Interactive Shade Control for Cartoon Animation

Yohei Shimotori\textsuperscript{1} Hidehito Nakajima\textsuperscript{2} Eiji Sugisaki\textsuperscript{3} Akinobu Maejima\textsuperscript{4} Shigeo Morishima\textsuperscript{5}
Waseda University

1. Introduction
Shade in traditional cel animation is an essential element and plays a symbolic role in the artistic portrayal of character and scene. The shade on a character’s face can be used to show emotion such as anger, sadness and hatred. Despite its usability, shade is relatively quickly or roughly drawn in animation for the following reasons. Firstly, cel animators require a large amount of time to draw shade. Secondly, because of the time requirements, producing complicated shade is expensive. On the other hand, 3D models enable users to more easily render or re-create shade than cel animation techniques. Consequently, 3D models are being gradually introduced into cartoon animation. Using Phong’s 3D shading model, however, shade is rendered too realistically for cartoon animation, and is therefore not appropriate for traditional-style cel animation. To solve similar problems with highlighting, Anjyo and Hiramatsu proposed a procedural method for designing stylized highlights [Anjyo and Hiramatsu 2003]. This concept has been applied to shade in our system.

We have developed an intuitive, direct shade manipulation system for 3D models. Our system enables users to practically create and edit shade in real-time. Users are required only to make intuitive mouse-clicks to create and edit shade. Since our system can create and edit shade using only a few parameters, the user’s interaction is considerably reduced compared with hand-drawn animation. Furthermore, since our idea is to manipulate only selected area, natural looking animation can be created in almost real time with maintaining 3D models’ attributes.

2. Defining Shade Areas
First of all, users roughly select an area for creating and editing shade and then click around the selected area. This area is then defined as a circle. Users can change the size of the circle according to their preference using a slide control. In addition, the area can be increased whenever users click what they would like to edit. Furthermore, our system allows users to reset edited parameters so that they can easily re-start to edit a selected area.

Shade in cartoon animation mainly consists of diffuse reflection factors, which are denoted by the following equation.

\[ I = I_o k_d \left( \hat{N} \cdot \hat{L} \right) \]  \quad \cdots (1)

\( I \) is the value of a diffuse reflection on the selected point, \( I_o \) is the intensity of the light source, \( k_d \) is the diffuse reflectance of an object, \( \hat{N} \) is the normal vector of a vertex, and \( \hat{L} \) is the vector showing the direction to the light source. Our idea is to edit selected areas by manipulating the normal vector. The manipulated vector \( \hat{N}' \) is calculated by equation 2.

\[ \hat{N}' = \hat{L} \times (\hat{N} \times \hat{L}) \]  \quad \cdots (2)

Equation 2 is applied to the vertex which has the largest value of inner products of the vector \( \hat{L} \) and the normal vector \( \hat{N} \) in the selected area. (Equation 3).

\[ Vertex_{max} = \max_{i \in area} \left( \hat{N}_i \cdot \hat{L} \right) \]  \quad \cdots (3)

\( m \) is the number of vertices in the selected area. Then, normal vectors of other vertexes are made parallel to the normal vector \( \hat{N}' \) calculated by equation 2. In addition, because a point light source does not move when this normal vector is manipulated, the other parts of the scene remain unaffected. Therefore users can create and edit shade without being concerned about changing other attributes within the scene.

3. Results
Our system runs interactive rates on a 2.3GHz Intel Core (TM)2 T7600 CPU with NVIDIA GeForce Go 7400 GPU. The demo movie demonstrates that shade areas have been successfully created and edited using our system. Figure 1 shows examples of the application of our system in a practical situation. First, we prepared the original image (Figure 1 top left). Then, we created and edited the shade area using mouse-clicks (Figure 1 top right). This enabled users to create and edit the character’s apparent negative emotion (shown on the lower right, Figure 1) which is more appropriate than the version on the left for traditional cartoon animation (the area around the eyes is darker in the image on the right than in that of the left, resulting in a gloomier expression). One of the advantages of our system is that users can create and edit shade without moving the light source. Another advantage is that edited shade appears to change naturally, despite the movement of the light source or the character. In addition, once normal vectors are manipulated, their attributes are maintained until users reset their parameters. Moreover, our system effectively handles key-frame animation.

4. Discussion
Our system enables users to effectively create and edit shade in real-time on a 3D model by intuitive manipulation. Despite using simple algorithms, we can express character emotion successfully, as shown in demo film sequences. However, in the present version of our system, users must click continuously to create and edit shade. We therefore need to further improve our system so that users can specify a selected area more intuitively, and create shade more efficiently.

5. Acknowledgement
This research is supported by the Japan Science and Technology Agency, CREST project.

Reference