Quick and Easy 3D Modelling for All:
A Browser-based 3D-Sketching Framework

Frode Eika Sandnes(✉)
Oslo and Akershus University College of Applied Sciences, Oslo, Norway
Frode-Eika.Sandnes@hioa.no

Yuriy Lianguzov
Westerdals School of Art, Communication and Technology, Oslo, Norway

Abstract—Modelling in 3D is considered time-consuming and difficult, relying on special skills. To make matters worse, there are currently no user interface conventions for how to work with 3D data and different modelling applications adopt varying interaction techniques. 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. An experimental evaluation confirmed that all of the eight participants successfully managed to complete simple modelling tasks in just a few minutes.

1 Introduction

Sketching is a valuable design tool especially useful during ideation. A sketch allows designers to rapidly represent their ideas on paper and share these with others. Ultimately, sketches should be quick and cheap to produce and they should have an unfinished look. The unfinished look means that they are not mistaken as the final product. Sketches are therefore often hand-drawn in 2D [1] giving them an organic appearance. They are not limited to flat drawings. For instance, if designing virtual reality environments [2], first person games or 3D learning resources [3], 3D sketches [4] may prove useful.

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, such as in augmented reality [5] one needs a panorama sketch or a 3D model [6, 7], but the vantage position is fixed [8].

3D modelling is laborious, slow, and depends on software operation skills. Black [9] concludes that 3D modelling software is unsuitable for sketching since the software slows down the designer and he or she may lose an important idea before it is

captured. Moreover, design is usually a collaborative team effort where the members bounce ideas off each other. In contrast, 3D modelling is a solo activity.
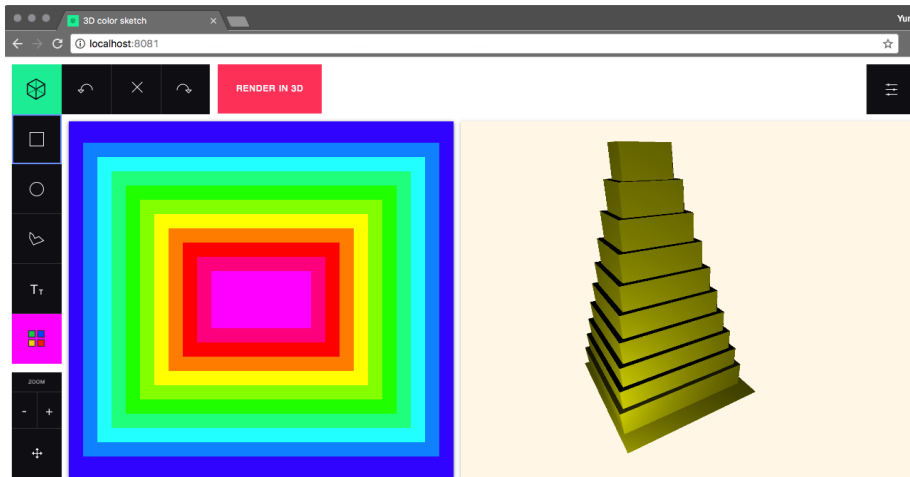

Fig. 1. Modelling a Mexican pyramid with different colored heights.

The 3D models built with software usually appear professional and streamlined. There is thus a risk that they can be confused with the final product, while a true sketch often has the unfinished, hasty, imperfect and organic strokes of a hand drawing [1]. Many attempts have been made at improving the input of 3D models [10], including freehand sketching [11], projective 2D strokes [12], using movable handheld devices [13] and inferring 3D shape from gray level shading [14], combining multiple flat sketches [15], to mention a few.

This paper therefore presents a tool that was designed with the goal of being easier to use relying on users' natural 2D drawing skills and familiarity with 2D drawing software. It is browser based and therefore accessible to all from everywhere. It therefore also facilitates online experimentation and learning [16]. The novel contribution herein is the use of hues to control height, as hues are uniquely disguisable by the human visual system. This study extends an earlier version of this paper [17] with the addition of a usability experiment.

2 3D modelling framework

The 3D modeler consists of three parts: a 2D drawing editor for drawing contours using different colors, contour height controls and the model viewer (see Figures 1 and 2).

2.1 Contour editor

The contour editor is a simple conventional 2D-drawing application (see Figure 1). It offers a small set of tools for drawing basic geometric shapes such as boxes and discs. Colors are selected from a fixed palette of 12 colors.

The x and y coordinates on the canvas correspond to the x and y coordinates in 3D space. Next, color controls the z coordinate of each point. This coding allows the 2D image to be transformed into a 3D point cloud [18]. Color is used to signal a specific height from cold colors to warm colors.
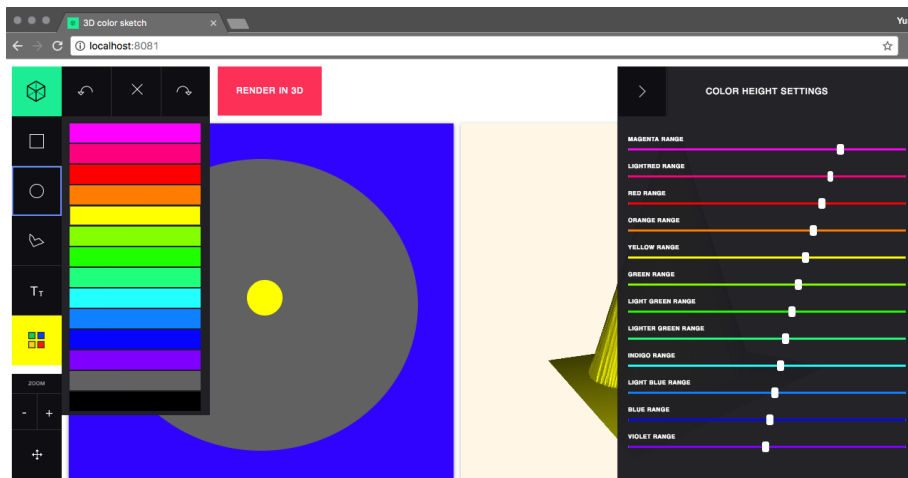

Fig. 2. Modelling a cone with different colored heights.

Hues are easier to distinguish than the absolute intensity of a shade of gray [19], a technique used in previous 3D-modelling research [14].

2.2 Contour height controls

The user adjusts the heights of the various contour levels using an array of sliders. Each height is controlled by one slider (see Figure 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 contour heights vary in uniform steps.

2.3 3D model viewer

The user inspects the rendered model in the 3D model viewer by rotating, panning and zooming the model using the mouse.

2.4   Contour interpolation

The framework implements linear 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. The technique is reminiscent to Gouraud polygon shading [20].

For example, to draw a smooth cone one interpolates 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 Figure 2). The gray pixels thus appear as a circular frame with a given width. It is the pixels in the gray area that will create a smooth transition from the bottom to the top.
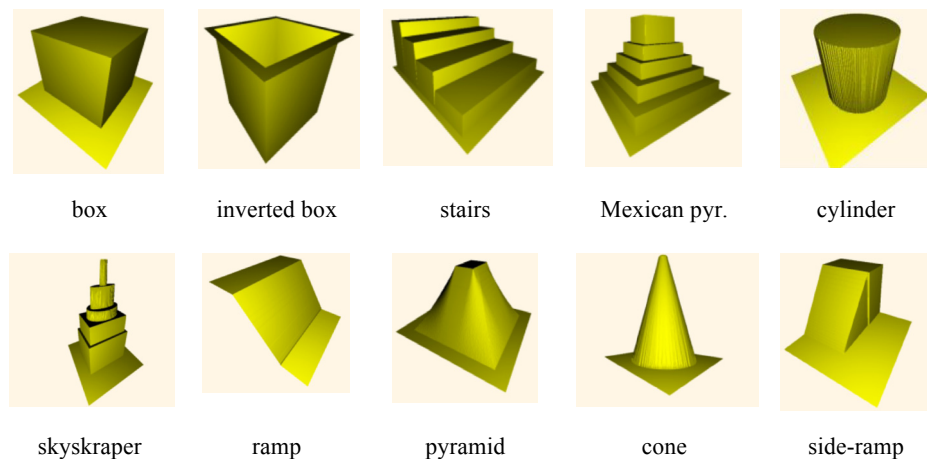

| box | inverted box | stairs | Mexican pyr. | cylinder |

| skyskraper | ramp | pyramid | cone | side-ramp |

Fig. 3. Design tasks used in the experiments.

2.5   Stops

The interpolation is computed relative to the nearest non-gray edge. However, occasionally one does not want the interpolation to be affected by a certain edge. Black lines are used to separate the edge from the gray interpolation area to disconnect the edge from the interpolated area.

A ramp is easily modelled using stops. First, one draws the top and bottom platform as rectangles in different hues. Then, the rectangle between the top and bottom platforms is filled with gray to mark the interpolation. However, to prevent the interpolation to be affected by the edges on the left and right side two black lines are used to isolate the interpolated area from the left and the right edges. Each of the two black lines are placed on the left and right side of the gray interpolated area.

2.6 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. All the other user interface elements were realized sing HTML and CSS.

3 Experimental evaluation

A simple within-subjects experiment was conducted to confirm whether the tool could be used without any prior knowledge of 3D modelling. Eight male participants in the range from 25-20 years were recruited. They were professional web-designers and developers with no specific experience in 3D modelling.

Each participant was asked to complete 10 simple design tasks individually and in isolation. The time to complete the task was the observed dependent variable and the type of task was the independent variable.
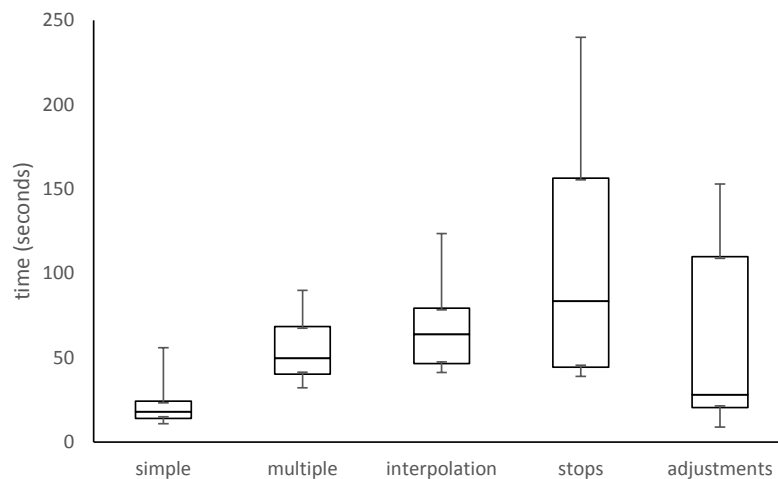

Fig. 4. Box plot showing the distributions of time to complete each task type.

The ten tasks involved designing a simple box, modifying a box into an inverted box, designing a staircase, a Mexican pyramid, a cone, a skyscraper, a ramp, a pyramid, a cone and a side-ramp (see Figure 3). These tasks can be classified as single level shape, multiple level shape, interpolation, interpolation with stop, and modification.

The participants were asked to talk-aloud during the session commenting. Each participant was asked to complete a questionnaire with seven seven-scale Likert questions addressing ease of use, learnability, usability, ease of recovering from errors (robustness), satisfaction with the results, whether they would use this tool for future modelling tasks and overall satisfaction.

4   Results

Figure 4 shows the distribution of times to complete each task type. A Mauchly test indicates that the assumption of sphericity is violated ($p < .05$). A non-parametric repeated measures Friedman test was therefore applied. This test revealed that there was a statistically significant difference in time to perform the five types of modelling tasks ($\chi^2(4) = 14.1$, $p = .007$). Median (IQR) seconds to complete each of the task types were simple 18.5 (12.8 to 26.8), multiple 49.5 (35.5 to 73.0), interpolation 64.0 (45.5 to 83.7), stops 83.5 (41.5 to 205.5) and adjustments 28.0 (19.5 to 110.0).

Post hoc analysis with Wilcoxon signed-rank tests were conducted. There were significant differences between single and multiple levels ($Z = -2.240$, $p = .025$), between simple and interpolation ($Z = -2.524$, $p = .012$) and simple and stops ($Z = -2.524$, $p = .012$). There was no significant difference between interpolation and interpolation with stops ($Z = -1.260$, $p = .208$) and interpolation and multiple levels ($Z = -0.980$, $p = .327$). Moreover, there was no significant differences between simple shapes and adjustments ($Z = -1.963$, $p = .05$) or interpolation and adjustments ($Z = -0.420$, $p = .672$).

If a Bonferroni correction is applied the significance level needs to be set at $p < 0.005$, suggesting that none of the paired comparisons were significant.

5   Discussion

All the participants successfully completed all the tasks. The longest task-completion time was 240 seconds (4 minutes) and the shortest task-completion time was 23 seconds (nearly half a minute). As shown by the results it takes longer to complete more complicated tasks than simple tasks. Single level models are the quickest, followed by models with multiple levels. Tasks involving interpolation take longer and the ones involving stops took the longest also with the largest range. This suggests that interpolation with stops is the most difficult. Height modifications were associated with some of the shortest time-completion times but also a very large range.

The results of the questionnaire show that the participants exhibited a positive impression of the tool as only the top end of the Likert scale (5 to 7) was used. The only exception was for whether the participant would use this tool again 5 (1 to 7). This is probably because the tool was a prototype and not a finished product.

6   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 heights, grey indicates interpolation and black

isolates interpolation. The framework is limited to certain shapes with only one z value for each x and y coordinate. An experiment revealed that all eight participants managed to design simple shapes within a few minutes. The tool is available at http://www.cs.hioa.no/~frodes/3D.

7 References

[1] Sandnes, F. E., & Jian, H. L. (2012). Sketching with Chinese calligraphy. Interactions, 19(6): 62-66. https://doi.org/10.1145/2377783.2377796

[2] Xia, P., Lopes, A. M., & Restivo, M. T. (2013). Virtual reality and haptics for dental surgery: a personal review. The Visual Computer, 29(5): 433-447. https://doi.org/10.1007/s00371-012-0748-2

[3] Jin, X., & Sun, Y. (2015). Path Planning Analysis of Decelerator Assembly Based on Virtual Technology. International Journal of Online Engineering (iJOE), 11(4): 30-33. https://doi.org/10.3991/ijoe.v11i4.4617

[4] Chen, X., Morvan, Y., He, Y., Dorsey, J., & Rushmeier, H. (2010,). An integrated image and sketching environment for archaeological sites. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE, pp. 35-42.

[5] Sandnes, F. E., & Eika, E. (2017). Head-Mounted Augmented Reality Displays on the Cheap: A DIY Approach to Sketching and Prototyping Low-Vision Assistive Technologies. Lecture Notes on Computer Science, 10278: 167-186. https://doi.org/10.1007/978-3-319-58703-5_13

[6] Sandnes, F. E. (2016). Communicating Panoramic 360 Degree Immersed Experiences: A Simple Technique for Sketching in 3D. Lecture Notes on Computer Science, 9738:338-346. https://doi.org/10.1007/978-3-319-40244-4_33

[7] Sandnes, F. E. (2016). PanoramaGrid: A Graph Paper Tracing Framework for Sketching 360-degree Immersed Experiences. In Proceedings of the International Working Conference on Advanced Visual Interfaces, ACM, pp. 342-343. https://doi.org/10.1145/2909132.2926058

[8] Sandnes, F. E., & Huang, Y. P. (2016, October). Translating the viewing position in single equirectangular panoramic images. In 2016 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, pp. 389-394. https://doi.org/10.1109/SMC.2016.7844272

[9] Black, A. (1990). Visible planning on paper and on screen: The impact of working medium on decision-making by novice graphic designers. Behaviour & Information Technology, 9(4): 283-296. https://doi.org/10.1080/01449299008924244

[10] Olsen, L., Samavati, F. F., Sousa, M. C., & Jorge, J. A. (2009). Sketch-based modeling: A survey. Computers & Graphics, 33(1): 85-103. https://doi.org/10.1016/j.cag.2008.09.013

[11] Kondo, K. (2009). Interactive geometric modeling using freehand sketches. Journal for Geometry and Graphics, 13(2): 197-209.

[12] Tolba, O., Dorsey, J., & McMillan, L. (1999). Sketching with projective 2D strokes. In Proceedings of the 12th annual ACM symposium on User interface software and technology, ACM, pp. 149-157. https://doi.org/10.1145/320719.322596

[13] Xin, M., Sharlin, E., & Sousa, M. C. (2008). Napkin sketch: handheld mixed reality 3D sketching. In Proceedings of the 2008 ACM symposium on Virtual reality software and technology, ACM, pp. 223-226. https://doi.org/10.1145/1450579.1450627

[14] Zhang, R., Tsai, P. S., Cryer, J. E., & Shah, M. (1999). Shape-from-shading: a survey. IEEE transactions on pattern analysis and machine intelligence, 21(8): 690-706. https://doi.org/10.1109/34.784284

[15] Sandnes, F. E. (2017). Sketching 3D Immersed Experiences Rapidly by Hand through 2D Cross Sections. Lecture Notes in Networks and Systems, 22 (in press).

[16] Restivo, M. T., & Cardoso, A. (2013). Exploring Online Experimentation. International Journal of Online Engineering (iJOE), 9(S8): 4-6. https://doi.org/10.3991/ijoe.v9iS8.3448

[17] Sandnes, F. E., & Lianguzov, Y. (2017). A simple browser-based 3D-sketching framework for novice and infrequent users. In Experiment@ International Conference (exp. at'17), IEEE, pp. 155-156. https://doi.org/10.1109/EXPAT.2017.7984422

[18] Gomez, J. V., & Sandnes, F. E. (2012). RoboGuideDog: Guiding blind users through physical environments with laser range scanners. Procedia Computer Science, 14: 218-225. https://doi.org/10.1016/j.procs.2012.10.025

[19] Sandnes, F. E. (2016). Understanding WCAG2.0 Color Contrast Requirements through 3D Color Space Visualization. Studies in Health Technology and Informatics 229: 366-375.

[20] Gouraud, H. (1971). Continuous shading of curved surfaces. IEEE transactions on computers, 100(6): 623-629. https://doi.org/10.1109/T-C.1971.223313

8   Authors

Frode Eika Sandnes is a professor in Computer Science at Oslo Akershus University College of Applied Sciences and Westerdals Oslo School of Art, Communication and Technology. He has served one term as pro-rector for Research and internationalization. Dr. Sandnes has been instrumental in the establishment of the first universal design master specialization in Norway.

Yuriy Lianguzov holds a master in Applied Computer Science from Westerdals Oslo School ACT and is currently working as a web-developer in Oslo.