THE VIRTUAL DAGUERREOTYPE

Mehmet K. Kocamaz, Adnan Kaya, Eun-Young Elaine Kang, Alexandre François
Integrated Media Systems Center
University of Southern California
Los Angeles, CA 90089
{kocamaz, akaya, ekang, afrancoi}@usc.edu

Abstract

In this report we present a system to simulate daguerreotype pictures. Daguerreotypes are one of the earliest forms of photography. They are created by using a direct-positive process that produces a very detailed image on a sheet of copper plated with a thin coat of silver without the use of a negative. The system we offer is built on IMSC's handheld mirror simulation system and adds image composition modules to reproduce the feeling of looking into a mirror and seeing a daguerreotype image superimposed on your reflection. The software components of the system are developed using the Modular Flow Scheduling Middleware (MFSM), an open source implementation of IMSC's Software Architecture for Immersipresence (SAI).

Keywords: Daguerreotype simulation, real-time video processing, human-computer interface.

1 Introduction

Daguerreotypes are one of the most distinctive and recognizable images throughout the entire history of photographic processes. A daguerreotype is made on a highly polished metal plate without the use of a negative and the image has a reflective, mirror-like appearance. The image is often likened to the small holograms that can be seen in a child's set of stick-on or affixed to a credit card [**1**]. Similar to the holograms, a daguerreotype image is only visible from certain angles. Daguerreotype images, most of which are portraits, are encased in velvet, leather, glass and metal trim. However, the fragility of these objects demands them to be displayed in cases, under carefully controlled light.

Figure 1: A Daguerreotype Picture[1]

[1]Courtesy of http://lb-web.com/daguerreotypes

Our approach to solution is to use the virtual mirror system that has already been developed in MFSM (Modular Flow Scheduling Middleware). The virtual mirror system is basically a generic interface for the virtual daguerreotype. We use it to obtain the viewer's image. In virtual mirror system, the image of the user is taken with a camera and reflected on the LCD screen with respect to the data that comes from two magnetic trackers. One tracker is for tracking the mirror and the other is for tracking user's head's position. Trackers provide the rotation and position information of the user's head and mirror. By using some image processing techniques from Intel's Image Processing Library and writing a compositing code we superimpose the viewer's image with the daguerreotype image. Superimposition depends on the position and the angle of the LCD screen with respect to the viewer, so that at some specific angles the viewer sees his/her image superimposed with the daguerreotype image. At other angles, depending on the position, the viewer sees the daguerreotype in different colors and different visibility. This provides us the special property of the daguerreotype pictures, being seen at just some certain angles.

This paper is organized as follows: In section two, we outline the geometric analysis of the mirror and camera imaging, and explain the necessary image transformations for most perfect display. In section three, we give information about the hardware design of the project and explain what each hardware unit of the project does. In section four, we delineate the software setup of the project. Section 4.1 is focused on the MFSM, and section 4.2 is focused on the image composing and interpolation. Paper is concluded at section five.

2 Geometric Analysis

The geometric analysis of virtual daguerreotype is same as virtual mirror's geometric analysis. A very detailed analysis of the mirror geometry can be found in virtual mirror paper [**2**].

3 Hardware Configuration

The virtual daguerreotype system has the same hardware setup as the virtual mirror system [**3**]. As illustrated in figure 2 the system is composed of four major units: an LCD screen manipulated by user, two magnetic trackers having six degrees of freedom which give the real time position and orientation of the mirror and user's head, a camera fixed on top of the LCD screen to capture user's image.

As seen in figure 3, the image of the user is taken with the camera as a video stream. This stream goes into processing unit to be combined with the daguerreotype pictures stored in the database. Each frame of this stream is combined with a daguerreotype image to produce a new stream. This new stream is displayed on the LCD screen according to the real time orientation of the mirror with respect to user's head, which is provided by the trackers.

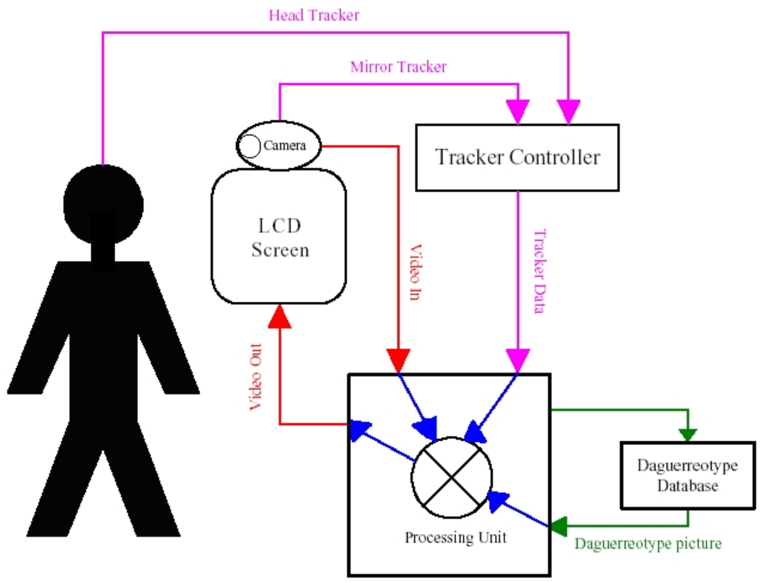

Figure 2: The Virtual Daguerreotype System Diagram

4 System Integration

4.1 MFSM

The software component of the system is built in Modular Flow Scheduling Middleware (MFSM) [**4**], the open source implementation of IMSC's software architecture for Integrated Media Systems. The most important reason for choosing an environment like MFSM is because of its extensibility and modularity. The applications built in MFSM are composed of major processing units called cells. The extensibility feature is that MFSM allows adding or modifying cells to produce new programs. As it can be seen from figure 3, which shows the flow graph of the virtual daguerreotype application, there are five processing units, namely *cells*. The task of first cell is to capture the image of the user via the camera. After capturing the user's image in real time, these images are conducted to the second cell as a stream. In second cell, the data from the trackers are captured and synchronized with the stream that comes from the previous cell. This synchronization task is underpinned by MFSM's data carrier pulse. In MFSM, all data corresponding to the same time stamp are stored in the same data structure called pulse. To clarify, we can make an analogy: Streams can be thought as pipes in which pulses flow in just one direction. After second cell, the stream flows into third cell where the image of the user undergoes necessary transformations to be displayed on the screen correctly. Forth cell is the cell where most important tasks are carried out. The superimposition is done in this forth cell. According to data that comes from trackers, an appropriate daguerreotype picture is formed by interpolating different images which are loaded from database and is combined with the transformation of the user's image. After composition, resulting image is displayed on the LCD screen.

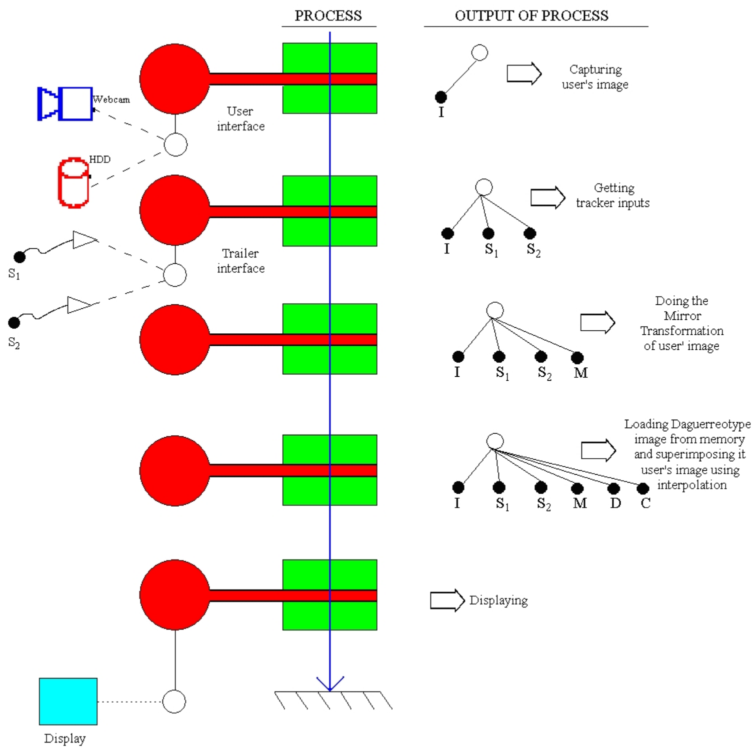

Figure 3: Virtual Daguerreotype Application Graph

4.2 Image Composition and Interpolation

As aforementioned above, there are two images that are composited at each instant of time. First image is the mirror transform of user's image; second one is daguerreotype image that is an interpolation of four images loaded from database where various daguerreotype images are stored. These various daguerreotype images are taken at different angles like $\pm 0°, \pm 10°, \pm 20°$ and $\pm 30°$... so forth. In this stage, first problem that needs to be solved is the task of developing a theoretical model for a real daguerreotype. Daguerreotypes are special types of images, they can only be seen a certain angle. At other angles, depending on the position, illumination and some other factors, daguerreotypes appear in different colors and have different visibility. The image pattern stays same independent of angle user's view. However the color change as we rotate the daguerreotype may not be linear with respect to angle change. That is why, firstly we have to find out and formulate color behavior of a daguerreotype image with respect to angle of view. In order to achieve this task, we obtained a daguerreotype image, and take its photos at different angles on both X and Y planes. This gave us a matrix of images similar to the following figure 4:

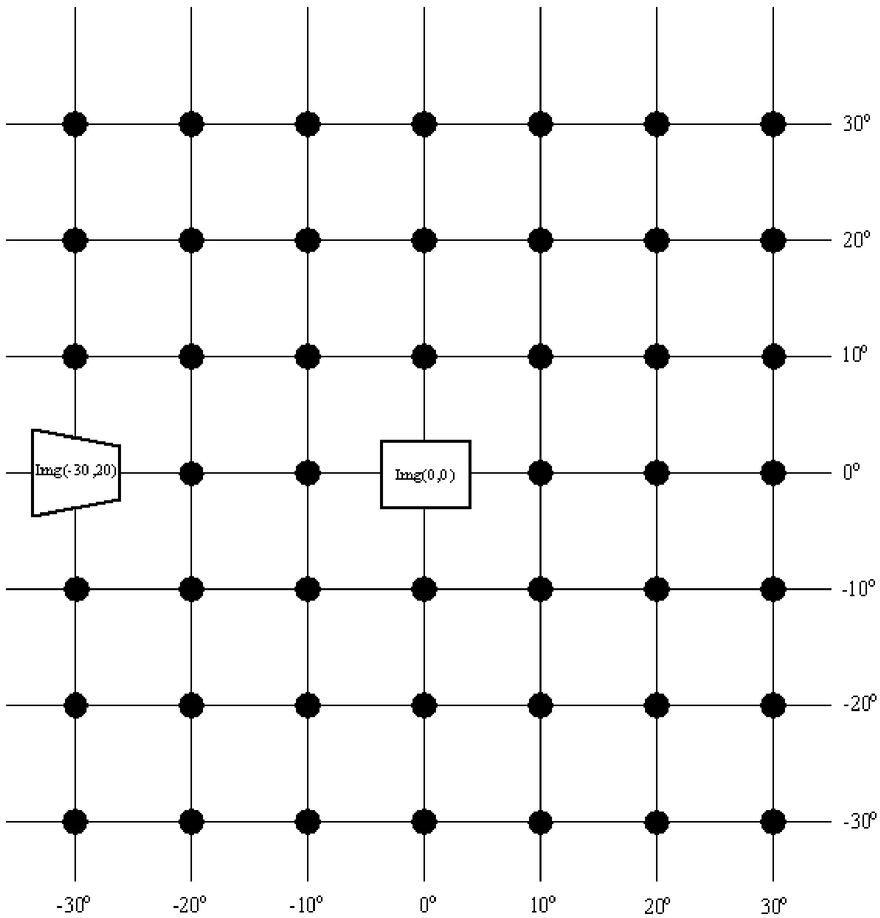

Figure 4: Daguerreotype images taken at different angles

Above, black dots represent images taken at the angle specified by the coordinate system provided. Because of the perspective distortion, the image of a rectangle appears to be quadrangle. That's why; we need to rectify all images in figure 4. To do that, we crop the region of interest in each image. Since two images of the same planar object are related by a linear transformation in homogeneous coordinates and we know the actual size of main image located at $(0°, 0°)$, by using a transformation matrix H, we can easily obtain rectified image that has same dimension and resolution with actual image. Let's say, quadrangle image pixel coordinate is $Q\left(x_1, y_1\right)$ and final rectified image pixel coordinate is $R\left(x_2, y_2\right)$ to find corresponding pixel in final image, we use the formula:

$$\begin{bmatrix} x_2 \\ y_2 \\ w \end{bmatrix} = H_{12} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \qquad \begin{matrix} x_2 = x_2/w \\ y_2 = y_2/w \end{matrix} \tag{1}$$

Above, H_{12} is our pixel mapping matrix. Since source image and final image resolutions are not same, after using above formula; some pixels can not be filled and they do not have their RGB values, to prevent this we used linear interpolation to fill empty pixels in final image. Once we have all the images rectified, next step is finding desired daguerreotype image. After getting the position of the virtual mirror with respect to the user from the trackers, we calculate the relative angle between user head and mirror in X and Y coordinates and select four closest images on both X and Y planes. For example, if the angle between the mirror and user is $(17°, 24°)$, we take the images shown with red dots to produce the image shown by the blue dot in the figure below:

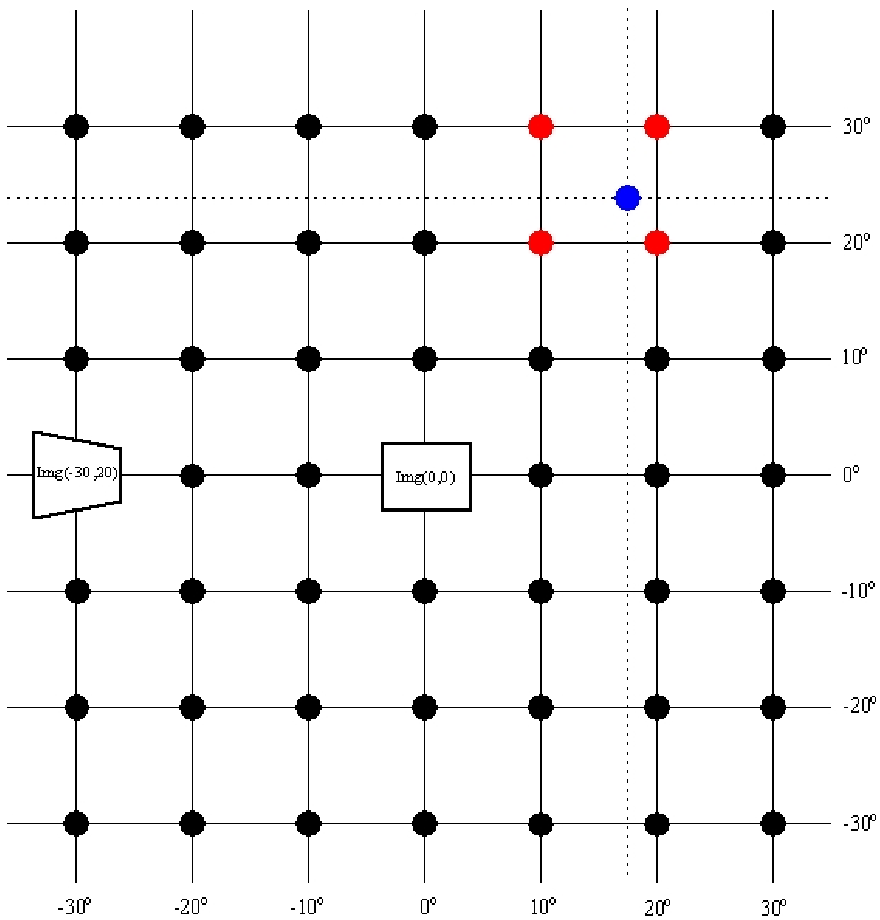

Figure 5: Illustration of composition

To achieve most close intermediary image to real daguerreotype, some other interpolation techniques can be developed to find image's RGB values, but in this stage of research, we have simply used linear pixel interpolation between selected four images:

$$F(x, y) = (\theta - \alpha)A(x, y) + (\theta - \beta)B(x, y) \tag{2}$$

F is final image, two images are A and B, having angles α and β, θ is the relative angle between the user and the mirror. We use this function three times, first to interpolate two top red dot images, second to interpolate two bottom red dot images in figure 5, and finally to interpolate those two intermediary images to obtain our desired daguerreotype image.

After interpolation, resulting picture is superimposed (with an alpha channel) on top of the mirror transformation of the user's image using Intel's Image Processing Library (IPL) [**5**], and then is displayed on the LCD screen. Some results of application can be seen in figure 6 and figure 7.

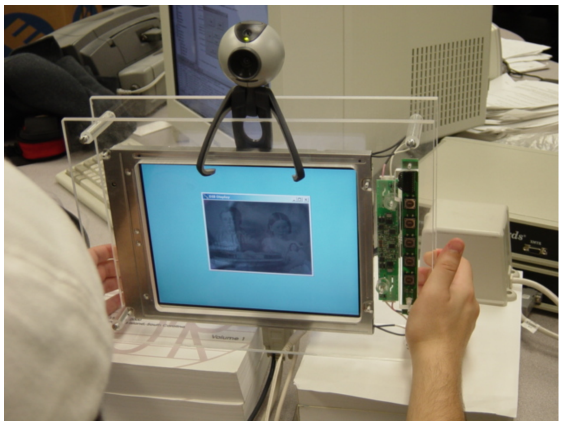

Figure 6: Virtual Daguerreotype demo, the user is rotating the mirror.

Figure 7: A close look at the image in Figure 6

5 Conclusions

We presented a system to simulate daguerreotypes. This system is an efficient, extensible system. These features are ensured through the use of MFSM. As aforementioned above, daguerreotype simulations possess one major component, which is the composition module.

Since daguerreotype pictures are very fragile, they are not available to public. That is why, this project can be used to make visitors feel museum exhibition. What is more, this project is an excellent example of Human-Computer Interface, which is a key of doors to various interactive experiments in entertainment, art, and communication.

6 References

[1] What is a Daguerreotype. http://lb web.com/daguerreotypes/what_is_a_daguerreotype.htm
[2] A Handheld Mirror Simulation Alexandre R.J. François and Eun-Young Elaine Kang, IMSC, 2002
[3] A Handheld Mirror Simulation Alexandre R.J. François and Eun-Young Elaine Kang, IMSC, 2002
[4] A. R.J. François, "Modular Flow Scheduling Middleware", http://mfsm.sourceforge.net/
[5] Intel Image Processing Library. http://www.intel.com/support/performancetools/libraries/ipl/index.htm

7 Acknowledgements

The authors wish to thank M. Lazzari, M. McLaughlin, Gèrard G. Medioni and W.L. Wong at the University of Southern California. This research has been funded by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152; and by a grant from the USC Arts Initiative.