Building Worlds with Strokes

David Cochard* Pierre Rahier† Sébastien Saigo‡ Mageri Filali Maltouf§

Silicon Studio Corp.

Abstract

The legendary Ub Iwerks designed the Mickey Mouse character out of circles to make it simple to animate while achieving both expressiveness and dynamism. Our entry for the World Builder 3DUI contest revolves around the same idea.

The user is allowed freeform modeling and natural selection through a device placed on his finger. The other hand holds a ball used to move, place and paint objects. Exploring the world is attained through a combination of both interaction techniques. Those movements are subject to no constraint, which allows for intuitive editing operations.

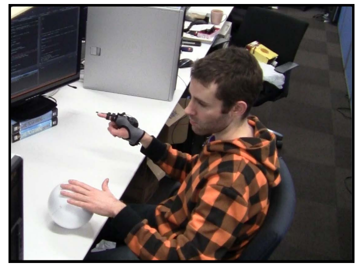

Figure 1: Using the ball and the glove

1 Introduction

Traditional modeling applications require mastering well-defined but complex geometric tools. Therefore, imagination is not the sole limit of these systems.

Our idea is to combine a flavor of the Teddy freeform modeling described in [2] with standard computer vision and machine learning algorithms. This allows to strike the balance between expressiveness and ease-of-use that is required for non-expert design tasks and rapid prototyping. We provide inexpensive hardware tools that match our virtual controllers, in the form of a ball and a glove.

Our ideas include:

- a tangible interaction technique for computer-aided design that relies on widely available hardware and which can be applied in a variety of other contexts
- a direct and natural mapping between our device and the possible actions in the virtual world
- a suggestion engine for shape recognition, to strike the balance between freedom and expressiveness

2 Hardware

We use a ball and an optical emitter/sensor and combine them into a new interaction metaphor. For a right-handed user, the ball is held by the left hand and the glove is worn on the right hand (Figure 1).

The system can be used with a traditional mouse, which can facilitate its adoption and its integration in industrial pipelines. This comes at the price of less intuitive and natural movements.

2.1 The Ball

We use a hollow polyester ball in which we place a smartphone held by two notches. The phone is running a custom application that broadcasts tri-axis gyroscope and accelerometer values measured by the sensor on the network. We use this information to keep track of the orientation of the phone in a robust manner. The virtual proxy for the ball is a hemispherical menu that displays the range of possible actions.

To track the orientation of the phone, we make use of the IMU filter developed by Sebastian O. H. Madgwick and described in [3]. This filter is computationally inexpensive, effective at low sampling rates and easy to tune using a limited set of parameters.

2.2 The Glove

We use a glove on which we attached an IR LED and two buttons, called Action and Menu, which correspond to the left and right click of the mouse. We keep track of the position in 3D of the IR LED with two Nintendo Wiimote controllers used as infrared cameras. Thus, instead of working on a plane like a traditional mouse, our system works in space. The virtual proxy is a 3D cursor that replicates its translational motion.

To accurately track the position of the IR LED, we adjust the positions given by the two Wiimotes using a least square minimization on the epipolar constraint, as described in [1].

3 Usage

We arranged the list of possible actions to reach balance between the ball and the glove.

Modeling. The user models objects simply by sketching strokes with his finger.

Selection. The user selects shapes by moving the 3D cursor on the object of interest and clicking on the Action button.

Placement. Placement of objects in the virtual world relies on translation from the glove and rotation from the ball. Therefore it is possible to rotate an object while translating it.

Scaling. Scaling of objects in the virtual world relies on the rotation of the ball along a given axis. The ball is used as a knob to shrink or stretch objects.

Coloring. Coloring of objects is achieved by moving the ball, which rotates the hemispherical menu texturized as a color picker.

Texturing. Texturing an object is achieved by using the ball to select texture patterns pinned to the hemispherical menu. Our implementation offers only a limited set of texture patterns but future work includes multi-level pattern selection, as well as procedural textures whose parameters can also be tweaked using the ball interaction metaphore.

Space exploration. Similarly to object placement, we retrieve relative rotation from the ball and relative translation from the glove to move the camera in space.

*e-mail:david@siliconstudio.co.jp
†e-mail:pierre@siliconstudio.co.jp
‡e-mail:se-saigo@siliconstudio.co.jp
§e-mail:mageri.filali@siliconstudio.co.jp

4 SOFTWARE

Our method relies on three software components: the freeform modeling tool, the shape suggestion system and the hemispherical menu.

4.1 Freeform modeling

A stroke drawn with the finger can be seen as the object silhouette, or a cut path. User strokes are automatically processed by shape builders. Each builder applies a specific algorithm to the stroke in order to create a 3D model from it. The generated shape is then placed in the workspace to directly match the silhouette location and orientation.

4.1.1 Extruder shape builder

This is a simple extruder that builds a flat surface from the stroke, duplicates it and connects the two (Figure 2 Left). This builder is very convenient to create architecural models.

4.1.2 Teddy shape builder

This builder is our implementation of the techniques presented in [2]. The input curve is resampled to erase redundant information in low curvature portions. We apply a Delaunay triangulation to the resulting point cloud. A clever classification of the triangles allows to construct a skeleton which is then used to blow up the shape like a balloon (Figure 2 Right).

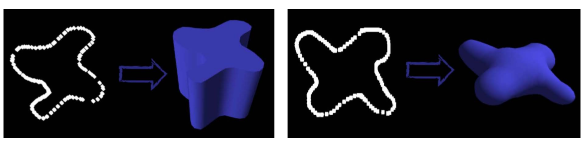

Figure 2: Extruder and Teddy shape builders

4.1.3 Cutting tool

Open strokes are used to modify shapes after their creation. The user simply has to draw a stroke in space and the selected mesh will be cut in two new shapes.

4.2 Shape suggestions

The Teddy shape builder tends to produce cushion-like models with no sharp edges. As a consequence, it is difficult to draw shapes with triangular or square bases without using the cut tool. On top of this, our interaction method is very sensitive to motion noise, which sometimes remains untouched by our resampling algorithms and can be seen in the output stroke. Thus, it is very difficult to draw perfect primitives such as spheres or squares.

To balance for this, we use a machine learning algorithm. Offline, we train an artificial neural network to recognize and classify primitives from input strokes. The system makes suggestions in real-time, depending on the input of the user. The suggestions include primitives as well as the output of the shape builders (Figure 3 Left).

4.3 Hemispherical menu

We provide the user with a hemispherical menu whose surface changes depending on the task being performed or the available choices.

The menu is hemispherical and matches the ball motion to keep our interaction metaphore consistent. It offers intuitive and rapide exploration of the different options available while keeping the left hand on the ball, and without moving the glove.

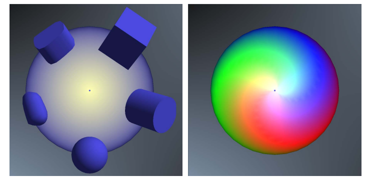

Figure 3: Menu suggestions (left) and color picker (right)

5 USER STUDY

We conducted our user study on 3 people, including one artist, using the *Mickey Mouse at the Crossroads of the World* statue. We measured ease-of-use of our system and esthetic results.

The use of the ball and the glove quickly became a natural way to interact with the system for all users. Everybody quickly understood the design-by-stroke interaction metaphor but it was still hard for inexperienced users to see how to create complex models made of many compound shapes.

Given a limited amount of time, the artist was able to produce a quite faithful reproduction of the statue (Figure 4). The laymen spent most of the given time grasping the modeling process, already familiar to artists. Therefore they could not quite achieve the same result.

While moving shapes in the workspace was intuitive for all users, it turned out that our orbital camera was hard to control with the glove and the ball. Another recurent remark concerned the limitation of our texture palette, specifically for the glossy sphere on which Mickey Mouse is standing.

Figure 4: Comparison of the original statue and our model

6 CONCLUSION

6.1 Limitations

The size of the 3D workspace is directly related to the position and the angle of the Wiimotes. Simply put, it consists of the intersection of the two Wiimote camera frustums since the IR LED must be seen by both cameras to be tracked. Working in a wide workspace is tiring for the user who must move his arm a lot to draw strokes. Conversely, a user working in a narrow workspace might experience recurrent loss of tracking.

Another obvious drawback of our system in a production environment is the lack of text input.

REFERENCES

[1] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, Second Edition*. Cambridge University Press, Mar. 2004.

[2] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3d freeform design. In *ACM SIGGRAPH'99*, pages 409–416. ACM, 1999.

[3] S. O. H. Madgwick, A. Harrison, and R. Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *International Conference on Rehabilitation Robotics (ICORR)*, pages 1–7. IEEE, 2011.