B-spline Surface Based Concept Design Module for Applications in Virtual Reality Environment

Harish Pungotra[1], George K. Knopf[2] and Roberto Canas[3]

[1]Beant College of Engineering and Technology, Gurdaspur, India hpungotr@alumni.uwo.ca
[2]The University of Western Ontario, London, Ontario, Canada gknopf@eng.uwo.ca
[3]National Research Council, London, Ontario, Canada roberto.canas@nrc-cnrc.gc.ca

ABSTRACT

Over the past decade, B-spline modeling has become the standard mathematical model for representing freeform or organic objects in CAD/CAM systems. Using a B-spline surface to represent the virtual model in a haptic concept design module helps to streamline the exchange of the information with existing CAD/CAM systems. B-spline surfaces also help increase visual realism because these represent continuous surface. At the same time, physically based virtual deformable models provide a sense of realism to the user. However, the algorithms used for interaction with the virtual model should be efficient to maintain acceptable level of virtual realism and perceive the simulation as continuous with no time lag. This paper presents a B-spline surface based concept design framework, which can be used to generate various concepts, evaluate these concepts, and provide user training. The proposed framework assumes that all deformable models are represented as B-spline surfaces. The tools used to interact with the deformable models can be implicit surface, tessellated surface, B-spline surface, or point-based. The technique exploits blending matrices for the B-spline surface that are independent of the control point positions and, hence, can be pre-calculated prior to haptic interaction. Once determined, the pre-calculated blending matrices are used to generate discrete points on the B-spline surface. Mass spring system is used to incorporate material properties to the virtual objects. Practical illustrations of the concept design framework are presented for modeling and evaluation of concepts and provide training to the intended user segment.

Keywords: concept design, deformable model, virtual reality, blending matrices.
DOI: 10.3722/cadaps.2013.247-263

1 INTRODUCTION

A VR-based concept design system can provide industrial designers with more familiar interactive capabilities for creating and representing their design intent easily, flexibly and efficiently on computers [33]. Compared to conventional CAD systems, a VR-based CAD system allows more tools for the designers to perform various design activities. Several studies have indicated that virtual reality is a very intuitive, creative and cost effective method of performing product concept design[8, 27, 32]. Clearly the seamless integration of 3D graphics and haptic devices with existing CAD/CAM software will enable virtual reality to better assist industrial designers and engineers with their creative endeavors. Product designers and engineers require concept design and graphical visualization tools that enable them to quickly modify the shape, style, and functionality of a product concept. The primary role of VR technology in creative product design is to provide the designer with the ability to intuitively create and manipulate the shape of complex freeform CAD models during concept generation.

In addition to realistic three dimensional graphics, a virtual reality environment for product concept design must support visual object collision detection, physics-based modeling, and haptic manipulation. At the same time, the user may like to merge two or more models to create complex and interesting design concepts. The *collision detection* sub-system provides detailed information about when and how multiple virtual objects make contact and interact within the VR space. The *physics-based* sub-system uses the information provided by the collision detection algorithm to determine the reactive forces to be fed back to the user and degree of deformation of any non-rigid elastic and plastic objects. *Force Feedback* is obtained through haptic devices. The nature and amount of force feedback depends upon the haptic device used, and the underlying algorithm used to calculate it. Finally, the *haptic rendering* sub-system is used for tactile and visual interaction with virtual objects and tools used during the creative design exercise. Fig. 1 shows the schematic representation of a general framework for a haptics-based concept design module.

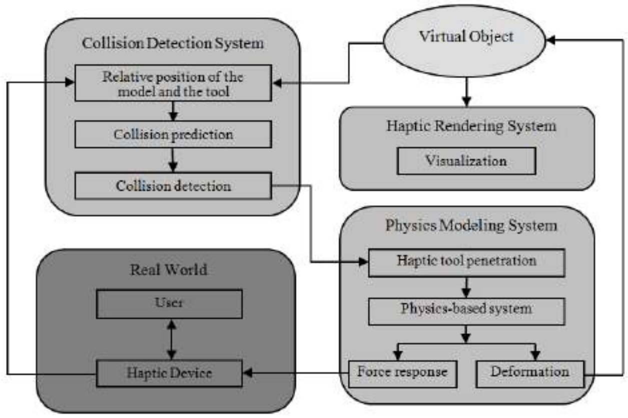

Fig. 1: Schematic representation of a haptic interaction with a virtual object.

Industrial designers tend to make extensive use of physical models created with their hands as it is natural and intuitive process. Unfortunately, current CAD and geometric modeling systems lack a natural and intuitive human-computer interface. In this regard, a haptics-based concept design system would be best suited to exploit the creativity of an industrial designer or engineer. It would make it possible to artistically modify and quickly evaluate different concepts. The user gets real-time visual and force feedback, while interacting with the virtual model. Haptics allows the user to benefit from the natural way of working with the model, without the constraints imposed by commercially available CAD/CAM software. The 3D visualization capabilities and the ability to directly interact with physics-based models using haptic tools provides the designer, an insight to the physical characteristics of the concept, which can be evaluated before going in for a detailed design.

Computationally efficient bi-parametric functions, such as B-spline and NURBS (non-uniform rational B-spline), are commonly used to model organic and freeform shapes and, therefore, provide a viable mathematical representation for describing the geometry of realistic objects in VR space. In the past decade, B-spline representation has become the standard for CAD/CAM systems. Thus it is imperative that any haptic concept design module should utilize B-spline surface to represent the virtual model in order to streamline the exchange the information with existing CAD/CAM systems.

This paper presents describes a computational framework for concept design and evaluation in a virtual reality environment. The proposed method extends the collision detection [20] and B-spline merging algorithms [22] previously developed by the authors and utilizes the blending matrices for efficiently integrating a mass spring system to incorporate material properties in the virtual model. No assumption regarding the complexity, degree of the B-spline surface, or the number of control points representing the surface patches are imposed on the solution. To illustrate the capabilities of the algorithm and verify its performance, practical illustrations are presented for concept design, evaluation of the concept, and user training.

2 PREVIOUS WORK

The underlying algorithm for a concept design framework depends upon the types of surfaces used to represent model and tool. Many data structures have been proposed to represent models for virtual sculpting and design. These include voxel-based system [17], a voxel-based system with iso-surface extraction [13], B-spline surfaces [11], sub-division surfaces [14], and field-based implicit surfaces [3, 6].

Both geometric and physics based techniques have been used to incorporate material properties in a virtual model. Unlike geometric techniques, computationally intensive physics-based techniques can yield real material behavior of a multiple-material/non-homogeneous virtual model [16]. Physics-based deformation models give the designer, more opportunities to try different types of materials during the concept design phase and validate product models in real-time [23]. The pioneering work of Terzopoulos *et al.* [28], Waters [30], and Platt and Barr [19] has shown the advantages of physically based models over geometric-based computer animation techniques. Igwe *et al.* [15] proposed to generate hexahedral mesh of mass spring system by using *volumetric self-organizing feature map* (VSOFM). FEM has been used for fairly simple physical systems to simulate tissue deformation [10, 26].

Various collision detection, merging and geometric/physics based techniques have been integrated to create a framework for concept design module. Initially, VR-enhanced 3D visualization and analysis systems, such as Virtual Design II [2] and ISAAC [18] were used. The COnceptual VIRtual Design System (COVIRDS) allowed rapid shape creation by using a bi-modal, voice and hand tracking interface [9, 12]. It provided parametric and free form design modes. Many VR-based design system have been reported in the literature such as 3-Draw [25], 3DM [5], DesignSpace [7], CDS [4], and CUP [1]. The Loughborough University Conceptual Interactive Design (LUCID) system [31, 32] was developed to integrate VR-based Human-Computer Interfaces (HCIs) into the design process in order to maximize its interactivity and efficiency so as to provide better support to conceptual design.

B-spline surface based concept design framework will help streamline the exchange of information with existing CAD/CAM systems. The underlying geometry of mass spring models can easily be modified to represent topology changes. This can be used to sculpt and evaluate B-spline surface based design concept.

3 BACKGROUND ON B-SPLINE SURFACES

A B-spline surface with r and s number of control points in u and v directions is mathematically represented by the equation,

$$S(u,v)=\sum_{i=0}^{r-1}\sum_{j=0}^{s-1}B_{ik}(u)\,B_{jl}(v)\mathbf{P}_{ij} \tag{1}$$

where $\mathbf{P}_{ij}$ is the control points vector, $B_{ik}(u)$ and $B_{jl}(v)$ are the blending functions of the surface with degree k and l in u and v directions, respectively. In general, the parameters are bound over 0 **Ò** *u, v* **Ò** 1. There are certain parameters which determine the characteristics of the blending functions. These include periodicity of the surface (in u or v or both directions), knot vector, the degree of the surface in u and v directions and the number of control points in u and v direction. However, the blending functions are independent of the position of the control points. A B-spline blending function has the property of recursion which is defined as:

$$B_{i,k}(u)=(u-u_i)\frac{B_{i,k-1}(u)}{u_{i+k-1}-u_i}+(u_{i+k}-u)\frac{B_{i+1,k-1}(u)}{u_{i+k}-u_{i+1}} \tag{2}$$

where $B_{i,1}=\begin{cases}1, & u_i \le u \le u_{i+1}\\ 0, & otherwise\end{cases}$

If the B-spline surface is periodic in u direction, the blending function is modified as:

$$B_{i,k}(u)=B_{0,k}((u-i+r+1)\ \mathrm{mod}\,(r+1)) \tag{3}$$

A B-spline and NURBS based surface can be expressed as a tensor product. A B-spline surface patch can be discretized into a set of a parametrically uniform grid of discrete points in u and v directions. The matrix of these discrete points $\mathbf{M}$, can be determined from the equation,

$$\mathbf{M} \;=\; \mathbf{A}_u\mathbf{P}_{ij}\mathbf{A}_v \tag{4}$$

where $\mathbf{A}_u$ and $\mathbf{A}_v$ are blending matrices (also known as B-spline Blending Transformation Matrices or simply Transformation Matrices). The values of blending matrices depend upon the blending functions and u and v parametric values. Equ (4) can be re-written as:

$$\mathbf{P}_{ij}=[\mathbf{A}_u^{\mathrm{T}}\ \mathbf{A}_u]^{-1}\mathbf{A}_u^{\mathrm{T}}\ \mathbf{M}\ \mathbf{A}^{T}_{v}\ [\mathbf{A}_v\mathbf{A}_v^{\mathrm{T}}\ ]^{-1}, \tag{5}$$

to determine the position of the control points that can represent a B-spline surface passing through parametrically uniform discrete points of matrix $\mathbf{M}$.

Eqn. (1, 4-5) will determine the correlation between control points and the discrete points generated on the B-spline surface.

4 MODELING TECHNIQUES FOR CONCEPT DESIGN FRAMEWORK

The concept design framework proposed in this paper uses three basic modeling techniques. The collision detection algorithm for deformable surfaces provides information regarding the intersection of the model and the tool. This information is used by the mass spring model to calculate the resultant deformation of the model and the force to be fed back to the user. At the same time, the merging of B-spline surface patches algorithm allows the user to integrate different B-spline surface patches to generate complex and interesting concepts. In this section, these techniques are discussed in detail.

4.1 Collision Detection

The collision detection algorithm used in this framework assumes that all the deformable models and tools are represented as B-spline surfaces. The rigid tools can be implicit surfaces. No assumption regarding the motion of the objects, size of objects, degree of the surface, area of contact and extent of deformation is needed. A brief description of this algorithm is presented in this paper. The detailed discussion was presented in [20]. The algorithm has two stages: the pre-processing stage and the run-time stage.

4.1.1 Preprocessing Stage

During the pre-processing stage, depending upon the parameters of the B-spline surface (periodicity, number of control points in the u and v directions, the maximum number of points to be generated), blending matrices (also known as transformation matrices) are generated and stored along with their inverses. Two blending matrices are needed for each B-spline surface for generating discrete points on the B-spline surface. The collision detection algorithm not only calculates surface normal at the points generated but also calculates tangents in the u and v direction. The surface normals help in correctly mapping the forces to the surface by calculating the forces normal to the surface. The blending matrices for finding tangents are also generated and stored during the pre-processing phase. A convex hull also generated during the pre-processing. As control points used for B-spline surfaces are usually small in number, the computational cost is small.

4.1.2 Run-time Phase

The run-time phase starts by checking for the intersection of convex hulls of B-spline surfaces initially calculated during pre-processing phase. Once the surfaces of the convex hulls intersect, the algorithm determines the corresponding minimum and maximum values of u and v parameters associated with the control points of these surfaces. It then generates sparse points within these limits on the surfaces of the model and tool. Spheres are generated using these sparse points in such a way that they cover the whole surface area. These spheres are checked for intersection with those on the tool or another B-spline surface model. If some of the spheres are intersecting, the algorithm generates more points (by using more values of u and v from the blending matrix) within the u and v parameter limits of the intersecting spheres and discards the points generated in non-intersecting spheres. This process of generating lower levels of details continues until all the rows of the blending matrix are used. The points generated at the end of this loop are then used for triangulation of the surface. If all the spheres generated on the model and the tool are intersecting, then the algorithm does not proceed to lower levels of details and rather stops at the last level of detail. It then compares the normals of new points being generated within the u and v parameters. The triangle-triangle intersection test is then carried out to find out the parts of the surfaces which are intersecting.

4.2 Merging B-spline Surfaces

The user may need to merge two or more B-spline surface patches to generate a complex and interesting concept. The concept design framework allows the user to merge two or more B-spine surfaces by using B-spline surface merging algorithm [22]. This algorithm uses blending matrices associated with each B-spline surface being merged [24].

The algorithm to merge N number of B-spline surface patches, S^1, S^2, **é** , S^N; having (r_1, s_1), (r_2, s_2), **é** , (r_N, s_N) number of control points in u and v directions respectively, starts by discretizing the surfaces to be merged. These discretized matrices **M**1, **M**2, é , **M**N are combined to generate a matrix **M** of discrete points. The algorithm calculates the revised number of control points òró and òs” for the merged surface in u and v directions respectively. If the two surfaces are being joined only in the u direction, the new number of control points in the u direction is then given by $r = r_1 + r_2 - 1$, where r is the number of control points of the merged surface in u direction, and r_1 & r_2 are the number of control points in u direction for the first and the second surface respectively. The number of control points in v direction would remain the same, provided the number of control points is the same for both the surfaces in that direction, that is, if $s_1 = s_2$. If surfaces have a different number of control points in v direction then the larger number is assumed as the new number of control points.

A knot vector determines the area of influence of each control point on the B-spline surface. The knot vector U and V are computed by combining the knot vectors (U^1, V^1), (U^2, V^2), é ., (U^N, V^N) of the surfaces being merged. If the degree of the initial surfaces is not being changed, the algorithm simply uses the knot vector of the first surface and then adds to it the knot vector of the second surface. The multiple knots at the common edge are reduced depending upon the type of connectivity needed at the edge. For providing only C^0 connectivity, a total of k multiple knots are retained, where k is the degree of the merged surface in the direction of merging. For C^1 connectivity, the number of multiple knots would be k-1 and so on. For the maximum connectivity (C^2 for a degree 3 surface) only one knot is retained at the common edge.

In this manner, the degree of the final merged surface, its knot vector, number of control points in u and v direction, and the total number of discrete points to be generated in u and v direction are calculated. Once these parameters are known, the new set of blending matrices, $\mathbf{A}_u$ and $\mathbf{A}_v$, can be generated. In a similar fashion, the inverse of these knot vectors ($[\mathbf{A}_u{}^{\mathrm{T}}\mathbf{A}_u]^{-1}, [\mathbf{A}_v\mathbf{A}_v{}^{\mathrm{T}}]^{-1}$) is also computed and stored. These newly calculated blending matrices, replace the previous blending matrices for the two B-spline surfaces. These new blending matrices are used to determine the position of the control points which can generate the matrix of discrete points (**M**). The position of the control points for the merged surface is once more calculated using Eqn. (5). The algorithm is capable of merging intersecting and trimmed surfaces.

4.3 Mass Spring system

A physics-based technique can realistically calculate the deformation and force response of the model based on virtual material properties [16]. A physics-based deformation model gives the designer options to try different types of materials during the concept design phase and validate it in real-time. Some physically-based models are computationally intensive and therefore, are unsuitable for real-time interaction. However, a mass spring damper system, consisting of a set of particles (nodes) connected through a network of spring and dampers can provide reasonable accuracy and speed for real time interaction [21].

The virtual object is modeled as a collection of point masses connected by springs and dampers in a lattice structure. In general, the spring forces are assumed to be linear. However, nonlinear springs can also be used to model objects which exhibit inelastic behavior. Such a system contains a mass, a spring with spring constant K that serves to restore the mass to a neutral position, and a damping element which opposes the motion of the vibratory response with a force proportional to the velocity of the system. Different combinations of linear springs and damper can be used to model deformable objects. Voigt model is the most commonly used combination of spring and damper and has been used in the proposed framework. The stiffness of the material primarily affects the linear and non-linear elasticity range in the deformation zone. For solids, the stiffness in proportional to the elastic modulus and also depends upon an element's dimensions. The linear model stiffness can be express as

$$K = \frac{A \times E}{L} \tag{6}$$

where A is the cross-sectional area, E is the Young's modulus of elasticity, and L is the length of the element. In one dimension, Young's modulus of elasticity can be considered as a measure of stiffness of material. Thus, stiffness and damping constants can be used to model a realistic behavior. The stiffness constant controls the elastic behavior and the combination of stiffness and damping constants control plastic behavior of the material. Three significant types of damping that are generally encountered in dynamic behavior of the model are; coulomb, hysteresis, and viscous damping. In this paper, these three types of damping are approximated as a velocity dependant viscous damping. The velocity dependent damping force exerted on the node i from the interaction with the node j is given by,

$$f_d = -D(\dot{x}_i - \dot{x}_j) \tag{7}$$

where f_d is the damping force, D is the damping coefficient, $\dot{x}_i$ and $\dot{x}_j$ are the velocities of node i and node j, respectively. The fundamental Lagrange equation of motion can be expressed as,

$$\rho \ddot{x} = -D\dot{x} - Kx + f_x + f_g \tag{8}$$

where ρ (in kg) is the mass of the node, D (in N.s/m) is damping coefficient, K (in N/m) is the stiffness coefficient of each spring, f_x and f_g are the external and gravitational forces acing of the node. In the absence of external and gravitational forces, Eqn. (8) can be written as,

$$\rho \ddot{x} + D\dot{x} - Kx = 0 \tag{9}$$

5 CONCEPTUAL PRODUCT DESIGN

The concept design process needs creativity and freedom to innovate and explore. During the concept generation phase a rough idea, which can come from the background research or from a previous design, is expanded into several solution alternatives. Valuable insight into refining the concepts can be gained using evaluation methods involving end-users. An important goal is to identify whether the new product concepts find acceptance amongst the intended target user group. Another goal may be to evaluate the design from a human factors perspective. User evaluation is useful as a tool for iteratively refining the designs based on user feedback in accordance with the concept of user-centered design. The concept design framework proposed in this paper can enable an industrial designer to sculpt and validate the design in the virtual reality environment. These modifications will often deal with a variety of needs that can be fulfilled using innovative but easily implemented changes to the existing generic model.

Fig. 2 shows various modeling techniques presented in this paper and their relationship to the conceptual design module. These techniques are integrated to develop a B-spline surface based concept design module for various applications in virtual reality environment. The same virtual reality issues exist for user training and the modeling techniques presented in paper can be used to provide a virtual environment to the user similar to the real world applications.

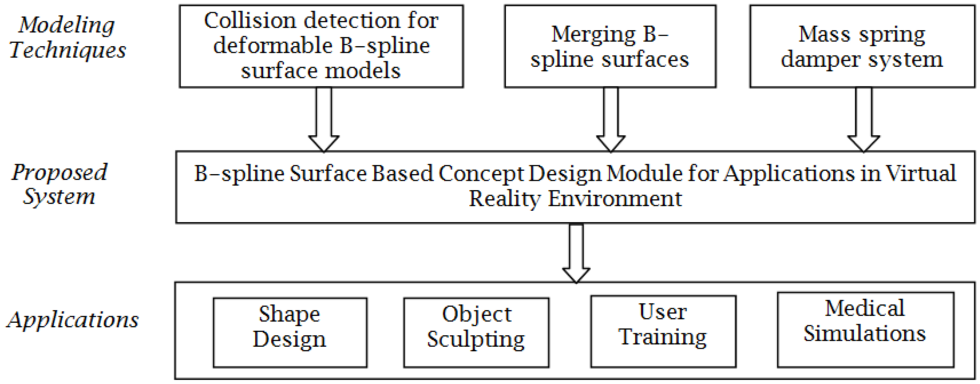

Fig. 2: Integration of various modeling techniques presented in this paper to develop B-spline surface based concept design module for various applications in virtual reality environment.

5.1 Conceptual Design of an Ergonomic Spoon

Real-time interaction with a product model during concept development provides a quick insight into the overall performance of the proposed solution. The example of designing a functional, stylistic spoon for different user segments is used to establish the application and desirability of the concept design framework proposed in this paper. Fig. 3 shows many of the commercially available spoons.

Fig. 3: Photograph of typical commercially available spoons. Product designs mainly focus on contemporary style, ease of use, and comfort of the user.

These designs presume that the intended users can efficiently work with their hands and fingers. However, for different sets of users, these designs may not be suitable from an ergonomic point of view in terms of *ease-of-use* and providing comfortable grip. One of the user segments is of the patients suffering from rheumatoid arthritis. Rheumatoid arthritis is a chronic inflammatory disorder that most typically affects the small joints in hands and feet [29]. Eating is one of basic tasks that can be impaired by rheumatoid arthritis. The spoons that are commercially available may not be helpful for this user segment. Virtual reality can make it possible for a designer to imitate a rheumatoid arthritis patient and develop ergonomic spoons for the users.

Due to weak grip (as in case of children and aged people) or restricted motion (as in case of patients suffering from rheumatoid arthritis), it is possible that a user cannot put the food properly in the spoon or can get the food on spoon, but cannot turn his/her wrist enough to bring it to mouth without spilling it. Even for other users, it is possible to come up with different concepts to develop ergonomic spoons. In a similar fashion, different materials can be incorporated in product design to reduce overall weight and enhance features such as high friction gripping surface to prevent slippage of spoon during use. In this context, a variety of concepts that cannot be addressed effectively using conventional CAD design packages may be examined using concept design simulation tools in virtual reality environment.

An initial design can be obtained by digitizing an existing design using reverse engineering approach. Point cloud data or tessellated surface can be used to obtain initial design represented as a B-spline surface. Conventional CAD system can also be used to create initial form. Alternatively, the initial design can be created from a virtual *lump of clay*. Fig. 4 shows the generation of initial design by different methods.

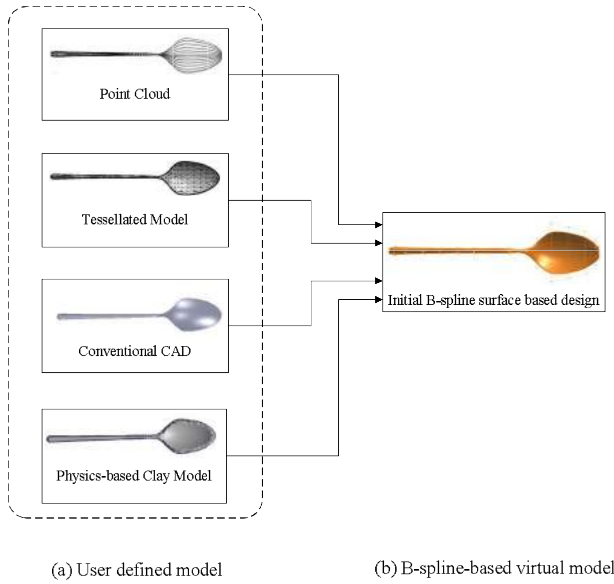

Fig. 4: Different methods to generate initial design represented as a B-spline surface.

Initial analysis can be carried out in VR space to determine if the product meets the requirements for all/intended user segments or if it needs modifications for particular segments of users. The industrial designer can investigate different shapes, sizes, and material of the spoon to determine a valid concept before final deciding on the best design for intended user segments. A spoon has two distinct parts, a handle for holding and a small shallow bowl (or shell), oval or round, at the end of the handle for fetching food. Fig. 5 shows two parts of a spoon.

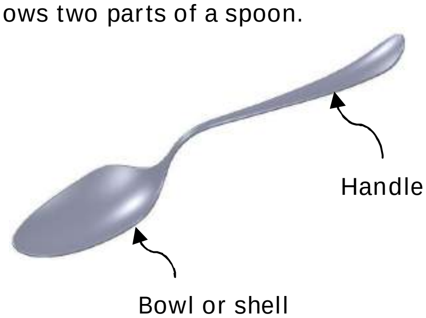

Fig. 5: Parts of a spoon.

A spoon can be modified as a whole or the two parts (handle and bowl) can be designed separately and later merged to get the desired concept. Separate parts are easier to manipulate and can give rise to more concept models by different combinations. The handle of a spoon can be modified in different shapes and sizes to accommodate impaired wrist movements or weak grip of fingers. To reduce the computational cost, the bowl (shell) part and handle are used as separate B-spline models. One of the concepts is to bend the handle so that the bowl part of the spoon faces the person and he/she does not need to bend the wrist to eat from the spoon.

Using the proposed concept design frame work, bending can be achieved by fixing one end of the spoon handle and applying force on the other part. Fig. 6(a) shows the handle of initial spoon design. In the design framework, a plane is used to fix a portion of handle and the force is applied by a sphere based tool. Fig. 6(b) shows the vertical plane and the sphere used to push the handle.

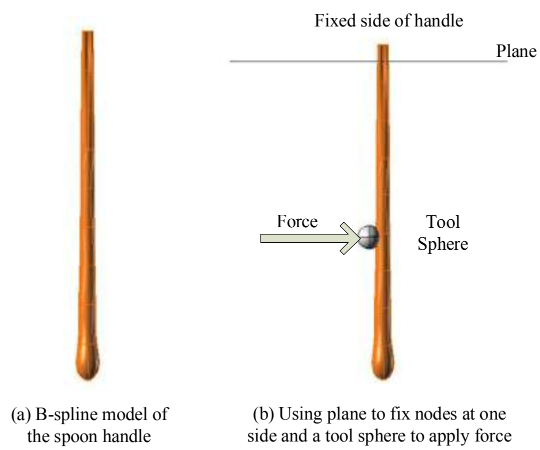

Fig. 6: (a) Handle of spoon from the initial design (b) A plane is used to fix nodes on one side of the plane (opposite to the direction of normal vector) and a sphere is used to apply force.

During the pre-processing phase, a mass spring mesh is created for the handle shown in Fig. 6(a). A 12×12×12 node mesh was created, from the points generated by using blending matrices. Fig. 7(a) shows the mass spring mesh of the spoon handle.

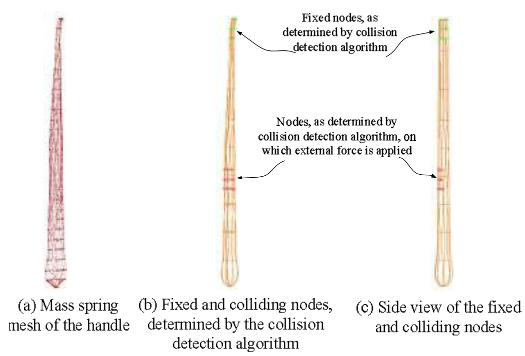

Fig. 7: (a) Mass spring mesh for the spoon handle (b) Fixed nodes (green) and the nodes colliding with the sphere tool (red) (c) Side view the B-spline model showing fixed and colliding nodes.

Collision detection algorithm checks the intersection of the plane with the spoon handle. All the nodes on one side of the plane (opposite side of the normal vector of plane) are fixed. The fixed nodes do not experience any movement or deformation. Even if these nodes experience internal or external force through spring damper system, these nodes do not move. The nodes (green color) shown in Fig. 7(b-c), are the fixed nodes detected by the collision detection algorithm. At the same time, the collision detection algorithm checks the intersection of a sphere and the B-spline model. It determines the region of the B-spline model colliding with the sphere and calculates the nodes of the mass spring system which will be experiencing external force. The information about the magnitude of the external force applied through the sphere and the nodes experiencing this force is transmitted the mass spring deformation system. Fig. 7(b-c) shows the nodes (red color), determined by the collision detection algorithm, which will experience external force.

As soon as a force is applied through the sphere, the B-spline model starts deforming. When a force is applied through the sphere, the portion of the handle near to the fixed nodes is bent and rest of the handle remains approximately straight. This design can accommodate the lack of wrist movement and the user does not have to rotate his/her wrist. However, due to the bend, the user will experience a torque. It is possible that the user cannot use fingers to hold the spoon due to lack of movement of fingers, or weak grip. If the user cannot hold the spoon with fingers, then the handle can be modified to for a firm grip using all the fingers. This will also help to effectively counter the torque resulting from the modified design.

(a) Handle of the spoon interacting with too sphere (b) Deformation due to applied force (c) Modified shape of handle after several interactions

Fig. 8: Modifying the shape of spoon handle to grip it with all the fingers of hand.

Fig. 8(a) shows the spoon handle and a tool sphere. Again, the collision detection algorithm determines the colliding nodes of the mass spring mesh, which would experience the external force applied through this sphere. As soon as the tool sphere is pulled out, the nodes of the mass spring mesh colliding with the tool sphere (determined by the collision detection algorithm) experience force in the outward direction. The external force acting on the colliding nodes starts pulling these mass spring nodes in the direction of the external force. In this case, local modification is more appropriate because we only intend to change the shape in a small region. Local modification uses small number of iterations and hence, the nodes which are away from the colliding surface do not get enough time to move under the application of the external force. Due to this reason, there is no need to have a plane to fix some of the nodes. Fig. 8(b) shows the deformation of the handle under the influence of external force applied through the sphere tool. By repeating this process at different points, the handle can be modified to accommodate all the four fingers to grab the handle while eating. Fig. 8(c) shows the spoon the with modified handle shape. Once various concepts of the spoon handle are generated, these handles need to be merged with the bowl section of the spoon. The B-spline surface patches merging algorithm can be used to merge the bowl and handle part to generate various concepts.

Fig. 9(a) shows the bowl and the modified spoon model. These parts must be merged to generate an integrated B-spline model of the design concept for a spoon. Due to the modification of the shape of the handle, the knot vector and the common edge of the bowl and handle did not match. The B-spline merging algorithm can merge these surfaces without matching or manipulating the control points of the B-spline surfaces.

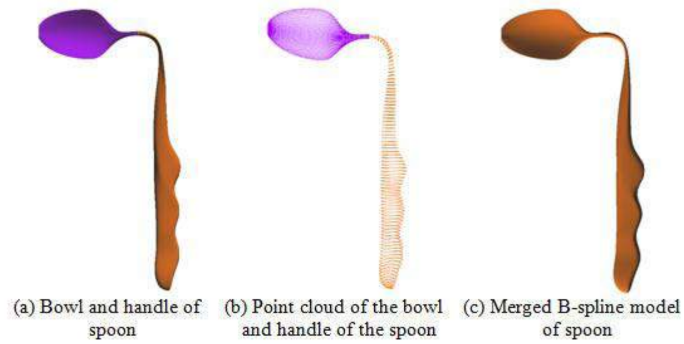

Fig. 9: Merging bowl and modified spoon handle to generate the B-spline model of a spoon.

Fig. 9(b) shows the point cloud of the bowl and handle, generated by using blending matrices stored during the pre-processing phase. These point clouds are merged to generate single matrix of the point clouds, $\mathbf{M}$. Using the matrix of points $\mathbf{M}$, revised knot vector, and revised number of control points new blending matrices are generated. These revised blending matrices are further used to generate a merged, single patch B-spline surface is of the spoon as shown in Fig. 9(c).

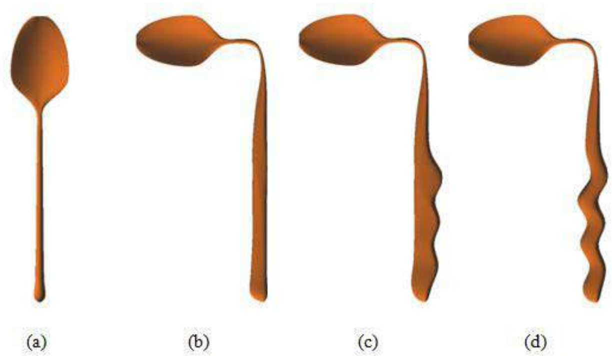

Fig. 10: Investigation of different shapes of a spoon for eating food to accommodate impaired wrist movements or weak grip of fingers. The original design is shown in (a) and design modifications are presented from (b) to (d).

This design can accommodate the lack of wrist movement and the user does not have to rotate his/her wrist. Fig. 10(a) shows the initial design and Fig. 10(b-d) show different variations of the design. The design shown in Fig. 10(b) can accommodate lack of wrist movement of the user. The designs shown in Fig. 10(c-d) provide better grip for different sets of users. These grips can be further modified to suit grip of the user. These designs can be evaluated by an industrial designer or a user in the virtual reality environment.

5.2 Evaluation of Concept

Once an industrial designer comes up with different concepts, these can be evaluated before moving on to the next step of detailed design. The concept can be evaluated in relation to its ergonomic shape and size, suitability for intended user group, strength and weight. Once the concept design is finalized, it can be used to impart training to the user group.

Considering the group of users affected by rheumatoid arthritis, aged people, and children, it will be pertinent to know if these users can eat with the spoon without spilling the food. A B-spline surface patch having 8×8 control points net was used to represent food (jelly) for interaction with the spoon to mimic eating with a spoon. A 12×12×12 mass spring damper mesh was used to incorporate material properties of jelly to this model. Fig. 11(a) shows the B-spline model and Fig. 11(b) shows the mass spring mesh of this model.

(a) B-spline model

(b) Mass spring model shown with facets. The nodes are shown as green dots

(c) Mass spring model. The thick red lines represent the boundaries of hexahedron mesh

Fig. 11: (a) B-spline model representing food (jelly) (b-c) Mass spring mesh to incorporate material properties to the model.

The concept design framework proposed in this paper, make it possible to model food (jelly) as a deformable model and simulate an environment in which a user can interact with food and spoon. Fig. 12(a) shows a spoon with a jelly on it, while the spoon was kept straight.

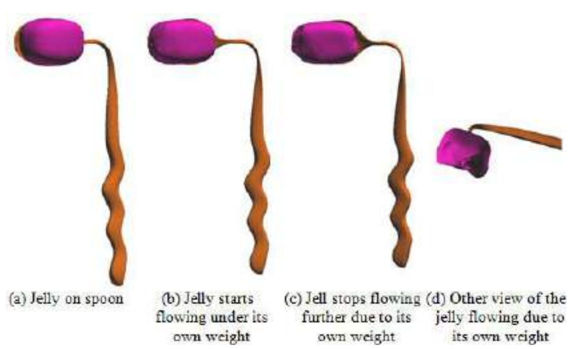

Fig. 12: Simulation of food (jelly) in a spoon without tilting it.

Due to its own weight, jelly started flowing downwards as shown in Fig. 12(b-c). This means that if this design is used, the user cannot eat the jelly of this size without spilling it. Fig. 12(d) shows the close up of the bowl of spoon. It is clear that the jelly is spilling from the front portion of the spoon bowl. At this point, an industrial designer can start modifying the design to so that the jelly does not get spilled. It is clear from Fig. 12(b-d) that the jelly was spilling from the front portion of the spoon bowl while the back portion was empty. The design of the spoon can be modified to see how will the jelly behave if the bowl is tilted about 10 degree in the backward (anti clockwise for the spoon shown in Fig. 12) direction.

Fig. 13(a) shows the jelly put in a spoon with tilted bowl. The jelly starts flowing due to its own weight as shown in Fig. 13(a-d). However, this time it does not spill out of the spoon as shown in Fig. 13(d). The bowl was tilted by 20 degree to see if it further improves the design. However the jelly started flowing from the back part of the spoon. The simulation suggests that the spoon bowl should be tilted by 10 degree to improve the design. In a similar fashion, other parameters of the spoon such as depth of the bowl, curvature of the bowl, or the width of the bowl, can be varied to determine best suitable design.

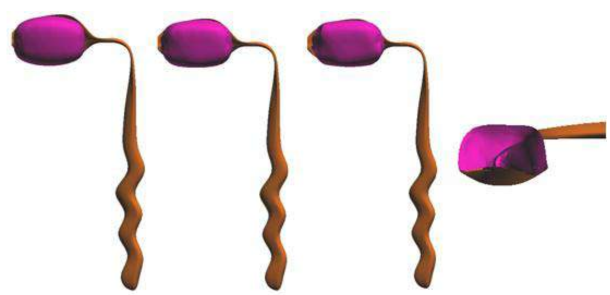

Fig. 13: Simulation of food (jelly) in a spoon when the bowl is tilted by ten degree.

The industrial designer can also determine weight of the spoon, the deflection of the spoon due to its own weight and that of the food, while using different materials for the spoon. The preliminary evaluation of the concept design, in the virtual reality environment would help the industrial designer to come up with the designs which can be successfully implemented. The tools developed in this paper, can provide this environment to the industrial designer.

6 CONCLUSIONS

Product concept generation within a virtual reality environment requires a large variety of interactive tools that enable the user to enhance his or her creativity. The proposed framework was capable of handling virtual models represented as B-spline surfaces. The haptic tools can be represented as a point, an implicit surface, a tessellated surface, and B-spline surface. Hence, a variety of rigid and deformable tools could be used during sculpting or validation of the concept design. No restriction was imposed on the number of control points representing a B-spline surface, degree of the B-spline surface, knot vectors, or extent of deformation. The B-spline surfaces can be merged without imposing restrictions which are generally imposed by commercially available software. Material properties like Young's modulus and Poisson ratio were incorporated into the model while generating mass spring system by volumetric self organizing feature maps. Global and local deformations were possible and were achieved by changing the number of iterations used by deformation algorithm. The mass spring system allowed the concept design framework to efficiently simulate deformation of deformable models. All the components of the framework for interaction with deformable models used pre-computed blending matrices. Once computed, these blending matrices could be used to find the new position of control points to represent deformed B-spline surface without calculating blending functions thereby increasing computational efficiency of the framework. In fact these blending matrices enabled the algorithm to efficiently merge B-spline surface patches, accurately check the collision, and generate nodes for the mass spring system to determine deformation using the physics-based model. Aside from concept product design, the modeling tools presented in this paper can be used for many other applications. The overall concept design module can be used for training in the area of surgery simulation and many cases of rehabilitation. In biomedical applications, soft tissues can be modeled as B-spline surface and assigned appropriate properties using mass spring system.

ACKNOWLEDGEMENTS

This work has been supported, in part, by All India Council for Technical Education New Delhi, India, through research promotion scheme; vide Ref. No. 8023/RID/RPS-12/2011-12 and Natural Sciences and Engineering Research Council of Canada. The authors also acknowledge the support from the National Research Council of Canada in London, Ontario, Canada.

REFERENCES

[1] Anthony, L.; Regli, W. C.; John, J. E.; Lombeyda, S. V.: CUP: A computer-aided conceptual design environment for assembly modeling, The ASME Journal of Computer and Information Science in Engineering, 1(2), 2001, 186-192.DOI: 10.1115/1.1385826

[2] Astheimer, P.; Dai, F.; Felger, W.; Göbel, M.; Haase, H.; Müller, S.; Ziegler, R.: Virtual design II ð an advanced VR system for industrial applications. In: Proceedings of Virtual Reality World '95, Stuttgart, Germany, 1995, 337-363.

[3] Bloomenthal, J.: Bulge elimination in convolution surfaces, Computer Graphics Forum, 16(1), 1997, 31-41.DOI: 10.1111/1467-8659.114

[4] Bowman, D., Conceptual design space ð beyond walk-through to immersive design, (a chapter in Designing digital space), John Wiley & Sons, New York, USA, 1996.

[5] Butterworth, J.; Davidson, A.; Hench, S.; Olano, T. M.: 3DM: a three-dimensional modeler using a head-mounted display. In: Proceedings of the 1992 Symposium on Interactive 3D Graphics, Cambridge, Massachusetts, USA, 1992, 135-138.

[6] Cani-Gascuel, M.-P.; Desbrun, M.: Animation of deformable models using implicit surfaces, IEEE Trans. on Visualization and Computer Graphics, 3(1), 1997, 39-50.DOI: 10.1109/2945.582343

[7] Chapin, W. L.; Lacey, T. A.; Leifer, L.: DesignSpace: a manual interaction environment for computer aided design. In: Proceedings of the Conference Companion on Human Factors in Computing Systems, Boston, Massachusetts, USA, 1994, 33-34.DOI: 10.1145/259963.260001

[8] Cheshire, D.; Evans, M.; Dean, C.: Haptic modeling an alternative industrial design methodology? In: Proceedings of EuroHaptics 2001, Birmingham, UK, 2001, 124-129.

[9] Chu, C. C.; Dani, T. H.; Gadh, R.: Multisensory interface for a virtual reality based computer aided design system, Computer-Aided Design, 29(10), 1997, 709-725.DOI: 10.1016/S0010-4485(97)00021-3

[10] Cotin, S.; Delingette, H.; Clement, J. M.; Bro-Nielsen, M.; Ayache, N.; Marescaux, J.: Geometrical and physical representations for a simulator of hepatic surgery, Studies in Health Technology and Informatics, 29(1), 1996, 139-151. PMid:10163746

[11] Dachille, F.; Qin, H.; Kaufman, A.; El-Sanat, J.: Haptic sculpting of dynamic surfaces. In: Proceedings of Symposium on Interactive 3D Graphics, Atlanta, GA, USA, 1999, 103-110.

[12] Dani, T. H.; Gadh, R.: Creation of concept shade designs via a virtual reality interface, Computer Aided Design, 29(8), 1997, 555-563.DOI: 10.1016/S0010-4485(96)00091-7

[13] Ferley, E.; Cani, M.-P.; Gascuel, J.-D.: Practical volumetric sculpting, Visual Computer, 16(8), 2000, 469-480. DOI: 10.1007/PL00007216

[14] Gregory, A. D.; Ehmann, S. A.; Lin, M. C.: inTouch: interactive multiresolution modeling and 3D painting with a haptic interface. In: Proceedings of the IEEE Virtual Reality 2000 Conference, New Brunswick, New Jersey, USA, 2000, 45-52.

[15] Igwe, P. C.; Knopf, G. K.; Canas, R.: Developing alternative design concepts in VR environments using volumetric self organizing feature maps, Intelligent Manufacturing, 19(6), 2008, 661-675. DOI: 10.1007/s10845-008-0118-0

[16] Knopf, G. K.; Igwe, P. C.: Deformable mesh for virtual shape sculpting, Robotics and Computer Integrated Manufacturing, 21(4), 2005, 302-311. DOI: 10.1016/j.rcim.2004.11.002

[17] McNeely, W. A.; Puterbaugh, K. D.; Troy, J. J.: Six degree-of-freedom haptic rendering using voxel sampling. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, California, USA, 1999, 401 - 408.

[18] Mine, M. R.: ISAAC: a Meta-CAD system for virtual environments, Computer-Aided Design, 29(8), 1997, 547-553. DOI: 10.1016/S0010-4485(96)00095-4

[19] Platt, J. C.; Barr, A. H.: Constraint methods for flexible models, Computer Graphics, 22(4), 1988, 279-288. DOI: 10.1145/378456.378524

[20] Pungotra, H.; Knopf, G. K.; Canas, R.: Efficient algorithm to detect collision between deformable B-spline surface for virtual sculpting, Computer-Aided Design, 40(10-11), 2008, 1055-1066. DOI: 10.1016/j.cad.2008.09.006

[21] Pungotra, H.; Knopf, G. K.; Canas, R.: Framework for modeling and validating conept designs in virtual reality environments. In: Proceedings of the IEEE Virtual Reality 2009 Conference (Symposium on Human Factors and Ergonomics), Toronto, Canada, 2009, 393-398.

[22] Pungotra, H.; Knopf, G. K.; Canas, R.: Merging multiple B-spline surface patches in a virtual reality environment, Computer Aided Design, 42(10), 2010, 847 - 859. DOI: 10.1016/j.cad.2010.05.006

[23] Pungotra, H.; Knopf, G. K.; Canas, R.: Novel collision detection algorithm for physics-based simulation of deformable B-spline shapes, Computer Aided Design & Applications, 6(1), 2009, 43-54. DOI: 10.3722/cadaps.2009.43-54

[24] Pungotra, H.; Knopf, G. K.; Canas, R.: Representation of objects in virtual reality environment using B-spline blending matrices. In: Proceedings of the 21st IASTED Int. Conference on Modeling and Simulation (MS 2010), Banff, Alberta, Canada, 2010, 281-288.

[25] Sachs, E.; Roberts, A.; Stoops, D.: 3-Draw: a tool for designing 3D shapes, IEEE Computer Graphics and Applications, 11(6), 1991, 18-24. DOI: 10.1109/38.103389

[26] Sagar, M. A.; Bullivant, D.; Mallinson, G. D.; Hunter, P. J.; Hunter, I. W.: A virtual environment and model of the eye for surgical simulation. In: Proceedings of the 21st Annual Conference on

Computer Graphics and Interactive Techniques SIGGRAPH '94, ACM, New York, NY, 1994, 205-212.

[27] Sener, B.; Wormald, P.; Campbell, R.: Evaluating a haptic modeling system with industrial designers. In: Proceedings of EuroHaptics International Conference, Edinburgh, Scotland, 2002, 165 -169.

[28] Terzopoulos, D.; Platt, J.; Barr, A.; Fleischer, K.: Elastically deformable models, Computer Graphics, 21(4), 1987, 205 - 214. DOI: 10.1145/37402.37427

[29] The, Arthritis Society, 2010. http://www.arthritis.ca

[30] Waters, K.: A physical model of facial tissue and muscle articulation derived from computer tomography data. In: Proceedings of Visualization in Biomedical Computing (VBC ǐ92), Chapel Hill, N.C., USA, 1992, 574ð583.

[31] Ye, J., Integration of virtual reality techniques into computer aided product design, PhD Thesis, Department of Design and Technology, Loughborough University, Leicestershire, UK, 2005

[32] Ye, J.; Campbell, R.: Supporting conceptual design with multiple VR based interfaces, Virtual and Physical Prototyping, 1(3), 2006, 171-181. DOI: 10.1080/17452750601017129

[33] Ye, J.; Campbell, R. I.: New CAD interfaces for the conceptual design process. In: Proceedings of the 3rd Annual International Conference on Rapid Product Development, Bloemfontein, South Africa 2002, 150 - 162.