Shadow Play (sap 0057)

Cem Sina Cetin*
Sabanci University

1 Introduction

Traditional shadow play is a form of art, which has been used both for folklore storytelling and entertainment. Its existence extends from primitive cultures to modern world as a performance art. Because of its nature, as the depicted scenario gets more complex and the number of actors increase, it gets harder for only one person to control the play. In this project, the aim is to adopt the basic principles of shadow play culture and build a novel environment to mimic this tradition. Taking advantage of computer aid, "Shadow Play" proposes a new interface for the interaction process of a play. It not only makes it possible for only one person to create the entire scene and actors, but also transforms the application into an autonomous installation.

2 Exposition

The system is made up of a camera, a white semi transparent projection plane, a projector and a computer. The plane serves as the environment, the projector works as the light source. A video camera constantly feeds the image of the plane to the computer where the necessary image processing occurs.

The application runs in two phases. In the first phase, scene construction occurs and it starts with a blank white image as the background. Just like traditional shadow play, the user creates a shape by casting a shadow (preferably his own) on the screen. However, unlike the tradition, after staying still for a period of time, the image of the shadow is captured via the camera and added to the initial blank image being currently projected on screen, thus making the shadow a permanent component of the envrionment. Iterating this process, a single person is able to construct the entire stage.

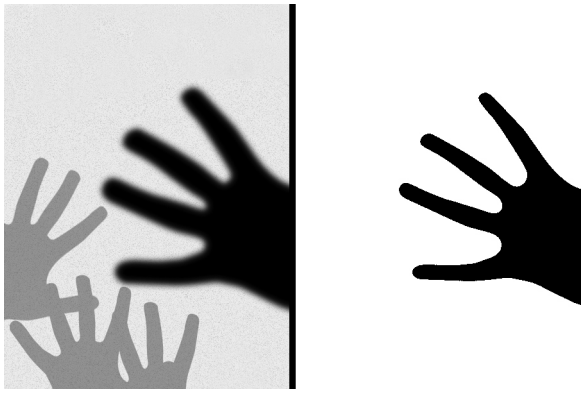

Figure 1: *A manually created image and its result, after the image processing algorithm of Shadow Play is applied to it.*

The camera is placed behind the semi transparent plane so that it can capture the entire scene and only what is being projected. It continuously receives the current frame and checks the amount of movement in the scene by comparing the current and previous frames pixel-wise. If the number of noticeably different pixels is below a threshold, stream is considered to be staying still. If there is no displacement for a period of time, the screenshot from the camera is added to the current background image; White pixels are considered to be transparent. This way, if there is a shadow being cast on the screen, it is permanently added to the background. Saved shadows are projected as light gray, so that a real shadow on the screen can be picked among the saved ones by its difference in brightness. Thus, not only the saved shadows but also any noise generated by the camera is eliminated during capture. At the end of the first phase, saved shadows are pulled back to black.

In the second phase, "actors" are released into the created environment. For simplicity, they are chosen to be butterflies. They are captured just like the first phase, only difference being that a specified blank area is provided for a butterfly's shadow.

Butterflies' movements are determined like a state machine with the following rules: **initialize to flying**, **flying to resting**, **resting to flying or walking** and **walking to flying**.

Depending on the current state, following coordinates and the wing separation of the fly are calculated and drawn accordingly. The butterflies randomly change their direction by five degrees at a time and keep moving at the same direction for a random amount of time. They recognize shadows (collision detection is handled by checking if the following coordinate corresponds to a black pixel on the background image or not) and do not penetrate into them. While flying, butterflies steeply change their direction if a dark spot is encountered. During the resting state however, they descend until landing on a shadow and may walk on it if there is room to do so.

Necessary image processing is handled by OpenCV library whereas motion of the butterflies in 3D space and the rasterization of the scene are done by using OpenGL API.

3 Conclusion

Because of the way the application works, quality and adjustment of the hardware largely determine the overall performance of the application. The lighting condition of the environment is also critically important. Since the images appearing on the screen are directly fed from the camera, focus, resolution and image quality are limited by camera capabilities. Except these factors, behavior of the actors and the scene construction works as expected.

Although the main goal is to mimic and hopefully advance shadow play into a new level of interaction, this project can branch into other applications of shadow theater. For instance, adding different types of actor, interacting with them in real time, adding environmental effects such as wind or stimulating the actors via sound would not only increase the possibilities of the outcomes, but also transform this application into a tool of narration, a contemporary environment for an age old performance art.

References

DAVE SHREINER, MASON WOO, J. N., AND DAVIS, T. 1995. *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL.* Addison Wesley Professional.

*e-mail: i.am@cemsinacetin.com