Detailed Intermediate Report

An Easy 3D Modeling Tool

COMP4801 Final year project

Department of Computer Science

Faculty of Engineering

The University of Hong Kong

Supervisor

Professor Francis C.M. Lau

TAI, Tsen-Jung, Airy

3035146669

January 21st, 2018

Abstract

The rapid upgrading of 3D printing technologies currently faces the bottleneck of its incapability of gaining popularity, largely due to the absence of 3D modeling tools which is portable and of low technical requirement while still robust. This report introduces an easy online 3D modeling tool, adopting block-based manipulation style and possessing remarkable deformation functions of skinning modeling and digital sculpting. The skinning technique is based on the B-Spline function, which realizes the construction of models with a free-formed side surface. The digital sculpting harnesses ray casting and surface subdivision, reforming the model's surface in detail, with various types of effects for choice. Furthermore, a co-editing feature is planed to be provided, which is a pioneer in the field of geometric modeling. The collaboration is handled by frequent mutual checking and updating between the server side and the client side. At the current stage, skinning modeling is approaching its completion, while interpolating vertices and creating triangular indices list to display the wrapping surface remains as future work. Digital sculpting will be implemented after the winding-up of the curved feature whilst co-editing remains optional depending on the time limit. With these enhancements finished, the flexibility and practicability of the platform can be improved, and subsequently the platform can contribute to the 3D printing's reaching out and application prosperity.

Acknowledgement

I would like to express my sincere appreciation to those who have inspired, encouraged and assisted me during the development of this final year project as well as this completion of report. Special thanks are given to my supervisor Professor Francis C.M. Lau, who has been guiding me from the very beginning towards the current stage, and will keep advising me till the accomplishment of this project, and Dr. Bin Chan, who designed and developed the prototype this project based on, and has generously shared his knowledge and expertise with me.

Table of Contents

List of Figures

1 Introduction

1.1 Background of Online 3D Modeling Tools

Introduced around 30 years ago, 3D printing has gone through rapid developments and improvements with decreasing in cost and increasing in quality, and is believed to cast a revolution in modern society. However, in contrary to the success in the area of technologies and devices, 3D printing suffers from a relatively low adoption rate in today's life [1][2]. One factor blamed for the lack of prevalence is the gap between available 3D printing hardware and obscure modeling software [3]. Most people don not acquire adequate high-level skills to produce complex 3D models that are normally created using advanced desktop applications such as Blender or ZBrush, which consist of overwhelming user interfaces even scaring users away.

The constrains of these traditional applications have give rise to the simplified online 3D modeling platforms and the methods provided can be classified into groups. SketchUp [4] provides sketch-based modeling but users are expected with artistic background [5]. VECTARY [6] allows modification based on single or multiple vertices, lines, and meshes, offer users without art training a stage to begin with, whilst the process is time consuming. Tinkercad [7] runs a block-based style, of which a complicated model is produced by grouping different components, and thus requires relatively low demand on artistic ability. Although these platforms have released the burden of intricate 3D modeling to some extend, there is a general lack of vigorous capabilities of generating free-formed or high-detailed surfaces.

1.2 Objective of the Project

This final year project, an easy 3D modeling tool, is aimed at making an improvement of general online modeling tools that currently exist, in terms of the ability of handling advanced 3D deformation features, applying the modeling techniques of surface skinning and digital sculpting. The technique of co-editing is optional but, to utilize the online feature, also worthy of implementation regarding the enhancement of productivity.

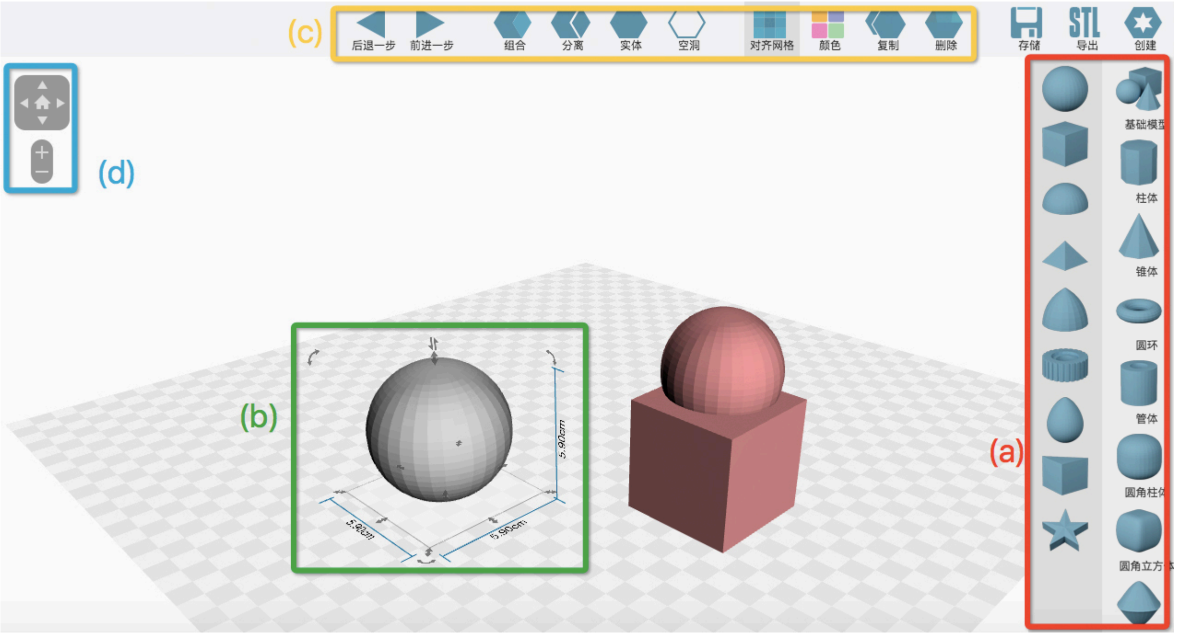


Figure 1.1 A screenshot of the project prototype

This modeling software is based on a prototype developed and provided by Dr. Chan Bin, which is a web project developed with Microsoft Visual Studio, using JavaScript for the frontend and C# for the backend over ASP.NET MVC framework. Similar to Tinkercad, the prototype (Figure 1.1) offers users with (a) basic building blocks, (b) handlers for adjusting a single model, (c) iconic commands for manipulating more than one target, a plane as the floor, a workspace containing rulers or a grid, and (d) a view controller for changing the viewpoint.

The application will be embedded in the corresponding website of the final year project. The website maintains accounts for users so that users can access their previous stored artworks, create new scenes, as well as collaborate on shared creations if co-editing is realized.

The rest of the report starts from a comprehensive elaboration on the methodology used to accomplish the 3D deformation functions and the co-creation service, to an evaluation of the work done and a schedule of the future work.

2 Methodology

2.1 B-Spline Curves, Surfaces and Skinning Models

In this project, also as an extension of the prototype, a B-Spline is used for representing and generating curved features. The prototype has complete logic and algorithms for creating a B-Spline curve using adjustable control points along with a predetermine knot vector and degree. And also the code to turn a closed curve into a flat surface. Surface skinning, also referred as lofting, which is a powerful technique allowing a B-Spline surface constructed by passing throw given B-Spline curves [8], is used specifically to join the set-up of flat surfaces by interpolating a wrapping surface. In practice, users are able to create various flat surfaces and a skinning side surface will be automatically generated. And adjustments of the skinning model is reflected in real time to make the process intuitive. The implement logic flow starts with interpolating the closed curves which are displayed and used to generate flat surfaces to help users gain an idea of the cross sections, then interpolating the wrapping surface with all points of all flat surfaces in reorganized order, and eventually filling the smooth surface with vertices and indices to display. The underlying mathematics and algorithms are briefly elucidated in the following sections.

2.1.1 B-Spline Curves Representation and Interpolation

A B-Spline curve is basically decided by a set of control points ***P***, a knot vector ***U*** and a degree parameter *p*, thus the data points function *C(u)* which represents the points on the curve can be denoted as

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u)\boldsymbol{P}_i$$

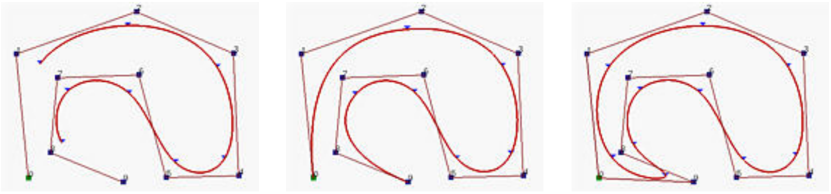
where $N_{i,p}(u)$ are B-spline basis functions defined with degree *p*, which have a recursive relationship denoted as

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

with a terminating condition

$$N_{i,0}(\mathrm{u}) = \begin{cases} 1, & \text{if } u_i \le u < u_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

showing the feature of local impacts that control points can have on the shape of the curve [9].



(a) Open (b) Clamped (c) Closed

Figure 2.1 3 types of B-Spline Curves.

There are generally three types of B-Spline curves distinguished by the characteristic of being open, clamped or closed (Figure 2.1), and the exact form is controlled by how the knot vector is formed.

Normally, a B-Spline curve is calculated using known control points for the benefit that both the implementation of the construction algorithm and the manipulation of creation can be accomplished with ease. Nevertheless, for the purpose of skinning, it is necessary to perform interpolation, a technique with which a curve or surface can be produced according to given data points lying on the target model.

The definition of the curve interpolation is that, given a set of n+1 data points ***D*** and a degree *p*, a B-Spline curve of the same degree and passing through all the data points in order can be found by obtaining the corresponding set of control points ***P*** [9].

The algorithm of the curve interpolation requires auxiliary information about the parameters telling how each data point is fitted (basically, the spacing between neighbor points), and the knot vector. There are diverse methods to calculate the two lists and in this program, the uniformly spaced method is selected for acquiring the parameters while an “averaging” method is used for the knot vector. Synthesizing the calculated n+1 parameter vector ***t*** with the n+1 B-spline basis functions $N_{i,p}(u)$ yields a matrix of $N_{i,p}(\boldsymbol{t}_i)$, and each entry of the matrix can be evaluated using the known data sets of parameters, the knot vector and the degree.

Thus a new equation is produced as

$$\mathbf{D}(t_k) = \sum_{i=0}^{n} \boldsymbol{N}_{i,p}(t_k)\boldsymbol{P}_i \qquad for\ 0 \leq k \ \leq n \qquad \Rightarrow \qquad \mathbf{D} = \mathbf{N} \cdot \mathbf{P}$$

and the control points set can be built by applying a linear system solver [9].

2.1.2 B-Spline Surfaces Representation and Interpolation

A B-Spline surface is constructed by lofting two B-Spline curves, which means that it is defined by a two-dimension sets of control points with m+1 points for its row and n+1 points for its column ***P***, and distinct knot vectors ***U*** and ***V*** and degrees *p* and *q* for u and *v* directions respectively. The mathematics function is presented as

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

which is actually a tensor-product surface. Similar as the features of B-Spline curves, B-Spline surfaces can also be divided into being open, clamped and closed, and can be produced by providing data points on the surfaces for interpolation.

The process remains alike, starting with selecting 2 sets of interpolation reference parameters ***s*** and ***t*** for u and v directions, thus the interpolation equation can be formed as

$$\boldsymbol{D}_{cd}=S(s_c,t_d)=\sum_{i=0}^{m}\sum_{j=0}^{n}N_{i,p}(s_c)N_{j,q}(t_d)\boldsymbol{P}_{i,j}$$

and due to the independence between $N_{i,p}(s_c)$ and $N_{j,q}(t_d)$, the equation can be rewritten as

$$\boldsymbol{D}_{cd}=\sum_{i=0}^{m}\boldsymbol{N}_{i,p}(s_c)\boldsymbol{Q}_{id}$$

where

$$\boldsymbol{Q}_{id}=\sum_{i=0}^{n}\boldsymbol{N}_{j,q}(t_d)\boldsymbol{P}_{ij}$$

which can be interpreted as the d^th^ column of data points $\boldsymbol{D}_d$ is obtained from the d^th^ column of intermediate points $\boldsymbol{Q}_d$ and parameters s_c. This is again a curve interpolation problem and so as calculating $\boldsymbol{P}_{ij}$ from $\boldsymbol{Q}_{id}$. And the algorithm for solving the points set P turns out to be performing curve interpolation twice by inputting appropriate data sets [9].

2.1.3 Manipulation Interface of B-Spline Skinning Models

Apart from the implementation of the algorithms, it is also a significant task to integrate the functions with the prototype's interface. The method of creating B-Spline models in most modeling software in the market is to provide a pen-like tool, which is more of a sketch style. In this project, however, the common method tends to be uncoordinated with the existing system, and anyhow, it requires a higher-level artistic skill of the user and destructs the original intention of adopting a model-based style. Hence, the compromise is to provide a canvas for the user to adjust the cross sections, a method inherited from the creation of B-Spline swiping models implemented in the archetype. Users are allowed to adjust the number and positions of interpolating points, and so as those of the cross sections. A detailed illustration is presented in the third section of work accomplished.

2.2 Digital Sculpting

Digital sculpting is the technique to manipulate the object as if it is made of clay thus allow delicate deformation. It affects the object surface by stretching the polygons or moving the vertices. In this project, the sculpting effect is provided as a brush tool. The vertices inside the brush tip region are activated and moved according to the sculpting effect. Different sculpting options include pulling the vertices out to form a bulge or pressing the vertices in to form a dent. However, the performance of sculpting is limited by the density of polygons.

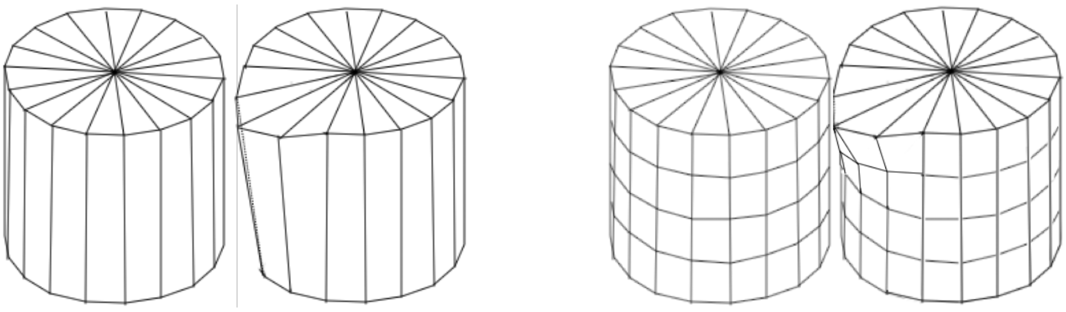


Figure 2.2 Primary building blocks of different polygon density and corresponding sculpting results.

With an object of a low polygon density, moving a single vertex influences a large part of the surface, which fails to accomplish detailed shaping (Figure 2.2). It seems necessary to maintain a high polygon density to obtain an adequate sculpting result. However, in this project, most of the primary building blocks only use a small number of polygons. And it is a waste of memory space and adding meaningless workload of computation due to the maintenance excessive vertices than needed to satisfy the sculpting need which might only occur locally.

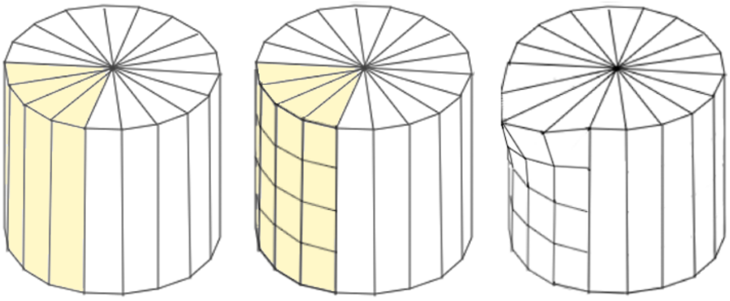


Figure 2.3 Dynamic sculpting in which subdivision occurs only when needed.

Dynamic operator sculpting [10] could be the trade-off between achieving the technique without depleting the data processing capability, in which further-detailed subdivision of meshes is performed only in the area selected by the sculpting brush (Figure 2.3). This reduces the total vertices that need to be stored but preserves the quality of sculpting.

Different sculpting effects are achieved by translating the vertices inside the selected region towards certain directions. The displacement depends on the effect weight which is set by the user as well as the distance between a vertex and the brush center. Pulling and pressing move the vertices along the normal vector of the plane where the brush tip lies, while pinching moves the vertices towards the center of the brush tip and twisting rotates the vertices by some degree.

2.2.1 Ray Casting for Determining Intersection

The application of sculpting is accomplished by moving and pressing a brush tip, which is a circle covering a region where the vertices inside are affected by the sculpting effect. Ray casting is applied to decide whether a vertex lies in the active region. The detailed flow begins with determining the triangular mesh on the model where the mouse-down event occurs, and therefore computing the surface normal of the mesh plane. The brush tip is then generated with the mouse position as the center and the user-defined radius, lying on a surface parallel to and close to the reference mesh. Then a perspective projection of the model on to the circle surface is operated, and those vertices falling on the circle are stored in an active list, along with their perpendicular distance to the surface normal line stored in a distance list. Additionally, a list of vertex normals are computed and maintained. Consequentially, with the coordinates of each vertex, and corresponding distance to the central line which serves as a weight indicating how much degree the effect is applied, and the vertex normals list which serves as the direction of translation, each active vertex is moved by some distance to collectively form a surface deformation. The computation process is iterated with updated input data when each time interval elapsed, until a mouse-up event is detected.

2.2.2 Catmull-Clark Subdivision Surface

To tackle the problem of insufficient amount of triangular meshes during sculpting while maintaining an acceptable calculation speed, a dynamic subdivision which only takes place inside the active region is used and the technique adopted in this project is the robust Catmull-Clark Subdivision. The algorithm is as follows. Firstly, a set of face points is computed by averaging all the control vertices of each triangular mesh, or the so-called face, and a set of edge points is calculated for each edge by averaging the face points of the two adjacent faces sharing it, and the two endpoints of it. Then each vertex is moved to a new position defined by the information of it original coordinates, the average of the face points of the faces it contributes to, and the average of the centers of the edges it is involved with. Eventually, a triangle face is replaced with three new polygon faces each constructed with the new vertex, two edges points and a related face point [11][12]. It is notable that the original arc of the surface will experience minor displacement during the process.

2.2.3 Effects Algorithms

It is the variety of effects that lends strong modeling power to digital sculpting. These include but are not limited to drawing, inflating, smoothing, positive or negative pinching and grabbing. The algorithms of realizing each of them are similar with respective specific modification. The drawing effect is accomplished by moving each active vertex along the same direction of the average normal of all the vertices, while inflating moving each vertex along its own normal [13]. Smoothing is to calculate the average displacement of each vertex to its neighboring vertices and apply the value to the vertex [14]. Pinching is to pull the vertices towards or away from the brush center decided by a user-selected positive or negative mode. Grabbing causes the most dramatic reformation, and distinct to the former effects in which the active vertices keep changing with the new position of the brush tip, those vertices to be affected is defined only once at the mouse-down event, and the mouse moving trajectory is used as the direction to produce new vertices and meshes all the way along [13].

2.2.4 User Interface of Digital Sculpting

By pressing the button of a burin, a brush tip of a circle centered at the mouse position is shown and the shape will be changing according to the mesh normal on which the cursor lies. A canvas with a scale bar for tuning the radius and a dropdown menu for effect selection is displayed. By hitting confirm or cancel buttons at the bottom of the canvas, the deformation can be truly applied or discarded, and the user is allowed to change the radius and the brush type before the application decision made.

2.3 Co-Editing

As an extension to utilize the online features, a service of co-creation is expected to be implemented if time allows. Primarily, the realization of co-creation takes three stages. The first step is to compare and calculate the changes made by the current user compared with the stored version, and transmit the data to the server. Subsequently the server merges all the changes made by all the users and handles the conflicts. After merging and confliction managing, the result is returned to and displayed on the client side.

It is notable that, compared with processing the conflict situation of text or code, the complexity and cost used to handle the conflict cases made on the same object in a modeling project is greatly higher. Besides, aesthetics has no standard, which makes the server difficult to determine which part of meshes should be kept while abandoning the others. An alternative is to schedule the server to perform regular check on each object. If an object is under modification of a certain user, the server locks the object to prevent other users from editing. This ensures no object can be manipulated by more than one user, and causes confusion during the merging step. If an object is left unchanged for some time, the server then compares the object with the old version to determine whether it had gone through any modification, and need to be updated among all the clients.

3 Work Accomplished

As a project with a prototype supplied, there are tasks requested to be deal with before the implementation of the core functions. The first is to understand the code of the project, and the second is to figure out the advanced deformation options that can be added, exploiting the existing data structures and some functions, and can be harmoniously integrated, and thus also understand the underlying geometrical mathematics and algorithms. The core functional part follows the order as organized in the second section of methodologies, including providing B-Spline skinning models, digital sculpting and co-creation service.

3.1 B-Spline Curves and Surface Interpolation

The interpolation algorithms of B-Spline curves and surfaces was implemented in November, which follows the description in the methodologies section, and can be summarized as selecting the interpolation parameters, calculating the knot vector, recursively generating the B-spline basis coefficient matix $\boldsymbol{N}_{i,p}$ and solving the control points set from the equation. The surface interpolation was fulfilled alike except for the double iterations obtaining intermediate reference points set $\boldsymbol{Q}$. Then the generation of closed curves and corresponding flat surfaces using retrieved control points was aided with the existing function provided in the prototype.

3.2 B-Spline Skinning Models Control Interface

The control panel of skinning models was designed and implemented in December, integrating the algorithm finished in November. By selecting the button for B-Spline skinning models, a default skinning model of conical frustum constructed by lofting three B-Spline closed curves is displayed and the user is provided with a canvas containing the default sets of interpolating points (Figure 3.1).

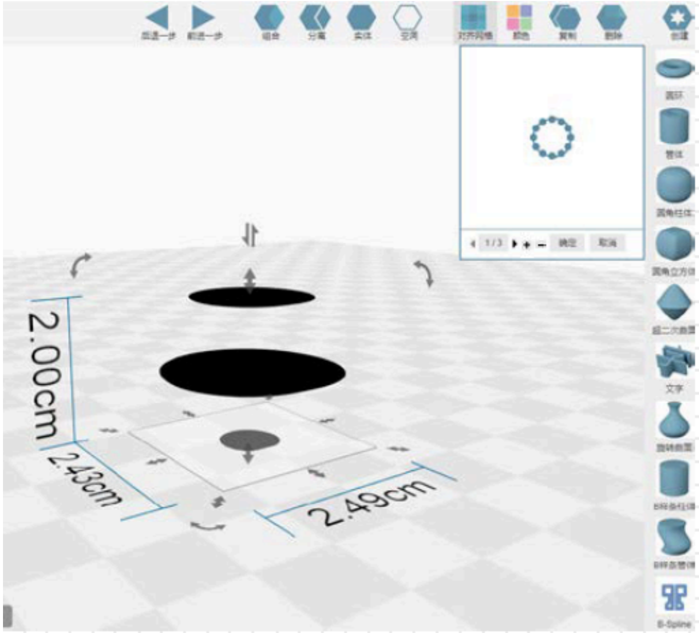


Figure 3.1 A screenshot of the adjustment interface of a skinning model.

The canvas holds a sequence of pages of cross sections, which can be flipped using the arrow handlers at the bottom of the canvas. A cross section can be deleted by navigating to the corresponding holding page and pressing the minus button next to the arrow handlers, and a new cross section inheriting the same configuration as the current selected page can be added by pressing the plus button. Each page is created at a distance of 1-unit length, and can be changed by dragging the cursor on the scale bar. On each page, an interpolating point can be moved via dragging it inside the canvas defined area and can be deleted by selecting it and pressing the minus button on the top-right corner. A new interpolating point can be added by hitting any position on the curve and pressing the plus button.

4 Future Work

4.1 Revised Schedule

The rest of the January will be used to realize the algorithm which interpolates the B-Spline surface function and displays the model in the scene using the consistent structure of lists of vertices and indices. In February, the basic logic of digital sculpting is going to be executed, including ray casting, dynamic subdivision and the control canvas. Till mid March, the different brush effects are planed to have been realized. Thus, web integration and co-editing can be implemented before April and the remaining days are reserved for final testing and debugging, as well as the final deliverables of the report and presentation.

4.2 Problem Encountered

Tiny problems have shown up throughout the development process mainly with the curve interpolation. The interpolated curve, which is expected to be of the closed type, is turned out to be clamped at the starting and ending interpolation point, therefore resulting a flat segment near that point. The problem is suspected to be caused due to the distinct computation methods of the knot vector used during interpolation and later reconstruction using control points (Figure 4.1(a)).

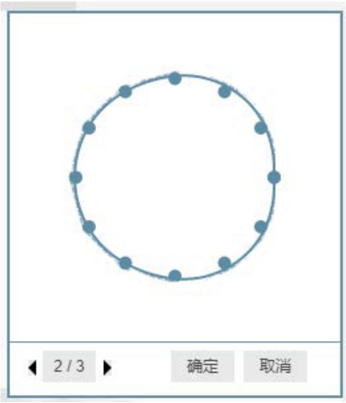


Figure 4.1(a) A screenshot of curve interpolation, with the rightmost dot as the starting as well as ending point.

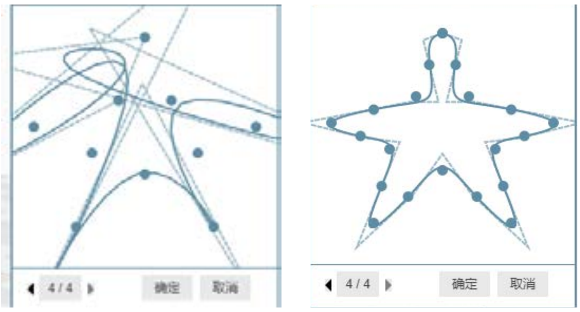


Figure 4.1(b) The interpolation using insufficient points on the right produce unacceptable result.

A potential solution may be applying a modification to the interpolation algorithm to use the same knot vector (which has been tried and seemed to fail, further inspection may be required), or writing a customized control-point-based construction algorithm, which might cause inconsistency between the display of the other B-Spline models and the newly added skinning models. Worse still, an interpolation with high accuracy demands a relatively large number of interpolating points, which might counterproductively increase the user's workload (Figure 4.1(b)). The reason is again supposed to lie with the usage of the existing construction algorithm and a similar solution might be able to alleviate the incorrectness.

The customized building algorithm of B-Spline curves using control points set is planed to be tried in January as well, and if the problems still exist and no other resolution comes up, they would be left and reattempted only until the completion of co-editing.

5 Conclusion

The easy 3D modeling project offers the users with the manipulating functions of high degree of freedom, namely the featured B-Spline skinning modeling and digital sculpting, which are scarcely seen to be simplified and integrated on existing online platforms. Co-editing service is wait-listed to be added if there is still remaining time after the completion of the core functions, so that the flexibility and the practicability of the platform can be extended owing to the convenience of the Internet.

At this stage, although minor problems exist and have been discovered in curve interpolation, a possible solution has been proposed and skinning modeling is close to its success, which would allow the produce of models with free-formed surfaces. The implementation logic and algorithms of digital sculpting have been carefully studied and as completed to be realized. As an auxiliary, co-editing has its methodology examined as well.

The anticipation behind the project is to bridge the gap between the hard-to-acquired 3D modeling skills and easy-to-obtain 3D printing hardware, assist in the spread of 3D printing technologies to be far-reaching, and consequentially contribute to the boom of the personal and customized 3D printing business, ranging from manufacturing industries to academic development.

References

[1] S. Mishra. "3D Printing Technology," Science Horizon, vol. 10, pp. 43-45, October 2014.

[2] B. C. Gross, J. L. Erkal, S. Y. Lockwood, C. Chen, and D. M. Spence, "Evaluation of 3D Printing and Its Potential Impact on Biotechnology and the Chemical Sciences," American Chemical Society, Washington, D.C., Rep. 86-(3240–3253), 2014.

[3] T. Rayna, L. Striukova, J. Darlington, "Co-creation and user innovation: The role of online 3D printing platforms," Journal of Engineering and Technology, vol. 37, pp. 90-102, July–September 2015.

[4] 3D Builder Online Software. (n.d.). SketchUp [Online]. Available: https://www.sketchup.com/products/sketchup-free [Accessed: 2017, Sep 2]

[5] Sketch-based modeling. Wikipedia [Online]. Available: https://en.wikipedia.org/wiki/Sketch-based_modeling [Accessed: 2017, Oct 2]

[6] VECTARY - the free, online 3D modeling software. (n.d.). VECTARY [Online]. Available: https://www.vectary.com/ [Accessed: 2017, Sep 2]

[7] Tinkercad | Create 3D digital designs with online CAD. (n.d.). Autodesk [Online]. Available: https://www.Tinkercad.com/ [Accessed: 2017, Sep 2]

[7] N. M. Patrikalakis, T. Maekawa, W. Cho. (2009, December). *Shape Interrogation for Computer Aided Design and Manufacturing* [Hyperbook]. Available: http://web.mit.edu/hyperbook/Patrikalakis-Maekawa-Cho/. [Accessed: 2017, Sep 17]

[8] Surface skinning revisited. L. A. Piegl, W. Tiller. The Visual Computer, vol. 16, pp. 273-283, June 2002.

[9] B-spline. Available: https://rosettacode.org/wiki/Catmull%E2%80%93Clark_subdivision_surface [Accessed: 2017, Nov 13]

[10] Raul Fernandez Hernandez, "Dynamic Subdivision Sculpting," Hermanos Saz University.

[12] Catmull Clark Subdivision. Available: http://www.rorydriscoll.com/2008/08/01/catmull-clark-subdivision-the-basics/

[13] Available: http://docs.pixologic.com/user-guide/3d-modeling/sculpting/sculpting-brushes/smooth/ [Accessed: 2017, Sep 2]

[14] Available: https://www.Tinkercad.com/ [Accessed: 2017, Nov 8]