Tweakable Shadows for Cartoon Animation

Hidehito Nakajima
Waseda University
3-4-1 Okubo, Shinjuku-ku,
Tokyo Japan, 169-8555
nakajima@mlab.phys.waseda.ac.jp

Eiji Sugisaki
Waseda University
3-4-1 Okubo, Shinjuku-ku,
Tokyo Japan, 169-8555
eijil11@toki.waseda.jp

Shigeo Morishima
Waseda University
3-4-1 Okubo, Shinjuku-ku,
Tokyo Japan, 169-8555
shigeo@waseda.jp

ABSTRACT

The role of shadow is important in cartoon animation. Shadows in hand-drawn animation reflect the expression of the animators' style, rather than mere physical phenomena. However shadows in 3DCG cannot express such an animators' touch. In this paper we propose a novel method for editing the shadow with both advantages of hand drawn animation and 3DCG technology. In particular, we discuss two phases that enable animators to transform and deform the shadow tweakably. The first phase is that a shadow projection matrix is applied to deform the shape of the shadow. The second one is that we manipulate vectors to transform the shadow such as scaling and translation. These vectors are used in shadow volume method. The shadows are edited directably by integration of these two phases. Our approach enables animators to edit the shadow by simple mouse operations. As a result, the animators can not only produce shadows automatically but also reflect their style easily. Once the shape and location of shadow are decided by animators' style in our method, 3DCG techniques can produce consistent shadow in object motion interactively.

Keywords

Cartoon animation, Shadows, Non-photorealistic rendering, Cel animation, interactive techniques

1. INTRODUCTION

Since the natural shape and animation of shadow are tedious to draw by hand, shadowing in traditional cel animation requires a lot of manual labor. However, shadows play an important role in cel animation as well as various other animations. The shadow is not so outstanding as to attract viewer's attention such as facial animations or human motions, but sometimes the expression of shadows gives a specific meaning to a scene. Especially, in hand-drawn cel animations, a shadow is drawn according to animators' style, which may not be physically correct but looks natural and fantastic. For instance, when animators intend audience to focus on a character's face, animators may edit the shadow covering character's face, may draw simple shadows or may not express the shadow. Likewise, animators who want to express impressive scene draw shadows as if the shadow itself were a lively character or a terrible monster.

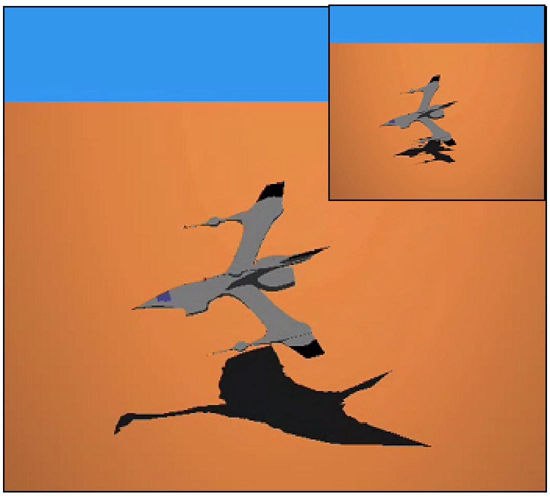
Figure 1. An example of impressive shadow by our tool. The shadow becomes a different shape from the original object. (in this example, the shadow is like a bird) Inset: the original shadow before editing the shadow.

However, it is difficult for the animator to express their styles interactively by using 3DCG technology. Since the shadow in 3DCG is produced automatically by calculating physics based illumination, the procedure for creating 3DCG shadows has a constraint that depends on the relationship between the light source and a 3D model, and this is why shadows in 3DCG have been considered difficult to edit their shapes, scales, and positions. When animators want to change or control shadows in 3DCG, they only can correct the position of light source, modify 3D models indirectly so far. These tedious operations might disturb animators' creativeness.

At the same time, there are many advantages of using 3DCG methods. For instance, we can check an animation result from any kinds of views for animatics. Using a 3D model enables animators to redraw and recreate scene easily, and all processes except setting environmental or modeling parameters are automatically performed. These advantages reduce animators' manual labor. Nowadays, in the cartoon animation industry, the technique of combining 3DCG method and 2D hand-drawn animation has been introduced [Kri06a].

Therefore, we propose the new world that leads to animators to bring out creative mind for editing shadow by combining animators' style and 3DCG technique. Figure 1 shows impressive shadow created by our proposed tool.

Our method is divided into two phases, i.e., deformation and transformation. First, an applied shadow projection matrix is used to deform the shape of the shadow. Second, manipulating vectors of the light obtained by shadow volume method enables transformation of the shadow. By using our method, shadows are considered as the 3D model that animators can transform and deform. These two methods enable animators to hybrid with both animators' style and 3DCG technology. By combining both advantages, animators can edit shadows in 3DCG to apply to their style, and add shadows even into the cel animation scene. Additionally, our tool can apply to key-frame animations. Therefore animators can edit not only shadows in an image but also in key-frame animation. Our method improves working efficiency and helps to express animators' style by using 3DCG technique.

2. RELATED WORK

In this section, we discuss the overall of the computer graphics techniques in cartoon animation. Moreover we mention related work of light and shadow design, and interactive rendering in non-photorealistic field.

[Las87a] was likely the first researcher to discuss the principles of traditional 2D hand-drawn animation. He mentioned these principles applied 3D computer animation. In his paper, he described how to make the audience entertain and presented this method to animators.

Recently, 3D computer graphics techniques have been used in the cartoon animation industry (e.g. Ghost in the Shell). The movie "Appleseed" was a landmark animation featuring hyper realistic imagery and a hybrid 2D and 3D style [Kri06a].

[Cor98a] was a method for applying complex textures on 3D objects to hand-drawn animation. [Rad98a] presented a technique called "View-Dependent Geometry". In conventional cel animation, characters or objects often have all kinds of view-dependent distortions that cannot be described with 3D models. Thus they capture the view-dependent distortions as the reference drawing. [Sug06a] proposed cartoon-like hair motion creating technique. Motion cloning from cartoon hair animation sequences enables animators to create cartoon hair animation easily. [Jue06a] presented the "Cartoon Animation Filter". This filter converts an arbitrary input motion signal to output motion that is more "alive" or "animated". [Chr06a] provided the re-using method for traditional animations. First, they divide cartoon images from their backgrounds. Then, they provide inbetween contours and textures.

As for light and shadow techniques, [Bar97a] described various lighting and shadowing techniques for computer cinematography. They described that the purposes of lighting for cinematography are to contribute to the storytelling, mood, and image composition, and to direct the viewer's eye. These factors are important for cartoon animation same as cinematography. [Pet00a] proposed to inflate a 3D figure based on 2D hand-drawn images to produce shadows semi-automatically for cel animation. Our method focuses on flexibility compared to their method. [Pel02a] proposed interactive interface for transforming the shadow directly on shadowed region. However, this method did not solve the constraint that shadows linked to relationship between lights and objects. In our paper, we try to solve this problem. [Lak00a] presented several real-time methods to emulate cartoon styles, but they did not describe the shadow technique. [Anj03a][Anj06a] proposed intuitive highlighting technique. This method enables animators to edit stylized highlight directly on a 3D object surface. This concept is applied to shadows in our method.

3. SHADOW EDITING PROCESS

In this paper, we introduce two phases for editing shadows. In the first phase, we create an object that represents the shape of a shadow by using shadow projection matrix. We call this object shadow object. We set the shadow object precisely in the same

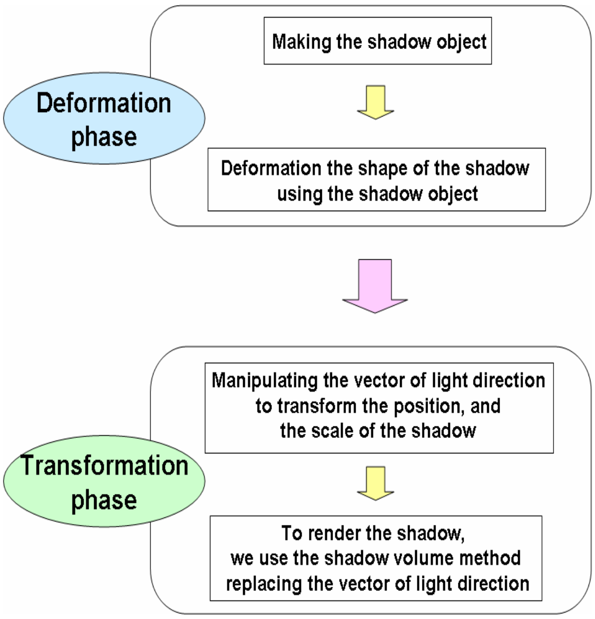
Deformation phase
Making the shadow object
Deformation the shape of the shadow using the shadow object
Transformation phase
Manipulating the vector of light direction to transform the position, and the scale of the shadow
To render the shadow, we use the shadow volume method replacing the vector of light direction

Figure 2.Process Overview

position where the 3D model is. By manipulating the vertex of shadow object directly, we can deform the shape of the shadow. This is called the deformation phase. In the second phase, a shadow is rendered using the shadow volume method. At that time, shadow volume is created by the shadow object instead of the 3D model. Then, animators edit the vector from the light to the 3D model to transform the shadow, such as translation and scaling. This is called transformation phase. An over view of the process is shown in Figure 2.

In section 4, we describe the deformation phase that includes making a shadow object and a deformation technique. In section 5, we describe the transformation phase that includes how to render the shadow and transformation technique.

4. DEFORMATION PHASE

4.1 Making a shadow object

This sub-section explains how to create the shadow object that contributes to the deformation. This object enables animators to edit the shape of the shadow directly. We produce the shadow object by using the shadow projection matrix. The matrix is denoted as $\mathbf{M}$ in the following Equation 1. We calculate $\mathbf{M}$ by solving a ground equation and an equation for a straight line in the direction of a light source. In this case, the ground is the 2-dimension ground that 3D model casts shadow and the line indicates direction of light. The shape of shadow onto the two-dimension ground surface is created using the matrix. The shadow object vertex multiplied by the matrix gives new vertex of the shadow shape projected onto the ground. We set the projected vertex to the position of the 3D model and adjust it to the same scale as the 3D model.

$$\mathbf{M} = \begin{bmatrix} \mathbf{n}\cdot\mathbf{L} - n_x L_x & -n_x L_y & -n_x L_z & 0 \\ -n_y L_x & \mathbf{n}\cdot\mathbf{L} - n_y L_y & -n_y L_z & 0 \\ -n_z L_x & -n_z L_y & \mathbf{n}\cdot\mathbf{L} - n_z L_z & 0 \\ (\mathbf{n}\cdot\mathbf{p})L_x & (\mathbf{n}\cdot\mathbf{p})L_y & (\mathbf{n}\cdot\mathbf{p})L_z & \mathbf{n}\cdot\mathbf{L} \end{bmatrix}, (1)$$

Where, $\mathbf{n} = (n_x, n_y, n_z)$ is the ground's normal vector, $\mathbf{p}$ is a point on the ground, and $\mathbf{L} = (L_x, L_y, L_z)$ is the direction of the light.

4.2 Shadow deformation

This sub-section explains how to control the deformation. We edit a shape of the shadow by using a mouse operation for manipulating each vertex of the shadow object directly in *the editing window*. Note that *the editing window* means the black window on the upper left in figure 3. We call this window *the editing window.* First, animators click on the part of the shadow object where animators want to edit the shape. This region $\Omega[\varepsilon]$ is specified by animators, according to Equation 2. Then, animators drag the part along the direction that they want to

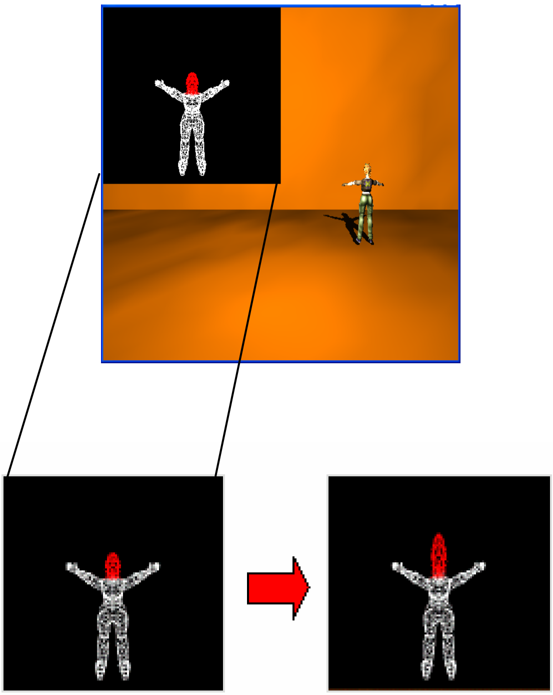

Figure 3. A deformation in *the editing window.* Left image is an original shape of the shadow object. Right image is a shape of shadow object after editing.

stretch or compress. In Equation 2, (V_x, V_y) denotes the vertex of a shadow object, and ε is a threshold that decides the region to move. Animators can move vertex (L_x, L_z) to vertex (L'_x, L'_z) of a new shadow object in the region defined by Equation 3 and Equation 4, with (M_x, M_z) being the amount of movement in proportion to a drag of the mouse operation.

$$\Omega[\varepsilon] := \{(Vx, Vy) \mid (Cx - Vx)^2 + (Cy - Vy)^2 < \varepsilon\}, \qquad (2)$$

$$L'x = Lx + Mx, \qquad (3)$$

$$L'z = Lz + Mz, \qquad (4)$$

In addition, we provide the fix mode as an additional mode of deformation phase that gives animators flexibility of editing. In this mode, animators specify the parts that should not move during deformation. Animators set a region by using Equation 2 to fix a vertex with the fix mode. Animators can deform a shadow object outside of the region specified in the fix mode. This mode enables animators to create a shadow object that cannot be achieved by the usual deformation.

Repeating the operation, animators can edit the shape of the shadow that they want to express. Figure 3 shows a deformation example in *the editing window*.

4.3 Animation

Animators can make the animation such as key-frame animation using deformed shadow by our method. Saving the amount of M in Equation 3 and Equation 4 enable animators to create a key-frame animation. For instance, we consider the shadow of the human character here. The shadow of character's head can be stretched by animators' edit. After editing the shadow, if the 3D model moves freely, the shadow of character's head stretches along the direction that the animators drag. Figure 4 shows this example. These operations are performed *in the editing window*.

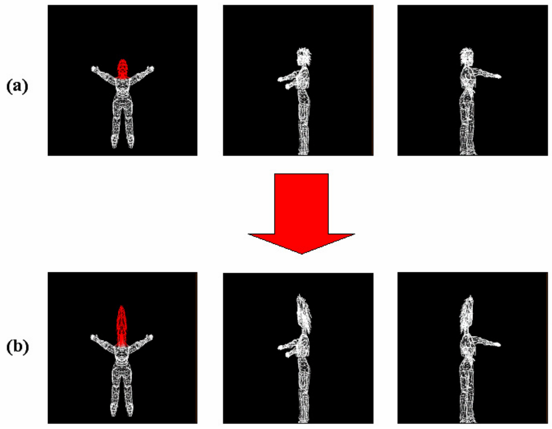

Figure 4. Simple animation example in *the editing window*(Deformation). We deform the shape in the left image, and rotate the 3D model in the middle and right images.

(a) Before editing.

(b) After editing.

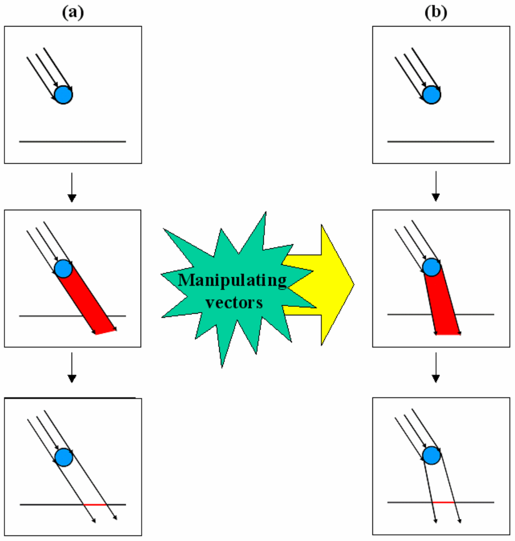

Figure 5. Overview of Transformation. In the top row images, the light illuminates the 3D model. In the middle row images, the shadow volume created using light vectors. In the bottom row images, the shadow on the ground is rendered.

(a) The standard shadow volume steps.

(b) The shadow volume steps after manipulating vectors.

5. TRANSFORMATION PHASE

5.1 How to render shadow

To render the shadow onto the ground, we use the standard shadow volume algorithm except replacing the light vector by a per-vertex vector. The shadow volume method is one of the most useful shadowing techniques in 3DCG. Recently, the shadow volume method is mentioned in a lot of research. For instance, [Ass03a] presented the soft shadow using shadow volume. However, soft shadow is not necessarily required for traditional cartoon animation. Therefore, we apply [Eve02a] technique created the umbra wedged shadow, not the penumbra wedged shadow. To define a shadow volume is the space where the light does not illuminate and it is generated by extending vectors from the light source to each outline vertex of the 3D model. In our method, a shadow is transformed by modifying these vectors. Then, the shadow on the ground is produced by the shadow volume using these vectors and the stencil buffer technique [Eve02a]. Figure 5 shows the overview of transformation.

5.2 Shadow Transformation: Translation and Scaling

To transform the shadow without changing material information, our method focuses on the extension vectors $\mathbf{L} = (Lx, Ly, Lz)$ and manipulates them. We provide the simple manipulating algorithm below. Equation 5 and Equation 6 represent the translation process. As for Ly, it is not used in this situation, because we edit x-z ground.

$$Lx' = Lx + \alpha, \tag{5}$$

$$Lz' = Lz + \beta, \tag{6}$$

In Equation 5 and Equation 6, α is the extent of the x-direction, β is that of the z-direction, and $\mathbf{L}' = (L'x, L'y, L'z)$ is obtained as a result of translation; α and β are inputted by a simple mouse operation.

Likewise, Equation 7 and Equation 8 represent the scaling process,

$$Lx' = Lx \times \gamma, \tag{7}$$

$$Lz' = Lz \times \gamma, \tag{8}$$

where γ is the extent of scaling decided by a mouse operation, and $\mathbf{L}' = (L'x, L'y, L'z)$ is also obtained as a result of scaling. In these ways, our method can easily achieve the desired transformations.

5.3 Animation

In the similar way of section 4.3, we can create the animation in transformation phase. We obtain the amount of transformation, α, β and γ in Equation 5, Equation 6, Equation 7 and Equation 8 when we transform the shadow object. To save these amounts enable animators to make a key-frame animation. For instance, a shadow is moved from original position to right position by the animators' editing. After editing the shadow, if the 3D model moves freely, position of the shadow moves right compared with the original position. Figure 6 shows a transformation example.

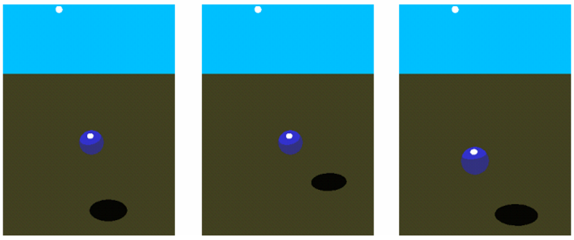

Figure 6. Simple animation example(Transformation). The left image is the original shadow. We transform the shadow in the middle image, and then translate the 3D model in the right image.

6. RESULT

In this section, we demonstrate the editing results created by our tool. The tool runs in real time on an Intel T2600 2.16-GHz platform with NVIDIA GeForce Go, and Directx9.0c as the graphics API.

Figure 7 shows the basic operations provided by our tool. The left image is an original shadow. Starting from original shadow in the most left image, the shadow is first deformed, then translated, and finally scaled. We can control the shadow simply by clicking and dragging with a standard three-button mouse. In usual, animators do not like tedious parameter setting with keyboard. Our method enables animators to create the exaggerating shadow easily by a simple mouse operation.

Figure 1 and Figure 8 demonstrate exaggerating shadow. We make the key-frame animation in the top row of Figure 8. Next, we transform the shadow to exaggerate. We then create the middle row of Figure 8 animation. The shadow in the middle left of Figure 8 is smaller and moves more backward compared to the top left of Figure 8. However, the shadow in the middle right of Figure 8 is bigger and more forward compared to the top right of Figure 8. As a result, the attention of the audience would focus on the shadow. To make more impressive animation, we then deform the shadow. Our method can create the bottom row of Figure 8 from the middle row of Figure 8. This result shows that our method enables animators to spread their expression.

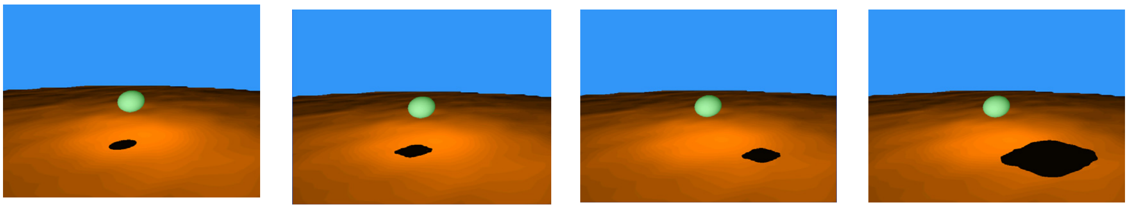

Figure 7. Basic Operation. The left image is the original shadow. Then, we deform, translate and scale the original shadow.

Finally, we demonstrate more practical situation using our method. In Figure 9, the shadow plays a dramatic role. The top row of Figure 9 is an original input, and gives the impression of mere shadow. On the other hand, in the bottom row of Figure 9, we exaggerate the shadow to make more dramatic scene in the animation. Traditionally, it is difficult to make the dramatic shadow such as this example, because the procedure for creating 3DCG shadows has a constraint that depends on the relationship between the light source and a 3D model. In that case, animators are required to adjust the light sources or the 3D model to change the shadow. These conventional approaches require animators to perform tedious trial and error in order to obtain an appropriate result. The bottom row of Figure 9 shows that other material information without shadows is same as information before editing the shadow. In this animation, we intended to emphasize a focus on "walking along the wall". Compared to the original animation, the effect is clearly proved by our result. So this also proves that our approach is useful for this kind of practical situation.

7. DISCUSSION AND FUTURE WORK

Unlike photorealistic shadows, the stylized shadows in cartoon animation should not be too detailed nor obeyed by physics-based reality. They should be simpler yet more expressive according to the scenario of the animation. As we mentioned in Section1, The role of shadows is not outstanding expressions in movies or animations such as facial animations or human motions. However, in cartoon animation, animators sometimes would like to use the shadows to foreshadow the successive phenomena such as transforming the 3D characters or objects. These kinds of expressions can be achieved by our approach.

In this paper, we have proposed a novel method of deforming and transforming shadow tweakably

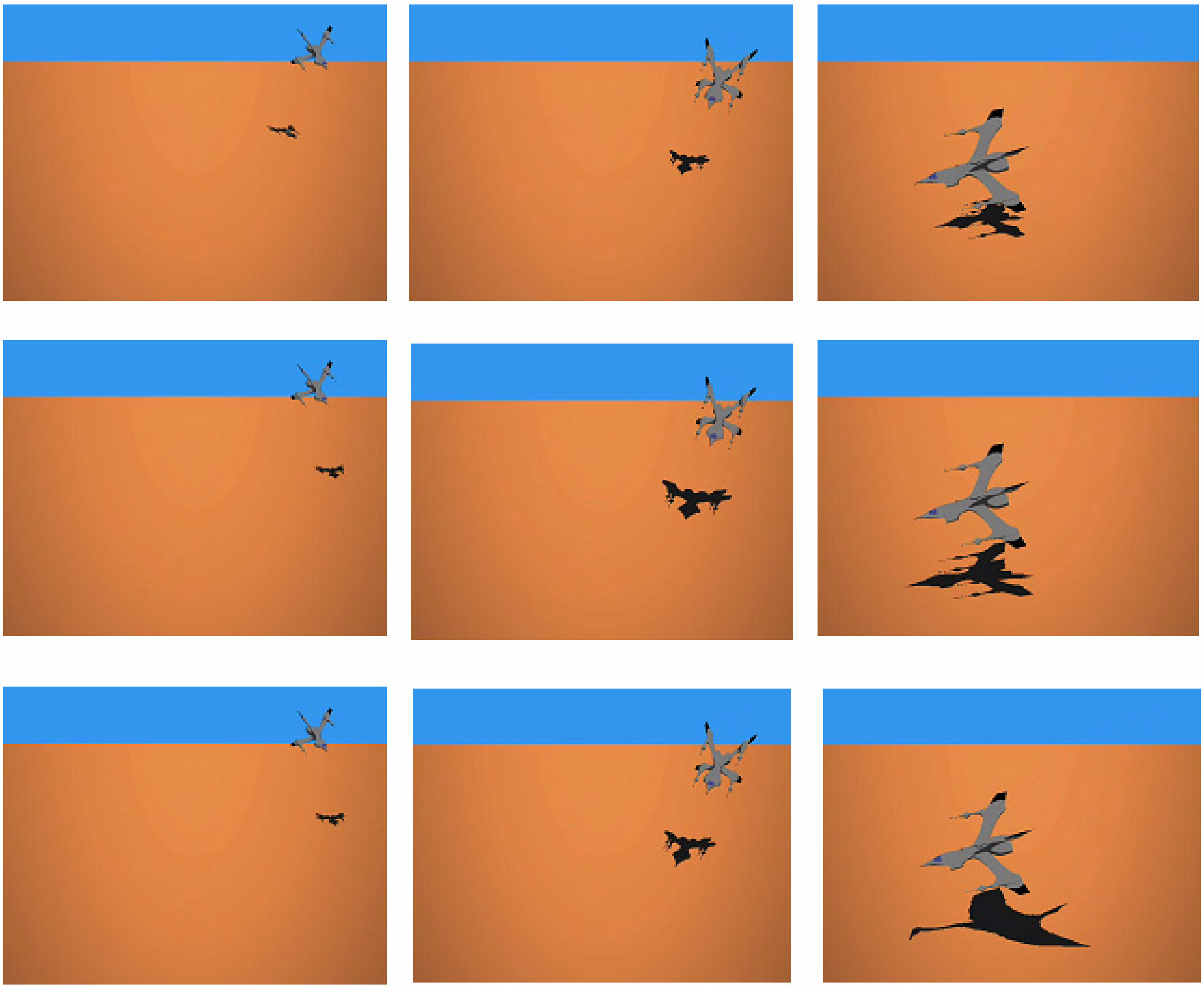

Figure 8. Exaggerating shadow example. The top row images are original key frame animation. The middle row images are transformation of top row animation. The bottom row images are deformation of middle row animation.

which allows animators to easily create such stylized shadows and their animations as desired in cartoon animation.

Moreover, our technique could maintain attributes of the edited shadow after transformation or deformation of the shadow object. This result shows that our approach enables animators to create animation intuitively and improves working efficiency.

Of course our final purpose of this research is to express 2D-like shadow by 3D models. Meanwhile our contribution is also expanding the way of expression of 3D shadows.

However, our method has not only several advantages, but also future work. When animators try to rotate the 3D object after deforming the shadow, the shape of shadow occasionally collapses. We improve our method to solve this problem in the future.

This tool is already directable one. However, since animators really need to and like to use an intuitive operation, we should improve our tool to be interactively. For example, animators must deform the shadow in *the editing window* so far. Therefore in order to deform shadows more intuitively, we should deform the shadow directly in the scene. In addition, when we handle easier shapes of shadows or a simple animation, we should develop a more simple method such as a template morphing. That is, given the original shadow and the designed shadow such as a circle, an ellipsoid, and a quadrangle, animators smoothly interpolate the original shadow into the designed shadow in animation. So we can perform animations more simply and quickly.

Treating of self-shadows should also be included in our future direction. As editing shadows, of course we consider how to edit the self-shadows. In addition, we should consider relationship between the shadow itself and its self-shadow.

8. ACKNOWLEDGEMENTS

We would like to thank Ken Anjyo and Aiko Nakatani for their comments and suggestions. This research is supported by Japan Science and Technology Agency, CREST project.

9. REFERENCES

[Kri06a] Krikke. J. Computer Graphics Advances the Art of Anime. IEEE Computer Society, May/ June, pp.14-19, 2006.

[Las87a] Lasseter. J. Principle of Traditional Animation Applied to 3D Computer Animation. Proceedings of SIGGRAPH, pp.35-44, 1987.

[Cor98a] Correa. T. Wagner, Jensen. J. Robert, Thayer. E. Craig, and Finkelstein. Adam. Texture Mapping for Cel Animation, Proceedings of SIGGRAPH, ACM Press, pp.435-446, 1998.

[Rad98a] Rademacher. P. View-Dependent Geometry. In Proceedings of SIGGRAPH, pp.439-

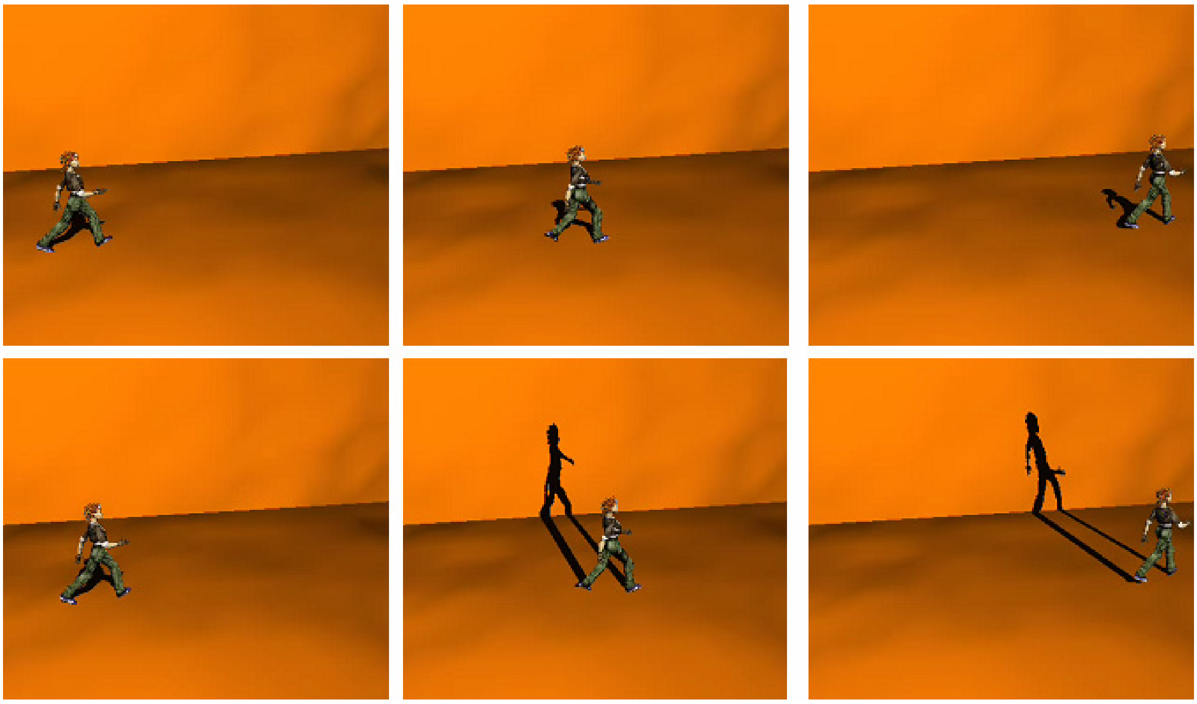

Figure 9. More practical shadow example. The top row images are original key frame animation. The bottom row images are editing of top row animation.

446, 1998.

[Sug06a] Sugisaki. E, Kazama. Y, Morishima. S, Tanaka. N, and Sato. A. Hair motion cloning from cartoon animation sequences. Computer Animation and Virtual Worlds July 2006, vol.17, pp.491-499, 2006.

[Jue06a] Jue Wang, Steven M. Drucker, Maneesh Agrawala, and Michael F. Cohen. The Cartoon Animation Filter. In Proceedings of SIGGRAPH, pp.1169-1173, 2006.

[Chr06a] Christina N. de Juan and Bobby Bodenheimer. Re-using Traditional Animation: Methods for Semi-Automatic Segmentation and Inbetweening. SIGGRAPH/Eurographics Symposium on Computer Animation, pp.223-232, 2006.

[Bar97a] Barzel. R. Lighting control for computer cinematography. Journal of Graphics Tools 2, pp.1-20, 1997.

[Pet00a] Petrovic. L, Fujito. B, Williams. L, and Finkelstein. A. Shadows for cel animation. In Proceedings of SIGGRAPH2000, pp.511-516, 2000.

[Pel02a] Pellacini. F, Tole. P, and Greenberg. D. A user interface for interactive cinematic shadow design. ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH2002 21, pp.563-566, 2002.

[Lak00a] Lake.A, Marshall.C, Harris. M, and Blackstein. M. Stylized rendering techniques for scalable real-time 3D animation. Proceedings of Non-Photorealistic Animation and Rendering 2000, pp.13-20,2000.

[Anj03a] Anjyo.K and Hiramatsu. K. Stylized highlights for cartoon rendering and animation. IEEE Computer Graphics and Applications 23, pp.54-61, 2003.

[Anj06a] Anjyo. K, Wemler. S, and Baxter. V. W. Tweakable Light and Shade for Cartoon Animation. In Proc. NPAR 2006, pp.133-139, 2006.

[Ass03a] Assarsson. U, Akenine-moller. T. A Geometry-Based Soft Shadow Volume Algorithm using Graphics Hardware. ACM Trans, pp.511-520, 2003.

[Eve02a] Everitt. C, and Kilgard. M. Practical and Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering. http://developer.nvidia.com, 2002.