論文 / 著書情報
Article / Book Information

Title	Hand-drawn Animation with Self-shaped Canvas
Authors	Masaki Fujita, Suguru Saito
Citation	Proceedings of SIGGRAPH ’17 Posters, No. 5
Pub. date	2017, 7
Note	(c) ACM 2017. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in Proceedings of SIGGRAPH ’17 Posters, , http://dx.doi.org/10.1145/3102163.3102212.

Hand-drawn Animation with Self-shaped Canvas

Masaki Fujita
Tokyo Institute of Technology
masaki@img.cs.titech.ac.jp

Suguru Saito
Tokyo Institute of Technology
suguru@c.titech.ac.jp

Figure 1: Inbetweening result. Left: two keyframes with red most and blue most colors and inbetween images with other colors. Right: a hidden self-shaped canvas with a shape derived using the aforementioned keyframes and its motion for the inbetweening. The lines of the keyframes are stored on the canvas as double lines.

ABSTRACT

Although 3D CG tools produce similar style animations with conventional 2D animation, conventional hand-drawn animation has advantages that cannot be substituted in animation produced by those tools.

This paper introduces a new method to assist animators in creating 2D keyframe animation, which utilizes the *self-shaped canvas* hidden from the user. Drawing keyframes is the main input. Because the canvas has a three-dimensional shape deformed according to keyframes, inbetween animation with them mapped on it obtains depth-aware motion.

CCS CONCEPTS

• **Computing methodologies** → *Non-photorealistic rendering*; *Parametric curve and surface models*;

KEYWORDS

animation, drawing, inbetweening, non-photorealistic rendering

ACM Reference format:
Masaki Fujita and Suguru Saito. 2017. Hand-drawn Animation with Self-shaped Canvas. In *Proceedings of SIGGRAPH '17 Posters, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: 10.1145/3102163.3102212

1 INTRODUCTION

3D CG techniques can produce similar animation with 2D conventional cartoon-like animation. However, they are not completely the same. Hand-drawn animation has unique characteristics in its style, having appearance-oriented shape deformation. Using 3D CG to emulate it is difficult. The advantage of such hand drawings is given by the fact that an animator can directly draw preferred shapes on the screen, not indirect indications via 3D object deformations and those projections.

However, full hand drawing is very time-consuming work. Thus, several methods were proposed to replace a part of the work, inbetweening, with a computer[Baxter and Anjyo 2006; Dalstein et al. 2015; Whited et al. 2010; Xing et al. 2015]. However, they do not account for the depth of the object shape. Therefore, drawing keyframes to make an animation that needs to account for out-of-plane rotation requires a highly skilled animator. In contrast, our method produces depth accounting for inbetween images because inbetweening is performed in 3D with the *self-shaped canvas*, on which the input strokes are mapped. The interesting points of the canvas are that it is hidden from a user and that its shape is automatically deformed appropriately according to user input strokes. Therefore, no 3D CAD like operation is required. Drawing keyframes, their orientation directions, and a global motion line are only required as input.

2 METHOD

2.1 User Interface

The user inputs two keyframes on the central area of the window shown in Figure2. The spherical gray area with cross lines is the guide to draw an object, which is a face in this paper. The position and orientation of the gray area are decided by the user for each keyframe before drawing. They mean a pose of the *self-shaped canvas* for a keyframe behind the guide. The shape of the guide is not related to the shape of the canvas. In addition, a global motion line used in inbetweening is also specified by the user. Moving the slider in the bottom of the window, smooth inbetweened images can be shown anytime.

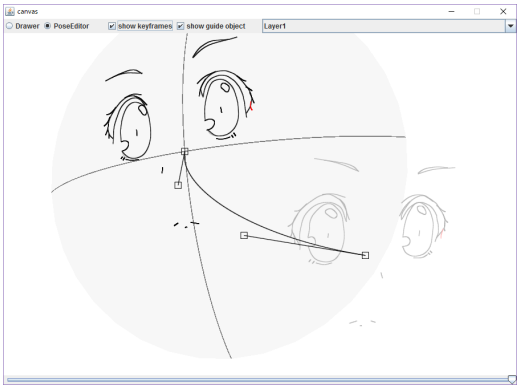

Figure 2: User interface

2.2 Self-Shaped Canvas

The *self-shaped canvas* is a three-dimensional surface, with a geometry defined as a parametric surface $z = f_{\boldsymbol{p}}(x, y)$ where $\boldsymbol{p}$ is the parameter vector. First, points that should be on the surface are calculated to determine $\boldsymbol{p}$.

Two keyframe images on a screen I_0, I_1 and the corresponding view positions are transformed to the canvas coordinate system according to the input canvas poses P_0, P_1. They are denoted by I'_0, I'_1 and v_0, v_1 respectively. From the pair of corresponding points c_{0i} on I'_0 and c_{1i} on I'_1, two points d_{0i} and d_{1i} are derived as the closest points to the opposing projection lines shown in Figure3. Ideally, they are in the exact same position. Considering target object's deformation, which may occur between keyframes, we deal with these two points as candidate points to be placed on the surface.

The parameter vector $\boldsymbol{p}$ is obtained by minimizing $e(\boldsymbol{p})$ shown as follows as the sum of square differences in the depth value of $\boldsymbol{d}_{ni}$.

$$e(\boldsymbol{p}) = \sum_{n=0,1} \sum_{i} |z_{ni} - f_{\boldsymbol{p}}(x_{ni}, y_{ni})|^2, \qquad (1)$$

where $\boldsymbol{d}_{ni} = (x_{ni}, y_{ni}, z_{ni})$.

When an input keyframe is updated, $\boldsymbol{p}$ is immediately recalculated. The right of Figure1 shows the derived canvas shape from input keyframes.

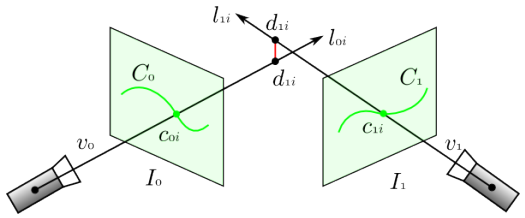

Figure 3: Calculation of points to be placed on the self-shaped canvas

2.3 Line Storing and Rendering

Lines on keyframes are projected and stored on the *self-shaped canvas*. The lines are usually stored like double lines, as shown on the right of Figure1, for several reasons. These include the gap of geometrical accurate projection and drawings, target object deformation occurring between keyframes, and local adaptation mismatch in the shape decision. When rendering an inbetween image at t $(0 \leq t \leq 1)$, the canvas is moved along with the global motion line at t and rotated using linear interpolated rotation parameters with t. The final lines are also interpolated between double lines with t on the canvas.

The left of Figure1 shows two keyframes(the red most and blue most lines) and inbetween images(other colors). The trajectories and the speed of parts are varied by considering the depth of the target shape. Please watch the supplemental video.

3 CONCLUSION AND FUTURE WORK

We presented a new method for inbetweening hand-drawn keyframes. The main contribution was to introduce the *self-shaped canvas*. It can produce depth-aware animation. Using this method, an artist can create animation in the conventional drawing way on a computer.

A multi-layer canvas will allow drawing more complex objects. Allowing three or more keyframes while considering continuity will be useful for long sequence animations. These extensions are future work. Contour lines might be better to stored in a different way from the one using the *self-shaped canvas*. This is the current issue in developing the method.

REFERENCES

William Baxter and Ken-ichi Anjyo. 2006. Latent Doodle Space. *Computer Graphics Forum* 25, 3 (2006), 477–485. DOI:https://doi.org/10.1111/j.1467-8659.2006.00967.x

Boris Dalstein, Rémi Ronfard, and Michiel van de Panne. 2015. Vector Graphics Animation with Time-varying Topology. *ACM Trans. Graph.* 34, 4 (July 2015), 145:1–145:12.

Brian Whited, Gioacchino Noris, Maryann Simmons, Robert W. Sumner, Markus Gross, and Jarek Rossignac. 2010. BetweenIT: An Interactive Tool for Tight Inbetweening. *Computer Graphics Forum* 29, 2 (2010).

Jun Xing, Li-Yi Wei, Takaaki Shiratori, and Koji Yatani. 2015. Autocomplete Hand-drawn Animations. *ACM Trans. Graph.* 34, 6 (Oct. 2015), 169:1–169:11.