SKETCH: An Interface for Sketching 3D Scenes A Technical Report

Ankita Christine Victor
International Institute of Information Technology, Bangalore
IMT2014005
Ankita.Victor@iiitb.org

***Abstract*—SKETCH is an application developed to enable the approximate sketching of a concept with the idea that visual representations are communicated more effectively to a third party without the need for precise communication and specialised knowledge. SKETCH was one of the first systems to try to replicate 2D gestural drawings to create a 3D model. The system uses a simple gesture based interface with strokes on a grid. Once an object is drawn it can be transformed using gesture strokes and interactors. SKETCH was designed to be menu free and depict models in a 'sketchy' style i.e., as though an actual rough sketch drawn by hand to convey a basic idea and not the precise details. SKETCH had a huge impact on ideas concerned with 2D to 3D modelling ad gesture recognition based modelling.**

***Keywords*—Sketching, 3D Modelling, Gestures, Strokes, Interactors.**

I. Introduction

Sketching is frequently used during the early stages of conceptual design when ideas are still in the process of being thought out [1]. The process of conceptualization is characterized by a general lack of precision and a tolerance for approximations. Sketching has thus been recognized as an important tool for communicating ideas and concepts [1]. Traditionally sketching has been done using pencil and paper without exploiting the abilities and conveniences of computer systems. Pencil and paper tools make it considerably harder to improve upon, edit and look at designs in different perspectives. Computer models are not subject to such limitations but can be harder to get used to. SKETCH was designed to bridge the gap between hand sketches and computer-based modelling techniques, combining some of the features of pencil-and-paper sketching and some of the features of CAD systems to provide an easy-to-use, gesture-based interface with strokes and interactors to approximate 3D polyhedral modelling [2].

II. Related Work

Much work at the time rarely involved 2D sketching to construct 3D models. Artifices Design Workshop allowed direct 3D construction but still used a menu-oriented interface (WIMP) with limited primitives set [3]. SKETCH on the other hand did not believe in the use of menus and icons. While SKETCH attempted to interpret 2D gestures one at a time, Wang and Grinstein's work focused on the interpretation of 2D line drawings all at once [4]. Viking was an application where the system, based on the user's 2D input, attempted to generate a set of constaints and appropriately construct 3D objects to satisfy these constraints. The only drawback was the uncertainty of finding suitable constraints or arriving at buggy constraints [5]. Rubine [6] used gesture recognition to create a 2D drawing program, but SKETCH was the first to apply gesture recognition with 3D modelling. SKETCH used constrained transformation techniques similar to those described by Bukowski and Sequin [7] where contraints are generated from an object's semantics, only SKETCH requires a user to explicitly define constraints and has less semantic information.

III. Background

SKETCH was conceptualized at a time when there existed no convenient, computerised tools to create design mockups and it was simpler to use pencils and paper. This is evident from the lines:

> Traditionally, people have attacked conceptual design with paper and pencil, not with computers, even though computer models offer numerous advantages. The reasons for this include the low overhead of a single-tool interface (pencil), the lack of special knowledge needed to draw, the ease with which many kinds of changes can be made, and the fact that precision is not required to express an idea [2]
> .

3D modelling tools at the time were not friendly enough to allow the average designer the expertise needed to use their systems. Moreover 3D computerized modelling was considered only for high precision models and not rough designs. The authors speak of related work that were are generally restricted to polygonal objects, often slow and difficult to implement, intolerant of noisy input, unable to find a reasonable 3D solution or result in an unexpected solution, limited by a menu-oriented interaction style and not considering the construction and editing of full 3D scenes [2]. No systems at the point had considered the use of gesture recognition for 3D modelling. The purpose of SKETCH was to give a user clear and direct modelling techniques, with minimum processing lags, input devices and no menu options. SKETCH was the first system to use gesture recognition via sequences of strokes and interactors to model 3D objects.

IV. Discussion

A. Interface

To maintain a simple interface SKETCH used a three button mouse and the Shift-key for all interaction versus using a menu based application. A stroke is pixel-track on the display made using the first mouse button. A simple click draws a dot. A click and immediate drag draws a line along the principle axes of the world. A click and drag with a tearing motion draws a line that is not axis aligned. A click followed by a delayed drag draws a freehand curve. A click and drag with the Shift-key pressed draws a curve on the surface of an object in the scene. Gestures can be repeated by clicking using the first mouse button to drag and drop [2]. The stroke gestures defined by SKETCH resemble computerized drawing tools we use today.
Interactors to transform objects use the second mouse button. The third mouse button is used to manipulate the camera including rotation.

B. Creating Geometry

SKETCH interprets three gesture strokes that meet at a corner to be the specifications for a cuboid (Figure 1). To recognise gestures SKETCH tries to force shapes to be axis-aligned. SKETCH also supports gestures for cones, cylinders, spheres, objects of revolution, prisms, extrusions, ducts and superquadrics. SKETCH maintains this limited set of recognisable primitives to minimize the cognitive overhead required to get familiar with the gestures needed for each polygon, again with the focus being on maintaining a simple interface.

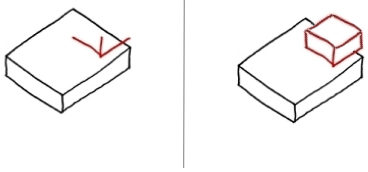

Fig. 1. A series of strokes is drawn (left). The resultant 3D object is drawn [2].

C. Placing Geometry

Objects is placed so that the its features project onto the strokes made on the screen. For this every gesture has a most important vertex (the trivalent vertex for a cuboid, the first vertex of the two parallel strokes that indicate a cylinder). A ray is traced through this vertex to hit a surface at some point. The object is then rendered so that this vertex is placed at the intersected surface point (Figure 2).

New objects are created in conatct with an already drawn object wherever possible. The direction of strokes and the normal of the surface in contact indicates CSG subtraction. If a stroke is made into an existing object then it is subtracted (intuitively this can be thought of as scooping out an object from an existing object) (Figure 3).

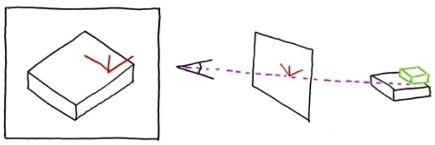

Fig. 2. The vertex is drawn (left). The vertex is projected to determine placement of the object [2].

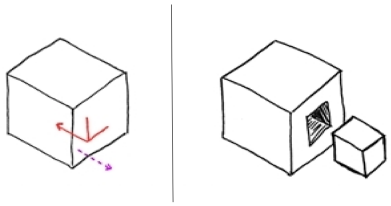

Fig. 3. A stroke is drawn into the object (left). Cuboid is removed [2].

A T junction indicates that a line segment representing the edge of one surface ends somewhere along a line segment representing the edge of another surface. T junctions generally signify that one surface blocks the other from view (occlusion). To indicate the occlusion of one surface by another, a gesture is first placed in the scene and a ray is sent out along the gesture line towards the T junction. If the ray along the gesture intersects the object that defines the top of the T junction and the normal at this point of intersection is approximately opposite to the direction of the ray, then the gesture edge is extended to meet the surface (since it is actually hidden). If the ray does not intersect the the surface, then its object is translated along the viewing vector toward the viewer until its edge exactly meets the end of the gesture edge i.e., there is no occlusion in this case. If the end of the gesture edge is never met (because it was farther away from the viewer), then neither the gesture, nor the existing objects are modified [2].

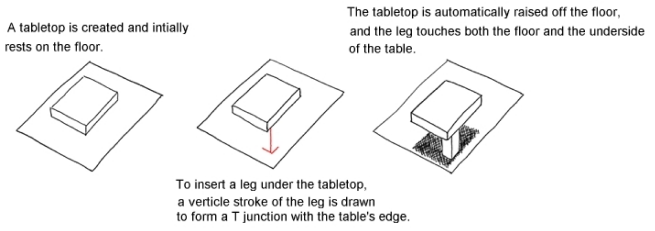

Fig. 4. T junctions for object placement [2].

D. Editing

SKETCH recognises a resizing operation when two strokes are made coincident with each other, in opposite directions and along an existing edge.

A user can draw shadows by first stroking a dot on an object and using the Shift-key while stroking shadow lines.

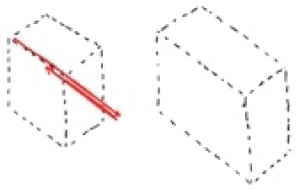

Fig. 5. The point at which the two coincident lines meet indicates the resize [2].

E. Constrained transformation

Objects can be translated by clicking the second mouse button and dragging. Motion can be constrained to rotational motion. SKETCH uses an interaction handler that maintains information about which direction the object is constrained to translate or rotate along. The current constraint is specified by a particular gesture and remains persistent until a new one is specified. The user first makes the constraint gesture followed by the interaction. For example to translate along an axis the user first strokes a constraint axis. Clicking and dragging parallel using the interactor mouse button to the axis produces translation motion. Dragging perpendicular to the axis stroke produces rotation about the axis. Single axis rotation and translation can be active at the same time since they both use the same gesture stroke. To translate in 2D, two perpendicular lines are stroked on an object to determine the plane of translation.

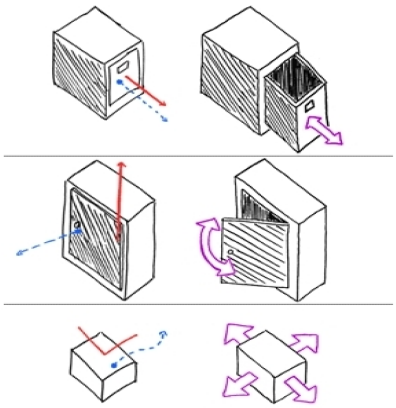

Fig. 6. Axis is stroked in red and motion in blue [2].

F. Grouping and Copying

Objects are grouped with the object on the surface of which they are created. Every object has a list of its grouped objects. Any transformation is applied to the group. So, if a box is sketched on top of a table, the box will move whenever the table does. Relationships are only one way. If a table is grouped to the floor the opposite is not implied. Moving the table woyld not move the floor, but moving the floor would move the table.

SKETCH also allows the user to explicitly define groups by drawing a lasso (any closed curve) around a group of objects. All objects whose geometric centre lies within the lasso are considered to be one group. Lassoing with the Shift-key copies all grouped objects and dragging the lasso scales the objects (Figure 7).

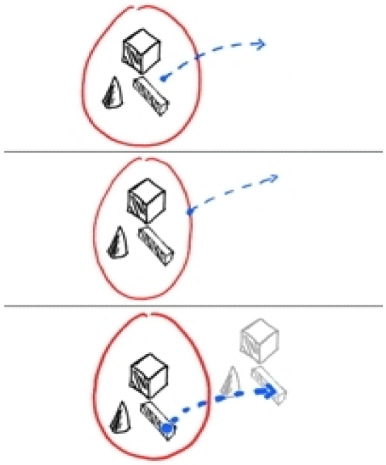

Fig. 7. A closed curve groups objects. A closed curve followed by click and drag scales. A closed curve followed by click and drag with Shift-key copies objects [2].

V. Conclusion

SKETCH made a huge impact on using gesture recognition using a simple interface for 3D modelling. Gesture recognition is a difficult problem [9]. A good gesture-based interface requires managing the tradeoffs between gestures that are natural and those that are effective [2]. SKETCH laid the foundation of 2D sketched gestures for shape construction and rapid creation of approximate shapes. SKETCH showed how a gesture-based modeller could be used to simplify conventional 3D modelling. The application was designed to help a user with geometric modelling using intuitive gestures to simplify the process. For example, if a user draws three segments meeting in one point the system creates a cube whose dimensions are determined by the segment lengths (Figure 1). Once a user has mastered a set of intuitive gestures like this, it is easy to create complex models consisting of many primitives [8]. The simple constraint system used by SKETCH kept the user's learning overhead small and made the application robust.

Teddy is an application the extended on SKETCH's gesture recognition 3D modelling. Teddy used SKETCH's main ideas to create a sketching interface for designing 3D freeform objects. The user draws 2D freeform strokes interactively specifying the silhouette of an object, and the system automatically constructs a 3D polygonal surface model based on the strokes [10]. Teddy like SKETCH, maintained a simple user interface policy without menu options.

SKETCH used non-photorealistic rendering to make comprehensible models in the initial conceptualize stage of a design

process. SKETCH made way for further research in gesture recognition for 3D modelling like the work of Sheng, et al [11] and Song, et al [12]. The SKETCH videotape shows how easy it is to come up with a 3D 'sketch' model. It was designed in a way to make it usable by anyone without needing specific knowledge of both the application (due to its intuitive interface) as well as of the object as SKETCH was designed to be used in the prototype phase. SKETCH laid the basis of application design that focused on a non menu-based user interface design and was able to come up with an application that tested well with users [2] and was robust in design and use. While the learning curve is initially steeper than with a WIMP interface (an interface that relies on icons and menus), the practised user experiences reduced cognitive effort [13].

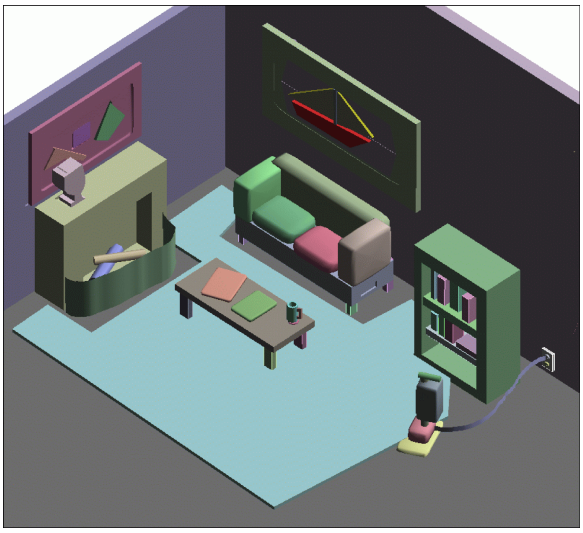

Fig. 8. An image drawn using SKETCH [14].

REFERENCES

[1] Fan, Zhe, et al. "A sketch-based interface for collaborative design." SBM. 2004.

[2] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. "SKETCH: An interface for sketching 3D scenes." ACM SIGGRAPH. 1996.

[3] Artifice, Inc. Design Workshop. Macintosh application.

[4] W. Wang and G. Grinstein. "A survey of 3D solid reconstruction from 2D projection line drawings." Computer Graphics Forum. 12(2):137158. June 1993.

[5] D. Pugh. "Designing solid objects using interactive sketch interpretation." 1992 Symposium on Interactive 3D Graphics. 1992.

[6] D. Rubine. "Specifying gestures by example". SIGGRAPH. July 1991.

[7] R. Bukowski and C. Sequin. "Object associations: A sim- ple and practical approach to virtual 3D manipulation." 1995 Symposium on Interactive 3D Graphics. 1995.

[8] Karpenko, Olga, John F. Hughes, and Ramesh Raskar. "Freeform sketching with variational implicit surfaces." Computer Graphics Forum. Vol. 21. No. 3. Blackwell Publishing, Inc, 2002.

[9] Cook, Matthew Thomas. *A 3-dimensional modeling system inspired by the cognitive process of sketching.* ProQuest, 2007.

[10] Igarashi, Takeo, Satoshi Matsuoka, and Hidehiko Tanaka. "Teddy: A sketching interface for 3D freeform design." ACM SIGGRAPH. 2007.

[11] Sheng, Jia, Ravin Balakrishnan, and Karan Singh. "An interface for virtual 3D sculpting via physical proxy." GRAPHITE. Vol. 6. 2006.

[12] Han, Song, and Grard Medioni. "3DSketch: modeling by digitizing with a smart 3D pen." Proceedings of the fifth ACM international conference on Multimedia. ACM, 1997.

[13] Van Dam, Andries. "Post-WIMP user interfaces." Communications of the ACM 40.2. 1997.

[14] Bob Zeleznik, Andy Forsberg, Loring Holden. Brown University Computer Graphics Group. http://graphics.cs.brown.edu/research/sketch/