Supporting Collaborative Ideation Through Freehand Sketching of 3D-Shapes in 2D Using Colour

Frode Eika Sandnes[1,2(✉)], Yuriy Lianguzov[2],
Osmar Vicente Rodrigues[3], Henrik Lieng[1], Fausto Orsi Medola[3],
and Nenad Pavel[1]

[1] Oslo and Akershus University College of Applied Sciences, Oslo, Norway
{frodes,Henrik.Lieng,nenad.pavel}@hioa.no
[2] Westerdals Oslo School of Art, Communication and Technology,
Oslo, Norway
yuriylianguzov@gmail.com
[3] School of Architecture, Arts, and Communication,
Sao Paulo State University (UNESP), Bauru, Brazil
{osmar,fausto.medola}@faac.unesp.br

Abstract. The modelling of 3D shapes is a challenging problem. Many innovative approaches have been proposed, however most 3D software require advanced skills that hinders collaboration and spontaneous ideation. This paper proposes a novel framework that allows designers to express their ideas in 3D space without extensive training such that they can reuse their 2D sketching skills collaboratively in teams.

Keywords: Collaborative design · Sketching · 3D-modelling · Color-map

1 Introduction

Sketching is a technique used by designers to rapidly represent and communicate ideas. Sketches are often freehand paper drawings allowing ideas to be quickly captured as ideas may leave ones head as quickly as they emerge [1]. Operating computer for making sketches is often too cumbersome and time-consuming and are often thus hindrance to the creative process [2] and collaborative design teams. Sketches may take many forms, including 3D objects. For example, product designers may create 3D sketches using flat perspective drawings.

Sometimes, one wants a more immersive representation than a simple one-view rendering. It may for instance be a panoramic image or sketch of some three-dimensional space that allows viewers to observe a scene in all directions [3–5] using spherical coordinates [6], or it may be some artefact to be explored through virtual reality. To create such experiences designers have to use some modelling tool. This process is often time-consuming, require special software operation experience and the resulting models may look like the finished products when rendered photo-realistically.

Y. Luo (Ed.): CDVE 2017, LNCS 10451, pp. 123–134, 2017.
DOI: 10.1007/978-3-319-66805-5_16

During the last two decades, courses related to the practice of drawing both 2D and 3D, including 3D modeling, has vanished from fundamental and high school curricula. This becomes a huge shortcoming in the development of students' abilities and tools needed in any creative process. Moreover, this is a challenge for society, in particular for educational boards and teachers worldwide that need new tools, methods and new approaches in the development of their course contents.

Thus, students who become the future professionals, is less able to visualize and physically interpret 3D space, such as the reading of 2D representations including drawings, sketches and renderings, and the right visualization of any 3D representation including physical objects, sculptures and products.

The novel contribution of this paper is a sketching framework that allows designers to quickly represent imperfect three-dimensional shapes. The sketch is drawn as two images. The first image represents the origin of the shape in a two-dimensional flat plane with its texture. The second image provides the shape information as height.

2 Background

A study of physical 3D modeling [7] involving three groups of participants with different levels of knowledge and experience in such area (professionals with over five years of experience, design students and those without any such knowledge), evaluated the capacity of the participants to translate and interpret 2D drawings in 3D physical models. The results were compared to results from a similar study conducted 21 years earlier. The results showed that all the participants had difficulties in translating the 2D information into 3D physical models. The students came first followed by the experienced professionals and those without knowledge. This phenomenon was especially noticeable during the start of the sculpturing phase where one searches for the basic shapes of the models. The results are quite different from those observed in the 1989 study, where the participants without any knowledge came first following by the students in second. The overall level of difficulty was lower compared to the results of the 2010 study. Therefore, the results suggest that offering students a narrower contact with courses focusing on 2D and 3D representation has contributed to reinforcement of this problem in Brazil, in particular, where this study took place.

In a study of the design studio application of visual media in the design collaborative groups [8] the researchers concluded that the design workflow including both CAD and sketching have advantages over isolated digital and manual workflows. Accordingly, the seamless transition between sketch and digital media seems to be beneficial especially for design novices because of the translation from tacit knowledge to explicit actionable knowledge. Furthermore, the communication in design studios benefit from a mixed method design process, such as initiating a quick conceptual sketch followed by detailed conceptual analyses. The current CAD solutions offer sharp transition between sketching, digital conceptualization and analyses.

The literature on sketching in 3D is vast and several innovative techniques have been proposed. Most of the techniques are based on modelling directly in 3D [9, 10] by somehow creating and shaping objects from simple primitives and placing these in a scene. Other approaches allow 3D shapes to be constructed from curves [11]. There are also domain specific tools that limit some of the choice provided by the general purpose modelling tools, and may thus be easier to use. For example, Ijiri et al. [12] proposed a 3D sketching tool for flower construction where the user first create a crude initial sketch and then the sketch is gradually refined where components are reused. Such tools are easier to use and can be used to generate very complex models, but the range of possible models are very limited.

Since the operation of 3D modelling tools often is difficult and require training several researchers have attempted to turn original line drawings on paper into 3D models. Most of these methods document concept implementations that can convert very simple sketches [13] and acknowledge that this is a difficult problem. Varley et al. [14] concludes that the success of such systems depends on the number of lines in the original sketch and whether the sketches represent certain basic shapes, as there are many shapes that it is easy for humans to recognize but very hard for machines.

Another approach is to sketch on top of 3D-dimensional views, being it real images, virtual reality or augmented reality, and then infer the three dimensional models from the 2D projective sketch and information about the scene geometry [15]. Example applications include annotation and sketching in archeological sites [14] and sketching and modelling of cartoon like scenes for animation [17]. Ambiguities in where points in the 2D sketches are in the 3D model can be resolved fixing the points using multiple views of the scene [16].

Simplifications can be made such as in the Harold system [18] where the goal is not to make photorealistic models, but rather understandable 3D environments that can be navigated. They introduced three drawing modes, namely billboard, terrain and floor. The billboard mode allow the user to edit planar sketches in the environment, or billboards. These billboards remain flat in the scene but are affected by the perspective projection as the user moves around the scene. They represent recognizable flat representations of objects in the scene, although the objects themselves are not three-dimensional. The terrain mode allows the ground to be modelled with hills and valleys, and the floor mode is used to model the floor.

Common for animation and sketching is that it is not necessary to make accurate and complete models, but rather sufficient scenery to create an experience. In addition to flat or billboards, curved canvases has also been proposed as objects that are easily drawable by 2D means to create scenery suitable for animation and “film-sets” and then these canvases can be placed anywhere in the scene [19]. Another approach for sketching 3D experiences is to sketch projective scenes from various angles and then combine these into panoramic images that can be viewed with panoramic viewers [20, 21]. Such 3D experiences are though only observable from one point and no actual 3D information is captured.

More interactive methods has also been proposed such as the Napkin sketch [22] where a napkin is placed on a table and the world is viewed through a tablet computer with a camera. The tablet tracks the napkin to gain information about the observation point of the tablet. The designer then draw on top of the touch screen showing the view. The stroke information can then be combined with the scene information to build the model. Moreover, the user can immediately move around the model.

A totally different approach is the use of database systems containing existing models [23, 24] where a sketch is made of a scene, and then through manual intervention the scene is broken into objects. Relevant objects are found in a model database. Finally, the computer helps place these objects back in the original world according to the sketch.

Some of the 3D modelling methods that are focused on 2D input include those that uses planar cross sections of the scene or objects. Then multiple planar sections can be combined in various placement of the 3D scene to obtain the 3D model [25, 26]. Various perspective views of an object can also be used to build a 3D model of an object. This for instance has been used for generating 3D models of cartoon characters based an artist's renderings from different angles [27].

Reliefs has also been mapped onto three dimensional objects using line drawings [28] where the 2D dimensional line drawing controls minor surface offsets on the three-dimensional object.

The relief approach can be considered a special case of shading based modelling where a shade is used to control the height of an object's surface [29]. Another approach is to use a shade of gray to indicate height of a surface where white means no offset, medium gray some offset and black max offset. One practical approach employing this scheme took simple line sketches as input, the system would come with a first suggestion to a height map based on shading which then the designer could adjust and edit through paining operations before the final model is rendered into the 3D model [30]. Another attractive prospect of shading based modelling method is to create models from photographs [31].

One problem with shading based approaches is that humans are unable to objectively asses the absolute intensity of a tone as neighboring colors affect each other. Simultaneous contrast occur as the effect that the same level of gray is perceived differently if it is surrounded by darker gray than when it is surrounded by a lighter gray. For this reason this study instead attempts to use fixed colors instead of shades of gray based on the assumption that it is easier to distinguish the main color classes than shades of gray.

3 The Proposed Method

3.1 Assumptions and Motivation

The motivation of this work is to allow people without 3D perspective drawing experience to make 3D illustrations. The framework is not intended to be as accurate and general as state of the art modelling software.

This work leans on the observations that modelling software require training and are generally hard to use. The proposed framework was designed to be used independently of specific software packages, hence allowing users to rely on skills they are already familiar with, namely drawing and sketching on flat two-dimensional surfaces. A rationale for 2D sketches is that they can be produced fast and thereby facilitate rapid and spontaneous ideation processes.

User indicate heights directly in the sketch. Unlike previous approaches that use gray-levels, the current approach uses a palette of distinct color hues. This is because it is hard for humans to determine the absolute intensity of a color [32–34]. Colors, on the other, hand are easier to recognize. These colors represent discrete height levels.

To allow for smooth shapes represented by the values between height-levels, a visual gradient semantic is proposed. This semantic allows smooth height transitions between the various levels for arbitrary shapes, where the level of smoothness can be controlled. The gradients are made automatically as it is challenging to manually make gradients for arbitrary shapes. Moreover, it is very hard to control the nature of the gradient to achieve the desired 3D shape. This is because it is very difficult to visualize the mapping between a gradient and the corresponding shape in 3D space. Another advantage is that it is easy and quick to alter the shape sketch.

3.2 Sketching Language

A shape can comprise a texture image, a shape image or both. The texture image is simply a direct representation of what will be painted onto the object. In our implementation, the color white is used to code transparency in the texture.

The shape image defines the height variations in the z-dimension by default. The user simply uses colors and shades of gray to define the height contour. In the current implementation the 12 hues on the color wheel defined by the projection of the color cube was used to define the various heights. The colors of the color wheel were representing equally spaced heights along the z-dimension. Yellow was defined as base height at 90° on the color wheel. Warm colors defined positive heights relative to the base, and cold colors defined negative heights.

It is difficult to manually create gradients between the hues that correspond to desired smooth transitions. Instead, shades of gray were used to define areas of gradients. That is, a gray area between two different colors are defined as an area of gradual transition from one color to the other. To avoid a transition being affected by certain color regions, black is used to define no transition and is instead replaced by the nearest color.

The level of gray controls the smoothness of the gradient. Dark gray signals linear interpolation and light gray signals smooth interpolation, where the degree of brightness is related to the degree of smoothness.

The texture image and height image are dependent on each other, and each of these can be used as basis for overlay tracing with respect to each other. Overlay tracing will ensure that the content of both the texture image and shape image are aligned.

3.3 Preprocessing Height Maps

The height maps are preprocessed [35, 36] before the 3D model is generated. First, checks are made to ensure that the texture image and the height map images have the same dimensions. If they are different, the height map image is resized to match that of the texture map using an off-the-shelf resampling algorithm.

Second, the image is quantized into discrete hues and shades of gray to emphasize the discrete steps and eliminate inaccuracies incurred by drawing applications. The saturation of each pixel is used to determine if it is color or grayscale.

3.4 Gradients Algorithm

The gradient algorithm is similar to the classic Gouraud shading algorithm, but considers arbitrary shapes. The algorithm for detecting the gradients first scan all the pixels to find all color pixels that is the neighbor of a gray or gradient pixels. These border points represent the color pixels around a given gradient area. These points are organized according to color.

A list of color pixels neighboring black areas are also extracted, and these points are the surrounding pixels of a black area. The border pixels between black and grey areas are not recorded, that is the boundary between the gradient area and blocking areas.

Next, the gradient pixels are filled as follows. For each pixel x, y the closest edge pixel for each color category is detected, and then two largest edge pixels are selected, namely pixel x_1, y_1 with color c_1 and pixel x_2, y_2 with color c_2. The color c of the gradient pixel at x, y is thus computed by interpolating between color c_1 and c_2 according to the distance between the gradient pixel and the two border pixels. More exactly, the linearly interpolated color was

$$c = \frac{c_1 d_2 + c_2 d_1}{d_1 + d_2} \tag{1}$$

where d_1 and d_2 where computed using

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \tag{2}$$

Linear interpolation was used for grays centered at value 192. For a softer interpolation the smoothstep function, Ken Perlin’s 6 h order step function and a 7th order polynomial step functions were used.

Hues are represented on the color wheel from 0 to 360°. Since the height origin is located at 90° the following H(x) transformation was used before the interpolation of the color values and again used to convert back to the color wheel representation:

$$H(x) = \begin{cases} 90 - x, x < 270 \\ 450 - x, x \geq 270 \end{cases} \tag{3}$$

Finally, black pixels are filled with the color of the closest edge pixel.

3.5 Model Building

Finally, the model is built as follows. For each pixel $i. j$ on the texture and height maps, the corresponding color point [x, y, z, c] in space is generated, where $x = i \cdot \delta, y = j \cdot \delta,$

$z = H(height(i,j)\frac{thickness}{360}$ and $c = hexture(i.j)$. Here, δ is the unit distance between consecutive points in the model space. If the width of the object in physical space id with, then δ = $width/x_{pixels}$, where x_{pixels} is the number of pixels in the texture along the horizontal direction.

Next, *thichness* is the maximum height bound of the object, *height*(x, y) is the pixel value of x, y in the height map and *texture*(x, y) is the pixel value at x, y in the texture map. Note that white pixels in the texture maps are considered transparent and not included in the final set of points. Sets of four neighboring points make up polygons, namely $p_{i,j}$, $p_{i+1,j}$, $p_{i+1,j+1}$ and $p_{i,j+1}$ with the color of $p_{i.j}$.

4 Case Studies

A height map interpreter and model synthesizer was implemented in java. The PLY format was used to represent the 3D models as polygon meshes. The models were rendered using CloudCompare.

Figure 1 shows a simple example of the modelling technique where the height map comprises a red background with yellow handwritten text, the yellow and red are both on the warm side of the color wheel and relatively close. The resulting image shows this text as a highened relief on top of the flat plane. A wood texture was also used in this example giving this example the impression of a carved wooden plate.

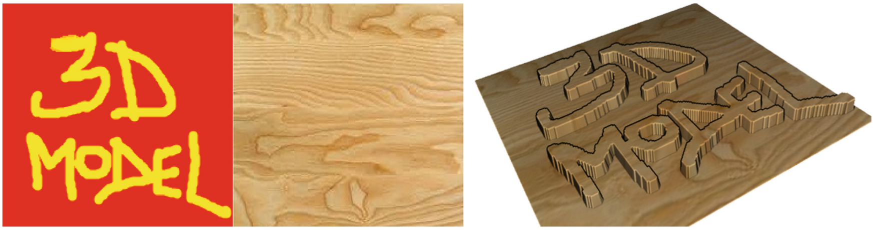

Fig. 1. Text engraving (Color figure online)

Figure 2 (top) shows an example of constructing a set of stairs. The ground is modelled with the blue and each consecutive step is modelled using a color up on the color wheel from light blue, via cyan to two shades of green. The bottom example illustrate the use of interpolation where there soft ramps are created on each side of the stairs. The soft ramp is indicated using a medium gray color. To ensure that it only interpolates between the top stair marked in green and the floor the two black lines are used to separate against the sides and the other stairs.

Figure 3 shows how to modelling a bathtub. First, the top edge of the bathtub is modelled using magenta and the tub bottom is modelled using blue. A rounded rectangle is used to get the roundedness of a bathtub. Then, a lighter gray is used to indicate a smooth interpolation between these two levels. Next, a uniform light brown is used for texture and white is used to cut out the top of the tub and the drain. Therefore, only the brown pixels in the texture map are included in the model.

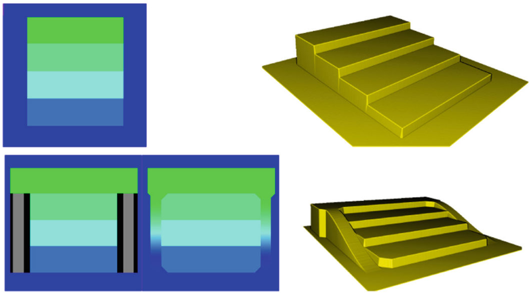

Fig. 2. Stairs (Color figure online)

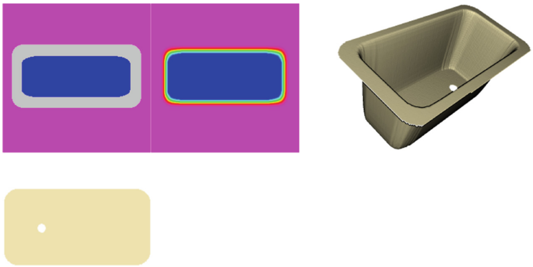

Fig. 3. Bathtub (Color figure online)

Figure 4 shows how to model a chair. First, the chair is modelled using the height map. Blue is used to model the floor as a thin line around the edges of the image and the seat itself is modelled using green, which is in the middle of the height scale. The backrest is modelled using magenta creating the highest top of the chair. A medium dark gray is used to specify smoothed interpolation from the ground to the seat.

The area of the interpolation is made wide allowing the decoration of the chair legs to be modelled more accurately. The first chair is decorated with a uniform red texture. The middle chair has four distinct legs cut out using white and the final chair is decorated with a more elaborate pattern and brown legs.

5 Experimental Evaluation

An experiment was carried out to test the hypothesis that the proposed framework simplifies 3D modelling. Eight male participants working as web-developers were recruited. None of the participants works with 3D. The experiment comprised ten tasks presented in increasing order of complexity involving designing a cube, open box, staircase, Mexican pyramid, cylinder, skyscraper, ramp, pyramid, cone and ramp with stops. A complete and interactive JavaScript version of the framework running in a browser was used for the testing. No texture mapping and only linear interpolation was included in the tasks. The participants were tested individually. They were given instructions and time to familiarize themselves with the tool.

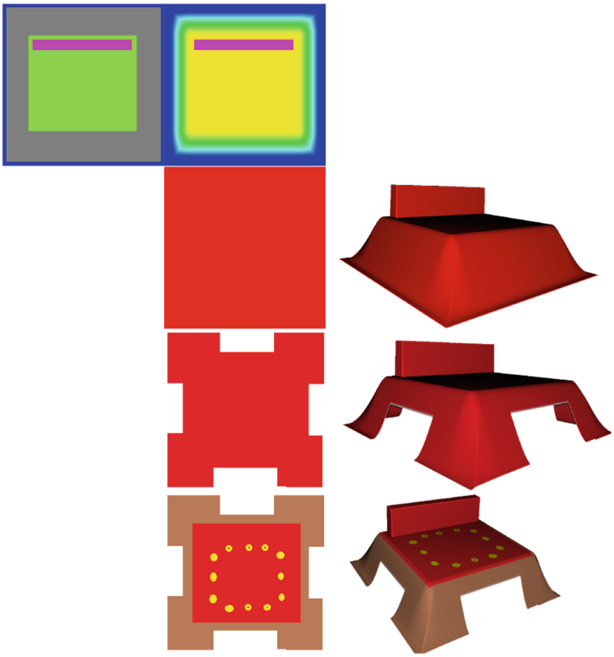

Fig. 4. Chair (Color figure online)

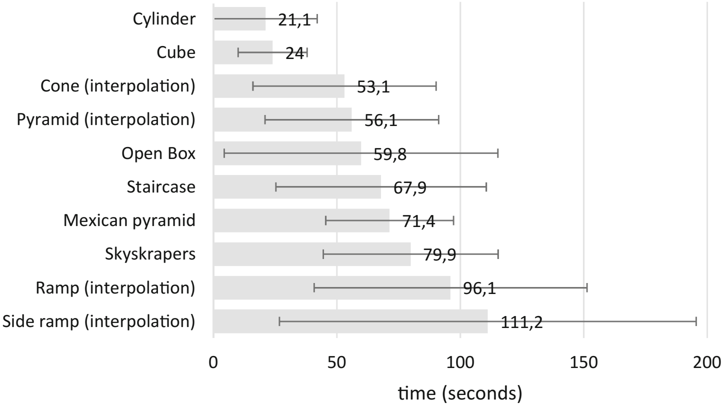

Fig. 5. Time to complete 3D modelling tasks (seconds). Error bars show SD.

All the participants managed to perform all the tasks and Fig. 6 shows that all participants managed to design the skyscraper, cylinder and Mexican pyramid on the first attempt. The most difficult task was the ramp with stops. Here, five of the participants made a total of 12 reattempts before successfully completing the task.

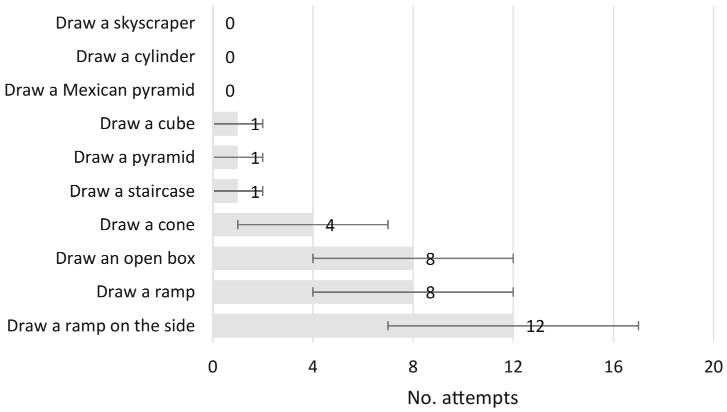

Fig. 6. Number of attempts per task. Error bars shows number of participants with reattempts.

Figure 5 shows that the cylinder and the cube were the fastest models to create both taking less than 30 s on average. The mean time to complete the remaining tasks increased gradually from 53.1 s for the cone to 111.2 s for the side ramp. Clearly, the time to complete the tasks correspond with the complexity of the task. Note also that the task completion times varied across the participants. However, a within-subjects repeated measures anova reveals the time to complete the ten tasks were statistically different ($F(9, 63) = 3.49$, $p < .001$).

The participants were also asked to complete a subjective survey after the session. The results support the hypothesis that the tool is perceived as easy to use. On a seven point Likert scale, the participants responded as follows: Easy to use ($M = 6.1$, $SD = 0.6$), easy to learn ($M = 6.4$, $SD = 1.1$), easy to recover from mistakes ($M = 6.4$, $SD = 1.1$), satisfaction with the results ($M = 5.9$, $SD = 0.6$) and satisfaction with the tool ($M = 6.1$, $SD = 0.6$). Clearly, all the responses are in the high end of the scale.

6 Conclusions

A simple method for modelling 3D objects using intuitive 2D sketches was presented. Shapes are specified using a color height map. Interpolation allows for smooth transitions between different height-plateaus of the model. Gray levels are used to control the smoothness of the interpolation. The models are decorated with textures where white is used to specify transparency and is used to make cuts and holes in the objects. The models can be generated with any drawing program and does not rely on any particular 3D modelling software. Closed objects such as spheres cannot be modelled.

The proposed method combined with digital stylus has a potential to be integrated into a design process and workflow as an initial quick conceptual sketch tool, whose outputs can be later used in more advanced computer software for analyses. The results of the conducted experiment show that users are able to get grasp of declarative command knowledge [37]. By introducing the pre-learned hot-cold color metaphor and gradients, a novice may have a better chance of effective adoption of specific procedural knowledge. Further testing is necessary to study acquisition of strategic knowledge of this CAD software and its adoption in a design workflow.

Acknowledgement. This work is supported by and SIU UTFORSK grant UTF-2016-long-term/10053.

References

1. Sandnes, F.E., Jian, H.-L.: Sketching with Chinese calligraphy. Interactions **19**, 62–66 (2012)
2. Black, A.: Visible planning on paper and on screen: the impact of working medium on decision-making by novice graphic designers. Behav. Inf. Technol. **9**, 283–296 (1990)
3. Sandnes, F.E.: Communicating panoramic 360 degree immersed experiences: a simple technique for sketching in 3D. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2016. LNCS, vol. 9738, pp. 338–346. Springer, Cham (2016). doi:10.1007/978-3-319-40244-4_33
4. Sandnes, F.E.: PanoramaGrid: a graph paper tracing framework for sketching 360-degree immersed experiences. In: Proceedings of the International Working Conference on Advanced Visual Interfaces, pp. 342–343. ACM (2016)
5. Sandnes, F.E., Huang, Y.P.: Translating the viewing position in single equirectangular panoramic images. In: Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 389–394. IEEE (2016)
6. Sandnes, F.E.: Determining the geographical location of image scenes based on object shadow lengths. J. Sig. Process. Syst. **65**, 35–47 (2011)
7. Rodrigues, O.V.: Modelagem de veículos utilizando espuma rígida de poliuretano expandido. In: Ensaios em Design – Ensino e Produção de Conhecimento, pp. 128–157. Canal 6 (2011)
8. Ibrahim, R., Rahimian, F.P.: Comparison of CAD and manual sketching tools for teaching architectural design. Autom. Constr. **19**, 978–987 (2010)
9. Olsen, L., Samavati, F.F., Sousa, M.C., Jorge, J.A.: Sketch-based modeling: a survey. Comput. Graph. **33**, 85–103 (2009)
10. Kondo, K.: Interactive geometric modeling using freehand sketches. J. Geom. Graph. **13**, 195–207 (2009)
11. Das, K., Diaz-Gutierrez, P., Gopi. M.: Sketching free-form surfaces using network of curves. In: EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2005)
12. Ijiri, T., Owada, S., Igarashi, T.: Seamless integration of initial sketching and subsequent detail editing in flower modeling. In: Computer Graphics Forum, vol. 25. Blackwell Publishing, Inc. (2006)
13. Cruz, L.M.V., Velho, L.: A sketch on sketch-based interfaces and modeling. In: 2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T). IEEE (2010)
14. Varley, P.A.C., Martin, R.R., Suzuki, H.: Can machines interpret line drawings. In: EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling, vol. 1 (2004)
15. Turner, A., Chapman, D., Penn, A.: Sketching space. Comput. Graph. **24**, 869–879 (2000)
16. Chen, X., et al.: An integrated image and sketching environment for archaeological sites. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE (2010)

17. Bourguignon, D., Cani, M.P., Drettakis, G.: Drawing for illustration and annotation in 3D. In: Computer Graphics Forum, vol. 20. Blackwell Publishers Ltd. (2001)
18. Cohen, J.M., Hughes, J.F., Zeleznik, R.C.: Harold: a world made of drawings. In: Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering. ACM (2000)
19. Fei, G., et al.: 3D animation creation using space canvases for free-hand drawing. In: Proceedings of the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry. ACM (2008)
20. Tolba, O., Dorsey, J., McMillan, L.: Sketching with projective 2D strokes. In: Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology, pp. 149–157. ACM (1999)
21. Tolba, O., Dorsey, J., McMillan, L.: A projective drawing system. In: Proceedings of the 2001 Symposium on Interactive 3D graphics. ACM (2001)
22. Xin, M., Sharlin, E., Sousa, M.C.: Napkin sketch: handheld mixed reality 3D sketching. In: Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology. ACM (2008)
23. Shin, H., Igarashi, T.: Magic canvas: interactive design of a 3-D scene prototype from freehand sketches. In: Proceedings of Graphics Interface 2007. ACM (2007)
24. Xu, K., et al.: Sketch2Scene: sketch-based co-retrieval and co-placement of 3D models. ACM Trans. Graph. **32**, 123 (2013)
25. Dorsey, J., et al.: The mental canvas: a tool for conceptual architectural design and analysis. In: 15th Pacific Conference on Computer Graphics and Applications, PG 2007. IEEE (2007)
26. Sandnes, F.E.: Sketching 3D immersed experiences rapidly by hand through 2D cross sections. In: Proceedings of REV 2017. LNCS (in press)
27. Triki, O., Zaharia, T.B., Preteux, F.J.: 3D virtual character reconstruction from projections: a NURBS-based approach. In: Electronic Imaging 2004. International Society for Optics and Photonics (2004)
28. Kolomenkin, M., et al.: Reconstruction of relief objects from line drawings. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2011)
29. Gingold, Y., Zorin, D.: Shading-based surface editing. ACM Trans. Graph. **27**(3), 9 p. (2008). Article no. 95. https://doi.org/10.1145/1360612.1360694
30. Feng, X., Shi, M.: Shading-based surface modeling and editing. In: 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE), vol. 2. IEEE (2012)
31. Zhang, R., et al.: Shape-from-shading: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **21**, 690–706 (1999)
32. Sandnes, F.E.: Understanding WCAG2.0 color contrast requirements through 3D color space visualization. Stud. Health Technol. Inform. **229**, 366–375 (2016)
33. Sandnes, F.E., Zhao, A.: A contrast colour selection scheme for WCAG2. 0-compliant web designs based on HSV-half-planes. In: Proceedings of SMC 2015, pp. 1233–1237. IEEE (2015)
34. Sandnes, F.E., Zhao, A.: An interactive color picker that ensures WCAG2.0 compliant color contrast levels. Procedia-Comput. Sci. **67**, 87–94 (2015)
35. Huang, Y.-P., Chang, T.-W., Chen, J.-R., Sandnes, F.E.: A back propagation based real-time license plate recognition system. Int. J. Pattern Recognit. Artif. Intell. **22**, 233–251 (2008)
36. Huang, Y.P., Hsu, L.W., Sandnes, F.E.: An intelligent subtitle detection model for locating television commercials. IEEE Trans. Syst. Man Cybern. Part B **37**, 485–492 (2007)
37. Chester, I.: Teaching for CAD expertise. Int. J. Technol. Des. Educ. **17**, 23–35 (2007)