_{Position}^{Pt}$ is the energy item for a positional point constraint. The following subsections, i.e., Subsection (a) - (e), will discuss each item.

(a) Energy-based positional point constraints $E_{Position}^{Pt}$

A positional point constraint forces a surface to interpolate a given position, which is illustrated in Fig. 4 (a) and (b) as the cyan sphere (i.e., the point-based haptic manipulation). As discussed in Section 3.2, this method can enforce a point constraint with a tolerance of less than 1.0E-12 meter. Energy-based positional point constraints can be expressed as

$$E_{Position}^{Pt} = K (S(u, v) - P_{Target})^2 \quad (9)$$

where K is the weight, P_{Target} is the given position to be interpolated, and $S(u, v)$ is the surface position at Parameter (u, v) . In Eqn. (9), like the surface energy items, the control points of the B-Spline surface are the unknown variables needed to be optimized, and all other parameters, e.g., the knot vectors and Parameter (u, v) , are given or fixed. Since the constraint energy is directly added into the B-Spline surface energy Eqn. (4), so no extra effort is needed for solving it. The system generates the optimized B-Spline surface by minimizing the system energy (refer to Section 3.5). The process of handling other energy-based constraints is similar to $E_{Position}^{Pt}$.

(b) Energy-based positional curve constraints $E_{Position}^{Cur}$

A positional curve constraint forces a surface to interpolate given curves, which is illustrated as the boundary curves in Fig. 3 or as the haptically manipulated green curve and four boundary curves in Fig. 5

(c). Energy-based positional curve constraints can be expressed as

$$E_{Position}^{Cur} = K \int (C_{CurOnSurf}(t) - C_{Target}(t))^2 dt \quad (10)$$

where K is the weight, $C_{Target}(t)$ is the given curve to be interpolated, and can be of any form, e.g., a line, an arc, or a b-spline curve. $C_{CurOnSurf}(t)$ is represented as an embedded curve on a surface, which is defined in the u, v parameter domain of the surface [15]. $C_{CurOnSurf}(t)$ can be expressed as

$$C_{CurOnSurf}(t) = C(u(t), v(t)) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u(t)) N_{j,p}(v(t)) P_{i,j} \quad (11)$$

Where $u(t)$ and $v(t)$ are u, v components of the embedded curve in the u, v parametric domain. To determine the coordinates of a point on an embedded curve in the 3D space, one first computes the pair of $(u(t), v(t))$ values for a specified t , then substitutes these (u, v) values into the Eqn. (11). As discussed in Section 3.2, this method can enforce a curve constraint with a tolerance of less than 1.0E-6 meter. The mathematical details of deriving the linear system to minimize the energy functional are discussed in Section 3.5.

(c) Energy-based normal direction point constraints E_{Normal}^{Pt}

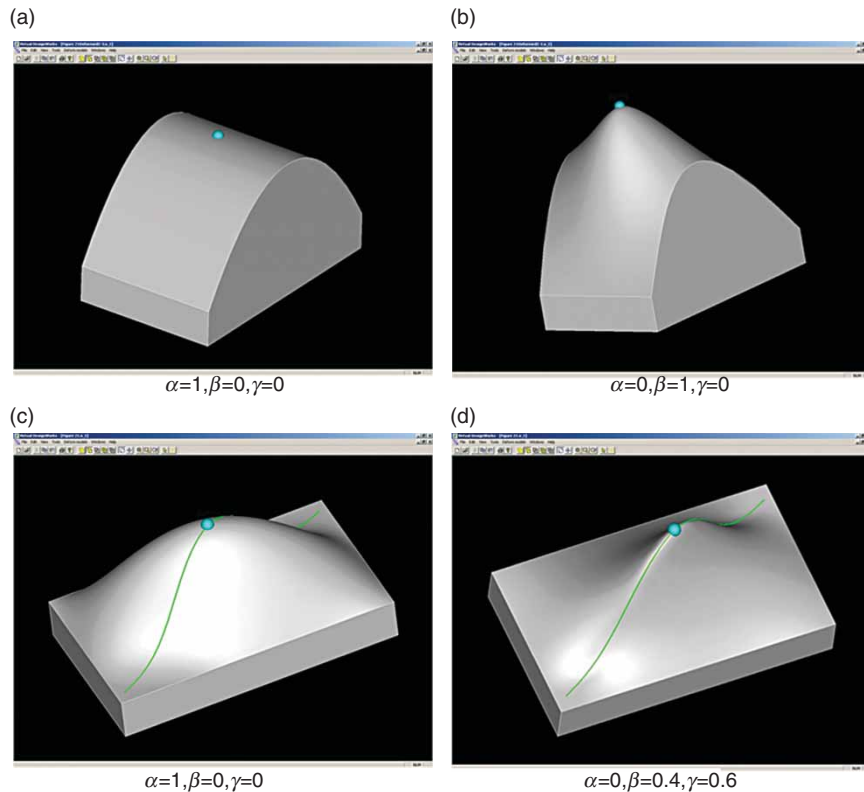


Fig. 5: Comparing surfaces created with different surface energy functional.

E_{Normal}^{Pt} specifies the normal direction of a surface at Parameter (u,v), and can be expressed as

$$E_{Normal}^{Pt} = K \left((S_u \bullet N)^2 + (S_v \bullet N)^2 \right) \quad (12)$$

Where \bullet is the dot product, N is the specified normal direction of a surface at Parameter (u,v). The normal constraint can be used to manipulate a surface normal. Fig. 6 illustrates the manipulation of a normal direction point constraint. The haptic cursor is shown as the green sphere and the normal direction constraint is shown as a cylinder. The small picture illustrates that, after the haptic operations, by simply adding fillet features, a telephone model is created. As discussed in the next subsection, for controlling the tangent plane of a B-Spline surface, using normal direction constraints is superior than using traditional C1 constraints.

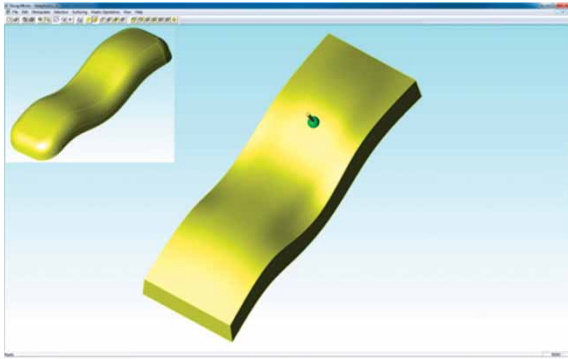


Fig. 6: Haptic-based normal direction operation.

(d) Energy-based normal direction curve constraints E_{Normal}^{Cur}

E_{Normal}^{Cur} specifies the normal direction of a surface along the target curve. As illustrated in Fig. 3 and Fig. 4 (b), E_{Normal}^{Cur} can be used to enforce a tangent continuity with neighboring surfaces (in Fig. 3) or to keep a fixed normal direction (in Fig. 4 (b), i.e., keeping the normal direction as (0, 0, 1) along its boundary curves while the surface is being “deformed” by a point-based haptic manipulation). E_{Normal}^{Cur} can be expressed as

$$E_{Normal}^{Cur} = K \int \left((C_n(t) \bullet N(t))^2 + (C_t(t) \bullet N(t))^2 \right) dt \quad (13)$$

Where \bullet is the dot product, $N(t)$ is the specified normal direction of a surface along the target curve at Parameter t . $C_t(t)$ is the surface C1 directional derivative in the direction of the curve, i.e., in the direction $(u_t(t), v_t(t))$, where $u_t(t)$ and $v_t(t)$ are the unit vector of the curve tangent in the u, v parameter domain at Parameter t . $C_n(t)$ is the surface C1 directional derivative defined in the direction normal to the curve, i.e., in the direction $(v_t(t), -u_t(t))$. By the chain rule, $C_n(t)$

and $C_t(t)$ are defined as

$$C_t(t) = u_t(t)S_u + v_t(t)S_v \quad (14)$$

$$C_n(t) = v_t(t)S_u - u_t(t)S_v \quad (15)$$

Generally speaking, since positional curve constraints $E_{Position}^{Cur}$ will automatically guarantee $C_t(t) \bullet N(t) = 0$, so E_{Normal}^{Cur} can be simplified as

$$E_{Normal}^{Cur} = K \int (C_n(t) \bullet (t))^2 dt \quad (16)$$

For the normal direction curve constraint, the u and v derivatives (both direction and magnitude) are free to change so as to generate the “smoothest” B-Spline so long as the generated surface satisfies the specified surface normal direction. Due to this, for a tangent continuity, using a surface normal direction constraint can generate a smoother surface than using a C1 constraint, which requires to artificially estimate and fix the magnitude and direction for the u and v derivatives. Also, for most cases, e.g., the generated surfaces in Fig. 3, it would be difficult or impossible to artificially estimate the magnitude and direction for the u and v derivatives, whereas it is quite straight forward to specify the normal directions based on the neighboring surfaces’ normal vectors. As a result, E_{Normal}^{Cur} can largely replace energy-based C1 curve constraints E_{C1}^{Cur} .

(e) Energy-based C1 point constraints E_{C1}^{Pt}

E_{C1}^{Pt} specifies the surface C1 directional derivative at Parameter (u,v), and can be expressed as

$$E_{C1}^{Pt} = K (S_{C1}(u, v) - P_{C1_Target})^2 \quad (17)$$

where K is the weight, P_{C1_Target} is the specified C1 directional derivative. $S_{C1}(u, v)$ is the surface C1 directional derivative at Parameter (u,v). By the chain rule, $S_{C1}(u, v)$ is defined as

$$S_{C1}(u, v) = aS_u + bS_v \quad (18)$$

Where (a, b) is the specified parametric direction, and is an unit vector. (Directional Derivatives: The common u/v derivatives of a surface, i.e., derivatives in the parametric directions (1, 0) or (0, 1), are special cases of “directional derivatives”. In other words, surface derivatives in any parametric directions can be specified, not just in the standard u/v directions.) Fig. 7 illustrates the use of normal direction curve constraints and a C1 point constraint. In Fig. 7 (a), only normal direction curve constraints are used. The middle surface is generated by interpolating a point constraint, the two red edges, and surface normal direction constraints along the shared red edges. From Fig. 7 (a), it shows that the derivatives of v of the generated B-Spline direct to directions different from the existing surfaces, but the generated B-Spline has the same surface normal directions with the existing surfaces - indicate the use of the normal direction

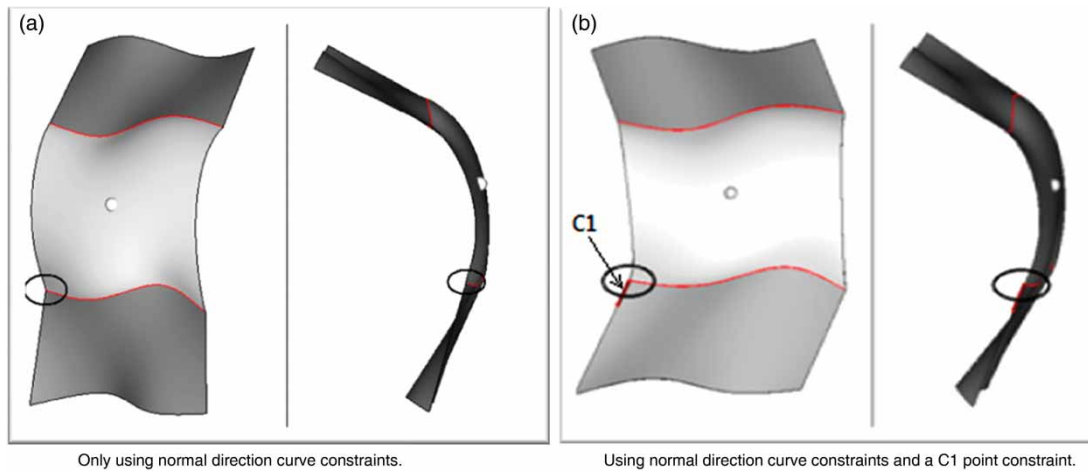


Fig. 7: The use of normal direction curve constraints and a C1 point constraint.

curve constraints. In Fig. 7 (b), both normal direction curve constraints and a C1 point constraint are used (The direction of the C1 point constraint is shown as a cylinder.). The generated B-Spline has the same v derivative with the existing surface at that point.

3.5. Performance and the Advantages of Using Haptics for Surface Design

By minimizing the energy functional in Eqn. 3, the solution, i.e., the optimized control points of the B-Spline surface, is generated. This is a standard mathematical issue of minimization. In this paper, all B-Spline energy items discussed above, i.e., the surface energy E_{Surf} and the constraint-based energy E_{Cons} , have quadratic forms, thus they are numerically efficient to solve. Basically, this can be done by setting the partial derivative of Energy E_v in Eqn. 3 with respect to each control point to zero. The mathematical details of deriving the linear system are similar to that discussed in [2] and [20], though [2] and [20] discuss the “Lagrange” method. The minimization process in this paper is simpler than the method in [2] because, unlike Eqn. (2) (refer to the following paragraph), no extra effort is needed for solving the constraints in Eqn. 3. (It should be noted that our implementation also adopts other methods, e.g., some traditional non-linear energy items could also

be included and be solved efficiently, but they are not related to the haptic operations discussed in this paper and are beyond the scope of this paper.)

Tab. 1 shows the performance benchmarks of VBS kernel, and Fig. 8 illustrates the testing models for producing the benchmarks in Tab. 1. As shown in Fig. 8 (a), the surface is required to interpolate eight red curves and a point constraint. The generated B-Spline surface is shown in Fig. 8 (b). This test is performed on a computer with an Intel i5-2320 3.0 GHz processor and 8 GB ram. It should be noted that the benchmarks in Tab. 1 are not just for these examples. Thousands of models have been tested, and the results are consistent for all cases. In other words, almost exactly the same benchmarks will be achieved if VBS kernel is tested with different models. Fig. 8 (c) shows the creation of a B-Spline surface from the boundary of a car body side (a fully fledged and industrial-size model), and that boundary includes about 50 curve constraints. The performance benchmarks are mainly determined by the number of control points, and other factors have little impact to the performance of the solver. For example, increasing the number of constraints could slow down the performance, but the slowdown is negligible. Also in practice, a B-Spline surface with a couple of hundreds of control points is generally enough to satisfy complex constraints. In other words, it can solve virtually all practical B-Spline surfaces in a matter of milliseconds. We agree with Piegls’ opinion [15] that designing a useful system is difficult and requires both theoretical research and practiced experience.

Compared with the “hard-constraint” method in Eqn. (2), the system in Eqn. (3) is more efficient and robust to solve because it eliminates Eqn. (2) and all the related processes of handling those ill-conditioned constraints: As discussed in Section 2.2, the constraint equations in Eqn. (2) are numerically very close to singular. So an initial reduction of the

Number of control points	Time for solving the B-Spline surface (milliseconds)
20×20	8
40×40	31
60×60	62
200×200	858

Tab. 1: Performance Benchmarks of VBS kernel.

constraints in Eqn. (2) to a set of independent rows must be performed and then these generated independent rows need to be merged with Eqn. (1), which is time-consuming and numerically unstable [2,12,20]. As Michalik et al. [12] state, for a surface with 500 control points, the SVD step alone already requires 15–20 seconds, and the run-time grows rapidly for larger examples. For a surface with 200×200 control points, it is practically unsolvable with the “hard-constraint” method. The result in Tab. 1 is literally hundreds (even thousands) of times more efficient and powerful than the performance in previous publications, e.g., [12] (Note that [12] used a slower machine, and the

discussion above already considers that factor. Here we need acknowledge others’ contributions to this work: the referred papers inspire this research, and without their work, we could not achieve the results discussed in this paper.). Based on the precision issue discussed in Section 3.2 and the performance issue discussed here, it is clear that the “hard-constraint” or “Lagrange” method should not be used for variational B-Spline technique in practice.

The technique discussed in this paper enables us to explore the use of the variational B-Spline technique for haptic-based surfacing design. It should be noted that the surface only needs to be solved

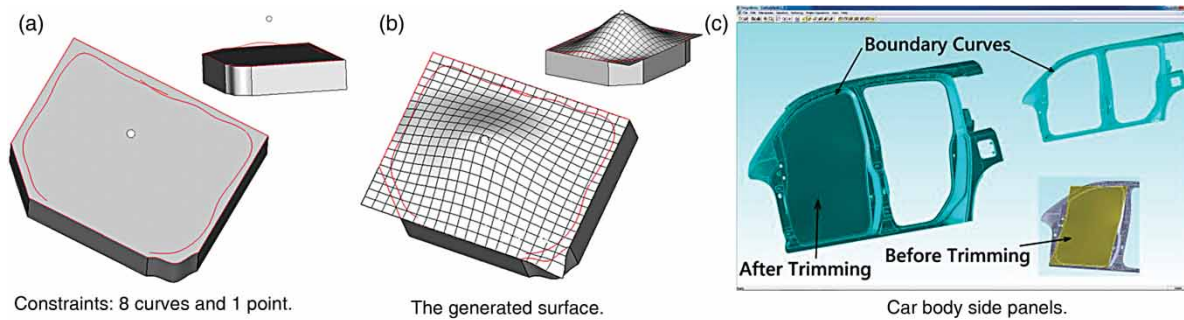


Fig. 8: The models for benchmark testing.

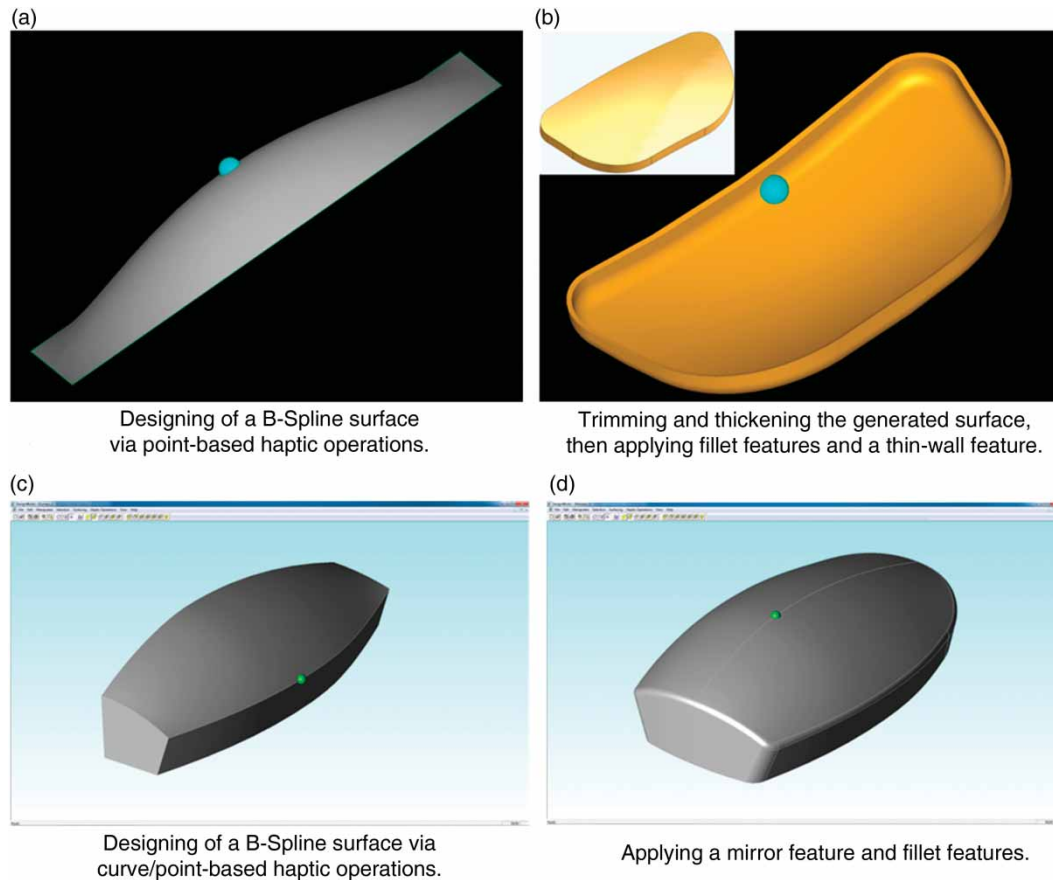


Fig. 9: The creation of a CAD model using haptics and common CAD operations.

once in OnHapticButtonDown, not for each updating operation in OnHapticMove. During the OnHapticMove event, the system simply uses the linear system, which has already been solved in OnHapticButtonDown, to update the B-Spline surface in response to haptic movements. So the calculation time needed in OnHapticMove is trivial.

Fig. 9 shows the creation of a CAD model by combining haptic-based surface operations and common CAD operations. The haptic cursor is shown as a sphere. The B-Spline surface in Fig. 9 (a) is created by the point-based haptic manipulation. In Fig. 9 (b), the generated surface is trimmed and thickened (the small picture at the up left of Fig. 9 (b)), and then fillet features and a thin-wall feature are added. The surfaces in Fig. 9 (c) is created by the curve/point-based haptic manipulation, and the operational procedure is similar to Fig. 4 (c) and (d). By adding some common CAD operations, i.e., a mirror feature and fillet features, a mouse model is created in Fig. 9 (d). The haptic-based surface operations allow designers to directly touch an arbitrary position on a surface or a curve, and then manipulate/design the B-Spline surface in a natural 3D environment. They provide a greater flexibility for the design of B-Spline surfaces. Compared with other non-force-feedback 3D devices, e.g., a Flock of Birds (FOB) tracking sensor, haptic interfaces show significant advantages:

- Tactile or force feedback: without a tactile sense or force feedback, many simple operations, e.g., locating a 3D tracking sensor or cursor on a surface or a curve and starting an operation, are difficult to achieve. With a tactile sense or force feedback, such operations are quite straight forward - For a surface, a haptic interface provides tactile feedbacks so that users can directly touch the surface and locate the cursor on a surface. For a curve, an attraction force can be used to guide the device and locate the cursor on the curve.
- Stability: Other 3D devices, e.g., FOB, may carry some magnetic noise and instability, and the instability and magnetic noise presented in these 3D interfaces make it difficult to gain fine controls. In fact, it makes accurate operations virtually impossible. Whereas the haptic device is very accurate, and there is no perceptible instability existed.

3.6. Future Work

While the variational B-Spline technique provides the performance and capacity that are powerful enough for using haptics in designing virtually any complex B-Spline surfaces, there is little known about how to design a suitable user interface and tools for using haptics to design B-Spline in CAD, i.e., what type of haptic tools are best suited for designers. It

takes the CAD industry and academic researchers 20 years before the parametric and feature-based design becomes available for CAD users. Developing efficient UIs and tools for using haptics in designing CAD surfaces needs further research. We plan to further explore this issue in the future research.

4. CONCLUSIONS

This paper presents the technique for using a haptic device to design B-Spline surfaces based on the variational B-Spline technique. With the variational B-Spline technique discussed in this paper, we achieve the performance and capacity that are much more efficient and powerful than previous work in the variational B-Spline field. The discussed technique delivers all the performance and capacity required for using haptics to design high-quality B-Spline surfaces, and haptic-based operations provide a greater flexibility for the design of B-Spline surfaces.

REFERENCES

- [1] Bourdot, P.; Convard, T.; Picon, F.; Ammi, M.; Touraine, D.; Vezien, J.: VR-CAD integration: Multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models, *Computer-Aided Design*, 42(5), 2010, 445-461. <http://dx.doi.org/10.1016/j.cad.2008.10.014>
- [2] Celniker, G.; Welch, W.: Linear constraints for deformable non-uniform B-spline surfaces, *Proceedings of Symposium on Interactive 3D Graphics*, 1992.
- [3] Evans, M.; Wallace, D.; Cheshire, D.; Sener, B.: An evaluation of haptic feedback modelling during industrial design practice, *Design Studies*, 26(5), 2005, 487-508. <http://dx.doi.org/10.1016/j.destud.2004.10.002>
- [4] Faeth, A.; Oren, M.; Sheller, J.; Godinez, S.; Harding, C.: Cutting, Deforming and Painting of 3D meshes in a Two Handed Viso-haptic VR System, *Virtual Reality Conference*, 2008, 213-216.
- [5] Gao Z., Gibson I.: Haptic sculpting of multi-resolution B-spline surfaces with shaped tools, *Computer Aided Design*, 38 (6), 2006, pp. 661-676. <http://dx.doi.org/10.1016/j.cad.2006.02.004>
- [6] Grossman, T.; Balakrishnan, R.; Singh, K.: An interface for creating and manipulating curves using a high degree-of-freedom input device, *ACM CHI Conference on Human Factors in Computing Systems*, *ACM CHI Letters*, 6(1), 2003, 185-192.
- [7] Ingrassia, T.; Cappello, F.: VirDe: a new virtual reality design approach, *International Journal on Interactive Design and Manufacturing*,

- 3(1), 2009, 1-11. <http://dx.doi.org/10.1007/s12008-008-0056-2>
- [8] Liu, X.; Dodds, G.; McCartney, J.; Hinds, B.: Virtual DesignWorks—designing 3D CAD models via haptic interaction, *Computer-Aided Design*, 36(7), 2004, 1129-1140. <http://dx.doi.org/10.1016/j.cad.2003.10.003>
- [9] Liu, X.; Dodds, G.; McCartney, J.; Hinds, B.: Manipulation of CAD surface models with haptics based on shape control functions, *Computer-Aided Design*, 37(14), 2005, 1447-1458. <http://dx.doi.org/10.1016/j.cad.2005.02.015>
- [10] Liu, X.: Filling N-Sided Holes Using Trimmed B-Spline Surfaces, *ASME IDETC/CIE 2012*, Chicago, IL, USA, August 12-15, 2012.
- [11] McNeely, W.; Puterbaugh, K.; Troy, J.: Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling, *SIGGRAPH - Computer Graphics Conference proceeding*, 1999, 401-408.
- [12] Michalik, P.; Kim, D.; Bruderlin, B.: Sketch- and constraint-based design of B-spline surfaces, *Proceedings of the seventh ACM symposium on Solid modeling and applications*, June 17-21, 2002.
- [13] Moreton, H.; Séquin, C.: Functional minimization for fair surface design, *Computer Graphics*, 26(2), July 1992. <http://dx.doi.org/10.1145/142920.134035>
- [14] Payandeh, S.; Dill, J.; Zhang, J.: Using haptic feedback as an aid in the design of passive mechanisms, *Computer Aided Design*, 39(6), 2007, 528-538. <http://dx.doi.org/10.1016/j.cad.2007.01.011>
- [15] Piegl, L.; Tiller, W.: *The NURBS Book*, 2nd Edition, Springer, Berlin, 1997. <http://dx.doi.org/10.1007/978-3-642-59223-2>
- [16] Qin, H.; Terzopoulos, D.: D-NURBS: A physics-Based framework for geometric design, *IEEE Transactions on Visualization and Computer Graphics*, 2(1), 1996, 85-96. <http://dx.doi.org/10.1109/2945.489389>
- [17] SensAble Technologies web site, <http://www.sensable.com/>, being valid as of Nov 2013.
- [18] Volkov, S.; Vance, J. M.: Effectiveness of Haptic Sensation for the Evaluation of Virtual Prototypes. *Journal of Computing and Information Science in Engineering*, 1, 2001, 123 -128. <http://dx.doi.org/10.1115/1.1384566>
- [19] VBS kernel Website, <http://geom.hebut.edu.cn/>, being valid as of Nov 2013.
- [20] Welch, W.; Witkin, A.: Variational surface modeling, *Computer Graphics (SIGGRAPH'92 Proceedings)*, 26, 1992, 157-166.