# Instant and Intuitive Shadowing for CACAni System

*Eiji Sugisaki* *Feng Tian* *Hock Soon Seah*

School of Computer Engineering
Nanyang Technological University
Singapore
E-mail: SEIJI,ASFTian,ASHSSEAH@ntu.edu.sg

## ABSTRACT

We introduce an instant and intuitive shadowing technique for CACAni System. In traditional 2D Anime, since all frames are drawn by hand it takes long time to create an entire animation sequence. CACAni System reduces such a production cost by automatic inbetweening and coloring. In this paper, we develop a shadowing technique that enables CACAni users to easily create shadow for character or object in the image. The only inputs required are sequences of character or object layers drawn in CACAni System with alpha value. Shadows are automatically rendered on a virtual plane based on these inputs. They can then be edited by CACAni users. Achieving this, CACAni System is able to handle instant shadowing and intuitive editing in a short time.

**Keywords:** computer graphics, shadowing, shadow map, 2D animation

## 1. INTRODUCTION

CACAni (Computer Assisted Cel Animation) System [17] - [22] has been developed for traditional animation production industry especially in 2D Anime. The key functions for the system are automatic inbetweening and coloring. There are several layers that correspond to each component, such as the character or object layer, shadow layer, and background layer. All layers are composed for finalizing animation in Anime production. Likewise, CACAni System handles the layering. The time needed for the production can be saved with CACAni system's production pipeline. The system currently is being used for the Anime production on a trial basis. However, the shadowing for the drawn characters or objects in CACAni System has not been accomplished yet. In this paper, therefore, we present an approach on how CACAni System handles shadowing for the created animation sequence.


**Fig.1: Role of Shadow: Specifying the Character's Position**
*(i) Uncertain position, (ii) Standing, (iii) Floating;* in the left image, character's position is unclear without a shadow. However, other images enable audience to perceive the character's position clearly. *The character QooBeeAgapi, © 2004 Lu Peng*

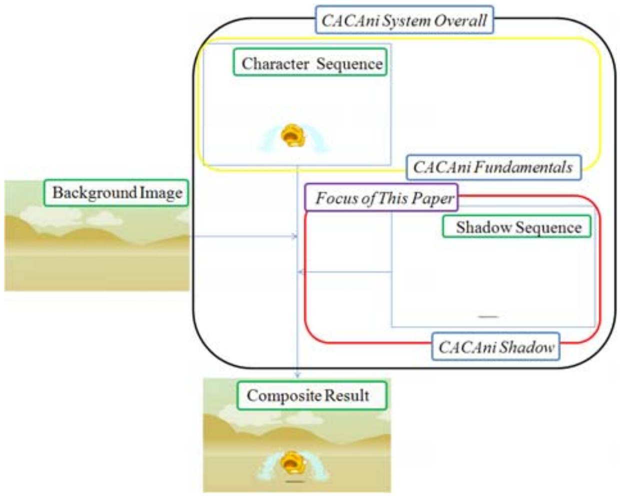


**Fig.2: Overview of Anime Production Workflow:** In Anime, Each layer is drawn individually and then composited. Currently, CACAni System can handle the character or object layer and shadow layer.
*The character QooBeeAgapi and background © 2004 Lu Peng*

In 2D Anime, shadows play an important role in the creation of symbolic visual effects related to the character's position and shape. In fact, audience is able to perceive whether a character is standing on the ground or not from the position of shadows (Figure 1). Unlike the detailed shadows of photo-realistic expression, shadows in Anime tend to be simplified because they are not required to be perfectly realistic

In Anime production, shadows are rendered not only to describe characters or objects in the scene in relation to a background image but also to get the attention of audience to particular characters and to exaggerate scenes. In spite of its usefulness, shadows sometimes are not drawn at all because of a lack of animators and time constraints in Anime production especially weekly Anime. Since drawing detailed shadows in each individual frame requires an amount of time, animators frequently do not have enough time to draw shadows according to their artistic intentions and preference [9].

Solving this problem is important for CACAni system because the system has been developed to reduce animators' workload. Figure 2 shows the overview of Anime production workflow. As we mentioned earlier, the system can handle the character or object sequence by auto inbetweening and coloring. The main focus of this paper is how to handle rendering and editing shadow based on character or object sequence in the system.

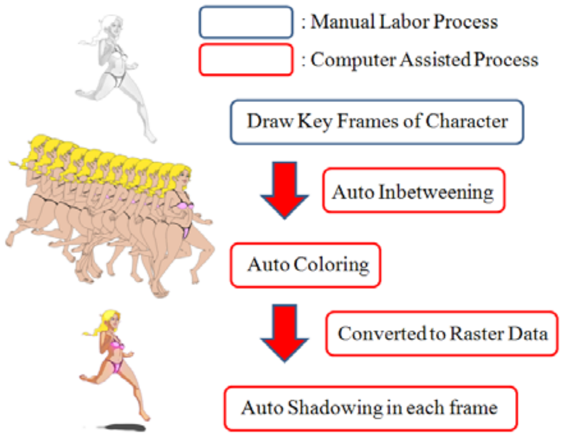


**Fig.3: Workflow in CACAni System:** First, CACAni Users draw several key frames in CACAni System. Second, auto inbetweening and coloring are applied. Then the character data is converted into raster data. Finally, the shadow is rendered.

To focus on this challenging task, we develop a system that can produce instant shadowing automatically and intuitive editing for the sequence created in CACAni System. Our final goal is to enable CACAni users to easily render desired shadows for characters or objects in a shorter time than to draw shadow by hand. Our system requires only hand-drawn key-frames of characters or objects as inputs from CACAni System itself.

## 1.1 System Overview

First, CACAni users need to draw key-frames of characters or objects in the system, and then automatic inbetweening and coloring are implemented to the drawn character or object key-frames. Since the drawn sequence of images in CACAni System is in vector form, the image data format needs to be converted into raster graphics for implementing our shadowing after creating the animation sequence. The images of character sequence or object sequence with alpha value are used as the input for shadowing. Since these inputs are completely in a 2D plane, it is not possible to render shadows by using 3D lighting technique as it is. Therefore, we use the alpha value for creating shadows by Shadow Map in our approach.

Because the inputs are transparent except for the region of character or object, Shadow Map is applied to the opaque region. Generally, Shadow Map is used for a 3D object. On the other hand, our system applies it to the image of character with transparent information in each frame. Then, the desired shadow shape is designed by editing rendered shadows intuitively. Our system enables CACAni users to produce instant Anime-like shadowing by intuitive editing in each key-frame. Figure 3 shows the workflow for shadowing in CACAni System.

# 2. RELATED WORK

Shadow mapping has been developed by Williams [1] as an image-based shadowing technique. Since it utilizes existing hardware functionality such as texturing and depth buffering, the technique is particularly suitable for hardware implementation. Hardware accelerated shadow mapping has been implemented by Segal et al [2]. This method is fundamental for Shadow Map. Their method does not consider the view point so that there is an aliasing problem in close-range shadow rendering from the view point. The following methods try to solve this problem.

PSM (Perspective Shadow Maps) [3] solved the problem for the close-range shadow rendering. While the resolution for the close-range shadow from the view point has been solved in PSM, there is a problem that the resolution of the long range shadow rendering from the view point. LSPSM (Light Space Perspective Shadow Maps) [4] improved on PSM by providing a solution for the problem. TSM (Trapezoidal Shadow Maps) [5] is more advanced. Handling the projected light space as trapezoid, they exploit the shadow map area effectively. The fundamental of this method is to create convex hull by using 2D Quickhull. They define the trapezoid that is perpendicular to the line by connecting near surface and far surface and that has the upper base surface and lower base surface of meeting of the top of convex hull. Furthermore, SSM (Subdivided Shadow Maps) [6] is an improved method for TSM. TSM's transformation is applied to each surface of cone in light space. As for the implementation of soft shadow, PCF (Percentage Close Filter) is well-known. VSM (Variance Shadow Maps) [7] is an improved version of PCF. Normally only the depth value is stored in Shadow Map. However, VSM stores the depth value and the square value of the depth. Additionally in VSM, the Gaussian filter can be applied to shadow texture in same way as applied to color texture. Therefore, this method can handle the filtering process for each shadow map so that it is much more efficient than PCF.

Several user interfaces have been proposed for editing shadows in computer graphics. Nakajima et al. [8] developed the system called "Kagezou" for easy shadowing in Anime production pipeline. We refer to their shadowing approach for the input 2D image plane and inherit their user interface for editing. Though the overall concepts are similar, our system is specialized for CACAni System's production pipeline. Petrovic et al. [9] proposed an innovative interface that creates Anime-like shadows semi-automatically. Our system follows their key concept of shadow generation. While their method requires an amount of time to create shadow animation, our system enables CACAni users to create shadows in the entire sequence instantly. Pellacine et al. [10] developed a user interface that enables animators to edit shadows directly on the editing screen. Like their interactive interface, we implement our user interface that provides CACAni users with intuitive and simple mouse operations. Decoro et al. [11] created a user interface that can be used to design several forms of shadows for rendering non-photorealistic shadows. We are influenced by this concept and develop our system that simplifies shadows as they appear in Anime. Nakajima et al. [12] developed a tool that takes advantage of both 3D and 2D techniques. Their goal was to edit shadows for Anime-like expression to 3DCG model intentionally. Conversely, our method focuses on rendering instantly and editing intuitively shadow throughout the 2D input sequence quickly using 3DCG techniques.

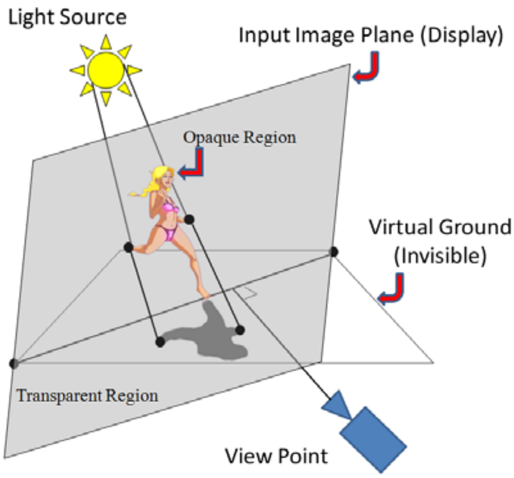

**Fig.4: The Allocation of the Input Image Plane, View Point, Virtual Ground, and Light Source:** The created animation sequence has an alpha value on drawn character. Shadows are rendered by Shadow Map using this alpha value. Background (gray region) is transparent and the character region is opaque.

Regarding highlighting techniques in Anime, Anjyo and Hiramitsu [13], Anjyo et al. [14] developed a tool for editing highlights tweakably. Likewise for shade in Anime, Todo et al. [15] developed a system that enables animators to edit shade according to their preference. While their goal was to emulate Anime-like edit on 3DCG model, we aim to improve work-flow for CACAni System using 3DCG techniques.

Our system obtains the outline of the character as input data, and then uses this shape to render shadows. Juan and Bodenheimer [16] developed a method that extracts the character from existing Anime sequences, and then generates inbetween frames using these characters. This method is useful for creating animations using archived Anime. The concept for creating animation sequence is similar to our system in terms of inbetweening. In CACAni System, key-frames are drawn by CACAni usres and then inbetweening and auto-coloring are implemented.

The fundamental techniques for CACAni System have been developed in following papers: Seah et al, [17], Wu et al, [18], Chen et al[19], Xian et al, [20] and Qiu et al [21], [22]. For creating shadows automatically for a character or object sequence, our shadowing process focuses on the CACAni System work-flow that creates animations by composing several layers without extracting the character from existing Anime.

# 3. SHADOWING IN CACAni SYSTEM

In this section, we describe how to render and edit shadow by using 2D character or object image sequence created in CACAni System. First, CACAni users set sequence of character animation drawn in CACAni System as input. Then, our system renders shadow by applying Shadow Map to the input image plane. The users can then edit the rendered shadow by adjusting parameters with several straightforward operations. As a result, Anime-like shadow is produced quickly from the input animation sequence.

## 3.1 Input Setting: Character or Object Animation Sequence

With our system, CACAni users utilize a character or an object image sequence with alpha value as input. The reason we use an alpha value as a factor of shadowing is that it is clear to distinguish whether a character or an object exists in an image. The region where alpha value exists (see Figure 4 as a reference) is used for shadowing. Thus, our system stores the drawn character sequence with alpha value internally. Usually in Anime production, everything within these image layers is transparent apart from the characters or objects themselves, so that our approach adopts this Anime production workflow in CACAni System. In this way, our system handles alpha value to apply Shadow Map.

## 3.2 Allocation for the Input, View Point, and Virtual Ground

Our system renders shadows onto the virtual ground using input sequence with transparent information. Figure 4 shows the position for the input image sequence plane, view point (camera position) and virtual ground respectively. The input image is set vertically to the direction from the view point. We then define the level of the virtual ground which is invisible to CACAni users. The created shadow is casted onto the virtual ground depending on the position of the light source.

## 3.3 Apply Shadow Map to the Input Image

Although there several possible shadowing techniques exist that could be incorporated into our system, we choose Shadow Map. It is commonly used for rendering shadows [1]. Several research projects have been dedicated to developing a means of producing high-quality shadows in a short amount of time [23]. Since our shadowing is not essentially related to these several improvements of Shadow Map, we do not discuss the technical details in our paper.

While Shadow Map is usually applied to 3D objects, our system renders shadows using a hand-drawn character or object image plane. First, our system stores the distance from the light to a character image plane and the plane (virtual ground) of shadow projection in the depth buffer looking from the light position. A texture visualized from this stored information is called Shadow Map. Then, looking from an initial view point, shadows are finally rendered by referring to the Shadow Map

## 3.4 Editing Operation

### 3.4.1 Position and Shape Editing

Several factors are used for editing shadow position and shape: Light Source Position, Ground Level and Lateral Translation, and Ground Declination (Red arrows in Figure 5).

**Light Source Position:** In CACAni System, CACAni users are required to set the light position first. The shape and the scale of shadows depend on this factor. This editing is achieved by an intuitive controller (In the upper left circle in each image of Figure 6). Figure 6 shows the editing results.

**Ground Level and Lateral Translation:** In order for our system for editing shadow, CACAni users are required to configure a ground level. This ground level is virtually set vertically onto the input plane. CACAni users are able to set the level and position freely so that they can reflect their preference. As we mentioned earlier, the position of shadow let audience know where the character or object exists. To achieve this, our system allows CACAni users to easily edit the ground level and position by straightforward operations.

**Ground Declination:** CACAni users can also configure the ground declination intuitively. This editing enables the users to create shadowing such as walking along the wall.

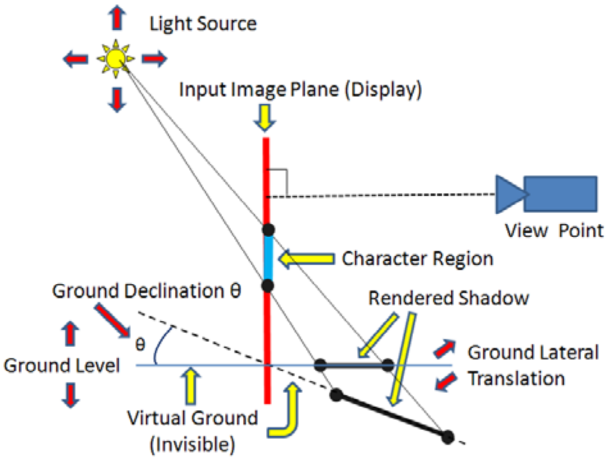


**Fig.5: Apply Shadow Map to the Input Image Plane (Side View):** Input image plane is set vertically from the view point (camera). The initial level of the ground is set on the lowest position of character or object. Declination $\theta$ is defined as the difference between the input image plane and the virtual ground.

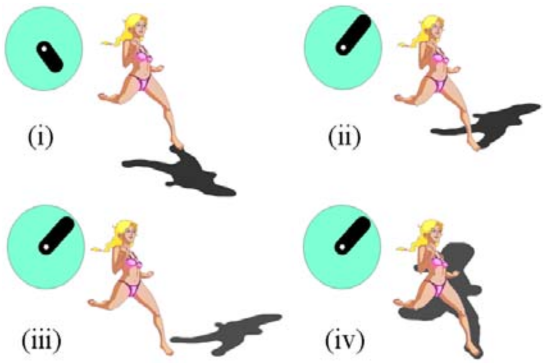


**Fig.6: Editing Result:** Initial State (i), Light Source Position Edit (ii), Ground level and Lateral Translation Edit (iii), Ground Declination Edit (iv).

**Fig.7 (a): Shape Simplification:** shadow shape gradually curling up. The shape is becoming simple.

**Fig.7 (b): Shadow Blurring**

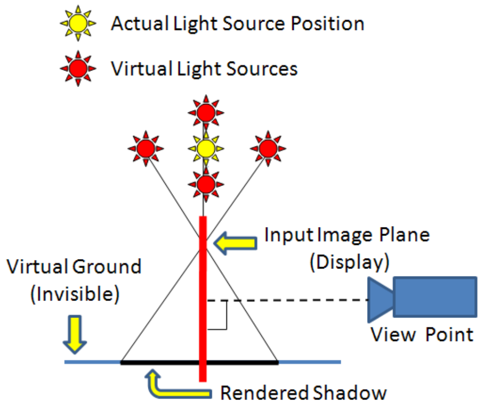


**Fig.8: Algorithm Overview for Overhead Lighting**

### *3.4.2 Shadow Simplification and Blurring*

To make shadows more Anime-like, we implement edge blurring and shape simplification to rendered shadow. Although several ways for blurring and simplification could be implemented, we chose to apply the Gaussian Filter because its implementation is fast enough. As for simplification, the filter is applied to the input character shape itself. In this case, the filter makes the shape of shadows gradually simpler in Figure 7(a) from left to right. As for blurring, on the other hand, the filter is applied to the rendered shadow so that the rendered shadow shape is blurred. Figure 7(b) shows that the boundary of shadow is gradually blurred from left to right.

### *3.4.3 Special Case Handling: Overhead Lighting and Character Position Movement*

**Handling Overhead Lighting:** Since our input image plane does not have a depth, the shadow is not rendered when the light source is placed directly above the input image. To handle the special case, we define the four virtual light sources set axisymmetrically above the input image plane when the light source is directly overhead of the image plane region. We define this domain where the light would be set as a virtual light area by a certain threshold. Although our system has already given the initial threshold and the number of virtual light sources, this range of the domain and even number of virtual light sources could be defined by CACAni users. Figure 8 shows an overview of the special case handling algorithm in the view from the vertical direction of an image plane.

**Character Position Movement:** When the light source is in the range of the Overhead Lighting, our system needs to handle another case. When the character position in the image is changing in the input animation sequence, the shadow position becomes misaligned gradually because the light source position is not changing. To avoid this, we first obtain the median point of the character's alpha value in each frame. Then the distance of median point between frames is assigned to the light source. As a result, the shadow is rendered under the character during the light source is above the character's overhead (Figure 9). Throughout these editing, the shadow shape and position is easily deformed and translated.

### *3.4.4 Output Data*

In our system, the outputs are generated in separate layers such as an input character or object layer, a shadow layer, and the image as seen in Editing Window. Figure 10(i), (ii), (iii) show output examples. To create the finalized animation, those layers outputs are usually composed by CACAni users. Figure 10(iv) shows the composited result.

## 4. RESULTS AND DISCUSSION

We have an experiment on creating character animation with shadow in CACAni System. This animation is created on an Intel Xeon 2.0 GHz platform with NVIDIA Quadro FX1700, and DirectX 9.0 as the graphics API. The input size is approximately 1000 * 600 pixels which could be used as a previewing size. The frame rate range for editing shadows practically in real-time for all operations. Figure 11 demonstrates a sequence of created animation. The upper row is the input image sequence. The middle row is the animation without shape editing. The bottom row is the animation with Anime-Like shadow. Usually in Anime, detailed shadow is not always required because too much detail disturbs the scene. Therefore the shadow tends to be rendered such as the bottom row images. Moreover our system enables CACAni users not only to create shadow edited only in the first frame, but also to create Anime-like shadow used in several key-frames independently from the inbetweening for input image sequence.

There are two advantages of using Shadow Map for CACAni system production pipeline. One is its high-speed performance that enables interactive and intuitive editing shadows. The other advantage is that Shadow Map is its flexible use with various backgrounds. Currently, in our system, shadows can be rendered on flat ground. Likewise it could be used for complicated backgrounds such as a 3D background. Though simple affine transformation to the input image, calculated by the relationship between input image plane, light, and ground, is possible for casting shadows on the flat ground, simple affine transformation is not practical for a complicated background. Therefore, we chose to apply Shadow Map to the input image sequence in the system.

## 5. CONCLUSION AND FUTURE WORK

We have developed an instant and intuitive system for rendering and editing that enables CACAni users to create animation with shadow easily and quickly on common used computer. In addition, our Shadow Map approach also enables CACAni users to imitate complicated backgrounds such as a 3D background. For creating animation in Anime production from the scratch, animators firstly draw or render images in each layer. They then create the finalized animation by composing these layers. By focusing on this Anime work-flow, CACAni system innovatively improves the efficiency of shadow rendering.

Although Petrovic's method [9] is more effective than previous hand-drawn practices, it is still labor intensive and time-consuming because animators must create character's 3D model frame by frame. On the other hand, by using image sequence drawn by CACAni users directly, CACAni System can create animation with Anime-like shadow in a shorter time than Petrovic's method and is more suitable for Anime production pipeline.

However, several issues need to be addressed in our future work. Currently, when the CACAni users use 2D background, undesirable situation sometimes happened such as covering the object on the 2D background image. To solve this problem an easy interface that enables adding several types of simple primitives should be developed. Furthermore, more intuitive editing such as local blurring, deformation, and boundary emphasis should be incorporated from CACAni users' feedback.


## ACKNOWLEDGEMENT

This work has been supported by the National Research Foundation grant, which is administered by the Media Development Authority Interactive Digital Media Programme Office, MDA (IDMPO). Additionally we would like to thank Jie Qiu, Hidehito Nakajima, Shuhei Wemler, and Shigeo Morishima for their comments and advice to develop our system, Lu Peng and Yang Chen for allowing us to use their characters in our paper.


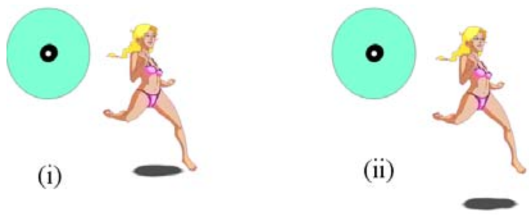


**Fig.9: Editing Result: Handling Overhead Lighting,** This shadowing shows audience the character position. (i) The character is stepping on the ground and (ii) the character is jumping in the air.

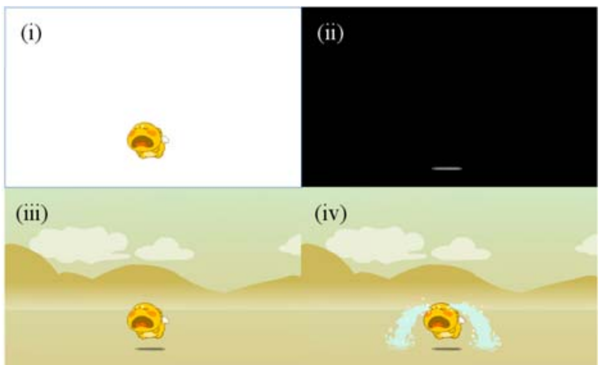


**Fig.10: Outputs:** (i) Character layer, (ii) Shadow layer, (iii) Editing Window which is used for previewing, and (iv) Composited Result with tears added.

*The character QooBeeAgapi and background, © 2004 Lu Peng*

# REFERENCES


[1] Williams H L, Casting curved shadows on curved surfaces. In Proceedings of SIGGRAPH '78, pages 270-274, 1978.

[2] Segal. M, Korobkin C, Widenfelt R Foran J, Haeberli P, Fast shadows and lighting effects using texture mapping. In Proceedings of SIGGRAPH '92, pages 249-252, 1992.

[3] Staimminger, M., and Drettakis, G., Perspective shadow maps. ACM Transaction on Graphics, 21, 3, 2002. Pages 557-562.

[4] Wimmer M, Scherzer D and Purgathofer W, Light Space Perspective Shadow Maps, In Proceedings of Eurographics Symposium on Rendering 2004 pages

[5] Martin T, Tan T-S Anti-aliasing and Continuity with Trapezoidal Shadow Maps, In Proceedings of Eurographics Symposium on Rendering 2004 pages 153-160.

[6] Lloyd B., Yoon S., Tuft D., and Manocha D. Subdivided Shadow Maps. Technical Report TR05-024, University of North Carolina at Chapel Hill, 2005.

[7] Donnelly W. and Lauritzen A. Variance shadow maps. In I3D '06: In Proceedings of the 2006 symposium on Interactive 3D graphics and games 2006, pages 161-165, ACM Press.

[8] Nakajima, H., Sugisaki, E., Welmler. S., and Morishima, S. Automatic Shadowing from Anime Sequence Layers. Poster & Animation Proceedings of the Symposium on Non-Photorealistic Animation and Rendering, 5, 2008.

[9] Petrovic, L., Fujito, B., Williams, L., and Finkelstein, A. Shadows for cel animation. In Proceedings of SIGGRAPH2000, pages 511–516.

[10] Pellacine, F., Tole, P., and Greenberg, D. 2002. A user interface for interactive cinematic shadow design. ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH2002 21, 3, 563–566.

[11] Decoro, C., Cole, F., Finkekstein, A., and Rusinkiewicz, S. Stylized Shadows. Proceedings of the 5th international symposium on Non-photorealistic animation and rendering, pages 77-83.

[12] Nakajima H., Susgisaki E., and Morishima S. Tweakable Shadows for Cartoon Animation. In Proc. of the 15th international Conference in Central Europe on Computer Graphics 2007, pages 233-240.

[13] Anjyo K. and Hiramatsu K. 2003. Stylized highlights for cartoon rendering and animation. IEEE Computer Graphics and Applications 23, 4, pages 54–61.

[14] Aonjyo K., Wemler S., and Baxter W. Tweakable Light and Shade for Cartoon Animation. Proceedings of the 4th International Symposium on Non-photorealistic animation and rendering 2006, pages 133-139.

[15] Todo H., Anjyo K., Baxter W., and Igarashi T. Locally Controllable Stylized Shading. 2007. ACM Transactions on Graphics Volume 26, 3, Article No. 17

[16] Juan, C., and Bodenheimer, B. Re-using Traditional Animation: Methods for Semi-Automatic Segmentation and Inbetweening. In Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation 2006, pages 223-232.

[17] Seah H S, Wu Z, Tian F, Xiao X, and Xie B. Artistic brushstroke representation and animation with disk B-spline curve. In ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2005), pages 88-93, 2005.

[18] Xiao X, Seah H S, Wu Z, Tian F, and Xie X. Interactive free-hand drawing and in-between generation with disk B-spline curves. In The Conference of Multimedia Arts Asia Pacific, 2004.

[19] Wu Z, Seah H S, Tian F, Xiao X, and Xie X. Simulating artistic brushstrokes using disk B-spline curves. In The Conference of Multimedia Arts Asia Pacific, 2004.

[20] Chen Q, Tian F, Seah HS, Wu Z, Qiu J and Melikhov K. DBSC-based animation enhanced with feature and motion. Computer Animation and Virtual Worlds, 17(3-4):189-198, 2006.

[21] Qiu, J, Seah HS, Tian F, Wu Z, and Chen Q. Feature- and region-based auto painting for 2D animation. *The Visual Computer*, 21:928-944.

[22] Qiu J, Seah HS, Tian F, Chen Q, Wu Z, and Melikhov K. Auto coloring with character registration, in Proceedings of the 2006 international conference on Game research and development CyberGames '06, 2006; pp. 25-32.

[23] Lokovic, T., and Veach, E. Deep shadow maps. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques 2000, pages 385-392.


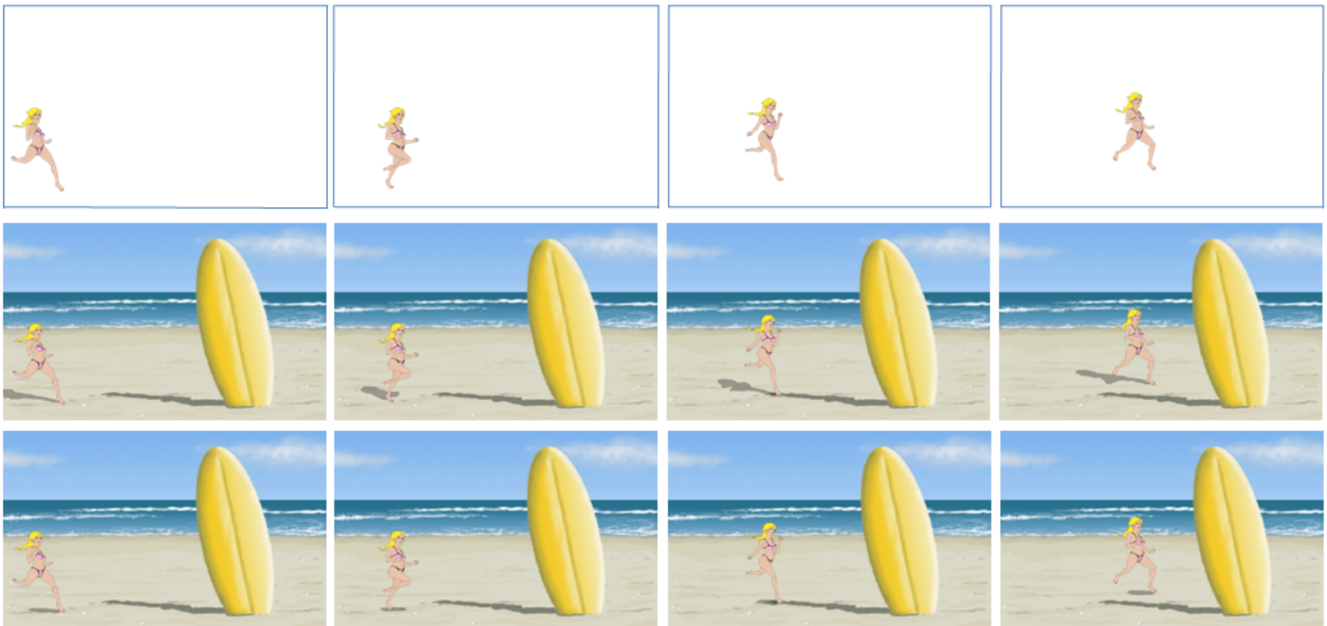

**Fig.11: Animation Result:** Input character animation sequence (upper row), animation result with detailed shadow (middle row), animation result with Anime-like shadow (bottom row)