A Simple Browser-based 3D-Sketching Framework for Novice and Infrequent Users

Frode Eika Sandnes[1,2] and Yuriy Lianguzov[2]
[1]Faculty of Technology, Art and Design, Oslo and Akershus University College of Applied Sciences, Oslo, Norway
[2]Faculty of Technology, Westerdals School of Art, Communication and Technology, Oslo, Norway
Frode-Eika.Sandnes@hioa.no, yuriylianguzov@gmail.com

***Abstract*—Modelling in 3D is considered time-consuming and difficult, requiring special training. There are currently no 3D modelling user interface conventions. This paper thus presents a framework for simple 3D modelling based on 2D drawings of height maps. The framework is intended to be quick and easy to use for untrained users and no special software is needed as the application runs in the browser.**

Keywords—3D sketing; 3D modelling; design; ideation

I. Introduction

Sketching is used in ideation where designers rapidly sketch down their ideas and share these with others. Sketches are typically quick and cheap to make and should appear unfinished making them distinguishable from the final product. They are therefore often hand-drawn in 2D [1]. It may also be useful to sketch in 3D [2], for example if designing virtual reality environments, games or learning resources.

It is relatively easy to learn perspective-sketching skills where objects can be viewed either from the outside or inside. However, such sketches are static. To achieve more interactive sketches with the user immersed in the sketch one needs a panorama sketch or a 3D model [3, 4]. Generating 3D models is time-consuming and require certain skills to operate the 3D software. Some thus view 3D modelling software as unsuitable for sketching since it takes too long to generate a sketch, while ideas appear and disappear instantaneously [5].

Moreover, design is often a collaborative effort performed in teams where the design team members bounce ideas off each other, while 3D modelling is an individual activity that hinders collaborative team activity.

Finally, the results of 3D modelling software usually appear “professional” clean and can easily be mistaken for the final product, while a true sketch often have the unfinished organic drawing strokes of a human hand [1]. Many attempts have been made at improving the input of 3D models [6], including freehand sketching [7], projective 2D strokes [8], using movable handheld devices [9] and inferring 3D shape from gray level shading [10], to mention a few.

This work attempts to narrow this gap. A tool is presented that allows designers to use their 2D drawing skills and the tool will generate a 3D model from the 2D drawing. The process is interactive allowing users to refine theirs models. The novel contribution herein is the use of hues to control height as hues are uniquely disguisable by the human visual system.

II. The 3D Modelling Framework

The framework comprises three main components, namely a 2D drawing editor area for drawing contours, contour height controls and the model viewer (see Figs 1, 2, 3 and 4).

A. Contour Drawing Editor

The contour drawing editor was implemented as a simple and conventional 2D drawing application (see Fig. 1). The application contains conventional tools for drawing simple primitives such as rectangles and circles. The user can select colors from a predefined palette of colors.

The *x* and *y* dimensions of the drawing editor corresponds to the *x* and *y* dimensions in 3D space. Next, color is used to control the *z* dimension of each point. Color is used to signal a specific height from cold colors to warm colors. Hues are easier to distinguish (see Fig. 4) than the absolute intensity of a shade of gray, a technique used in previous research [10].

B. Contour Heigh Controls

A set of contour height controls allows the user to adjust the heights of the various contour levels. Each height control is controlled with a slider, one slider per palette hue (see Fig. 2). Each hue can thus be controlled individually within the bounds of a cube (defined by the minimum and maximum slider values). By default the sliders are set such that the heights defined by the hues increase from low to high in even steps.

C. 3D Model Viewer

The 3D model viewer allows the user to inspect the rendered model by rotating the model using the mouse.

D. Contour Interpolation

The framework implements contour height interpolation allowing the user to specify a smooth transition between different heights in the model. Gray is used to indicate an interpolated area. Each point in this area is interpolated relative to the heights of its nearest edge points.

To interpolate between two levels one starts by drawing a large disc in one hue defining the base level, then add a gray disc on top but inside the first disc and finally a smaller disc in yet a different hue to define the top level of the shape (see Fig. 1). The gray pixels thus appear as a circular frame with a given width. It is the pixels in the gray area what will create a smooth transition from the base level to the top level.

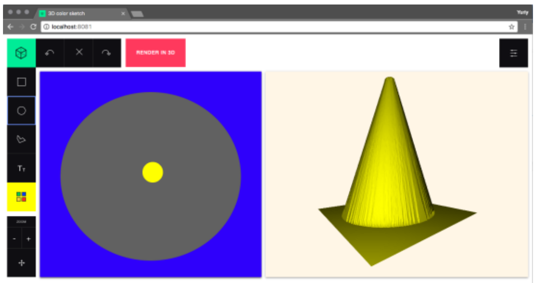

Fig. 1. Drawing and viewing a cone with a gray interpolation area.

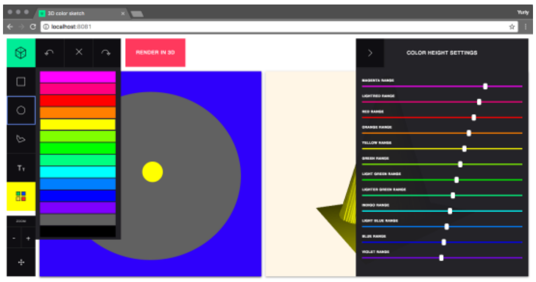

Fig. 2. Adjusting the contour heights.

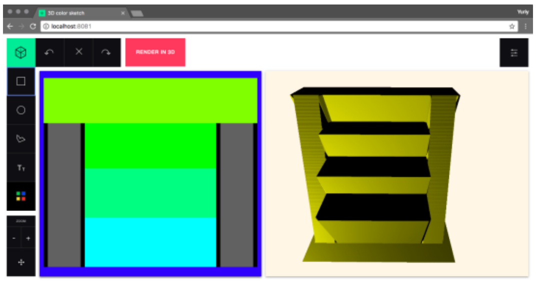

Fig. 3. A staircase. Black lines are used to the limit interpolation.

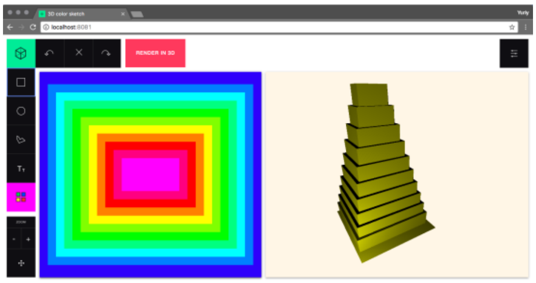

Fig. 4. Mexican pyramid. Hues are easily perceivable.

E. *Stops*

The interpolation is computed relative to the nearest non-gray edge. However, there are situations where one does not want the interpolation to be affected by a given edge. To prevent this the user can use black lines to separate the edge from the gray interpolation area (see Fig. 3).

To model a ramp one needs to draw stops. First the top and bottom plateaus are drawn in different hues. Then, the area between the top and bottom edge is filled with gray to indicate the interpolation. However, to prevent the interpolation to be affected by the edges on the left and right side the left and right edges are separated by two black lines, one on the left, and one on the right side of the gray ramp rectangle (see Fig. 3).

F. *Implementation*

The framework is implemented in client-side JavaScript and HTML5. The three.js library was used for the 3D model viewer and an open source JavaScript drawing editor was altered for the purpose of this project. The visual elements, apart from the sliders, are realized using a HTML-canvas.

III. Conclusions

A simple browser-based framework for rapidly creating 3D models was presented. A web-browser was chosen as the framework platform with the aim to make it easy for novices to explore and learn basic 3D modelling and designs as well as allowing designers to quickly set up improvised ideation workshops. The framework relies on a visual language where hues denote height, grey indicate interpolation and black stops interpolation. The framework is only capable of creating certain open shapes with only one z value per x and y value. Future work involves testing the system on designers to see if the tool is easier to use than conventional 3D modelling applications. The Tool is openly and freely available at http://www.cs.hioa.no/~frodes/3D.

References

[1] F. E. Sandnes and H.-L. Jian. "Sketching with Chinese calligraphy," Interactions, Vol. 19, No. 6 (2012): 62-66.

[2] X. Chen, "An integrated image and sketching environment for archaeological sites," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),. IEEE, 2010.

[3] F. E. Sandnes, "Communicating Panoramic 360 Degree Immersed Experiences: A Simple Technique for Sketching in 3D," International Conference on Universal Access in Human-Computer Interaction. Springer International Publishing, LNCS Vol. 9738, Springer, pp. 338-346, 2016.

[4] F. E. Sandnes, "PanoramaGrid: A Graph Paper Tracing Framework for Sketching 360-degree Immersed Experiences," Proceedings of the International Working Conference on Advanced Visual Interfaces. ACM, 2016.

[5] A. Black, "Visible planning on paper and on screen: The impact of working medium on decision-making by novice graphic designers," Behaviour & Information Technology 9.4 (1990): 283-296.

[6] L. Olsen, et al. "Sketch-based modeling: A survey," Computers & Graphics, Vol. 33, No. 1 (2009): 85-103.

[7] K. Kondo, "Interactive geometric modeling using freehand sketches," Journal for Geometry and Graphics, Vol. 13, No. 2 (2009): 197-209.

[8] O. Tolba, J. Dorsey, and L. McMillan. "Sketching with projective 2D strokes," Proceedings of the 12th annual ACM symposium on User interface software and technology. ACM, 1999.

[9] M. Xin, E. Sharlin, and M. C. Sousa. "Napkin sketch: handheld mixed reality 3D sketching," Proceedings of the 2008 ACM symposium on Virtual reality software and technology. ACM, 2008.

[10] R. Zhang, et al. "Shape-from-shading: a surve,." IEEE transactions on pattern analysis and machine intelligence, Vol. 21, No. 8 (1999): 690.