CFBOX™ : Superimposing 3D Human Face on Motion Picture

Sonou Lee[1], Kwangyun Wohn[1], Yoosoo Ahn[2] and Sukki Lee[2]
[1] *Division of Computer Science, EECS, KAIST,*
373-1, Kusong, Yusong, Taejon, 305-701, South Korea
[2] *Digit@lion, Suite 902, Daegong BD., Yuksam, Kangnam 135-080, Seoul, South Korea*
solee@vr.kaist.ac.kr

Abstract

We present the standalone kiosk named CFBOX™. The CFBOX™ is a kind of personal commercial film studio that replaces the face of the person in an existing CF with that of the modeled face of a user by utilizing 3D face on & off integration technology on real-time. It also has photo manipulation functionality to change the texture of modeled face to one's taste. In short, the CFBOX™ is a space where one can create personalized digital video. The prototype of the CFBOX™ is implemented and exhibited in SEE-KAIST' 2001.

1. Introduction

In general, superimposing means 'put one thing on top of something else' or 'inserting or overlaying some image over other image'. The simple example of superimposing is a caption on VCR movies. A popular synthesized photo, which substitute actor's face with personal face is also good example of superimposing. For another example, in weather forecasting on TV, human image is captured with chroma-key background and is just overlaid in front of weather chart rendered by a computer. Virtual stage system is similar but more complex. Human image is also captured with chroma-key background, but it is located exact position in virtual environment rendered by computer with depth information. Since seamless synthesizing of rendered virtual object with real video image is important in augmented reality, efficient and realistic superimposing technique has been addressed recent years [4][8].

In this paper, we present the CFBOX™ system, which superimposes individual's face over actor's one on motion picture. The CFBOX™ is somewhat similar to augmented reality system or general super imposing system. The main difference between the CFBOX™ with other superimposing system is that the CFBOX™ superimposes user's textured 3D face model with correct position and direction information of original face that will be hided. To combine per-frame information of position and direction into single efficient structure, we designed the SIDB and implemented the SIEtool for constructing the SIDB manually. The CFBOX™ is standalone kiosk and has user-friendly interface, so novice user could operate it without difficulty.

The next section will discuss the overall structure of the CFBOX™ and its three-component process. Implementation issue and the result of experiments on exhibition are described in Section 3, and Section 4 discusses further work to improve the CFBOX™.

2. The CFBOX™ System

The CFBOX™ is a standalone kiosk that can perform taking user's photograph, modeling 3D face model, superimposing user's face over actor's one on motion picture, and printing out results such as a photo sticker or a CDR. As shown in figure 1, the system consists of three parts: SIDB constructing process, face model making process and superimposing process. In fact, SIDB constructing process is performed off-line and just The SIDB is included in the CFBOX™ system. We will describe these parts more precisely in the following subsections.

2.1. SIDB Construction

2.1.1. What is the SIDB?: The SIDB – Scene Information DB – is the collection of per-frame information, which consists of three parts: background image, basic information, and additional information. Background image is frame image itself. The basic information includes information for face and light of the scene. The face information includes the position and direction of actor's face. The light information includes position, direction, intensity, and type of light in the scene. The additional information includes information for masking, motions blur, and others. Currently, we does not use motion blur information.

2.2.2. Preparation: Although there are so many resource of motion picture, they can be categorized in just two format, analog and digital. Film, VHS tape, Betacam format is the type of analog source, and DV, AVI, MPEG format is the type of digital source. If given source is analog format, we convert it to digital with frame capture board such as MIRO-DV500 of Pinnacle Systems, Inc. Then we export motion picture into sequence of image frame using Premiere software of Adobe Systems, Inc. before any further processing to construct the SIDB.

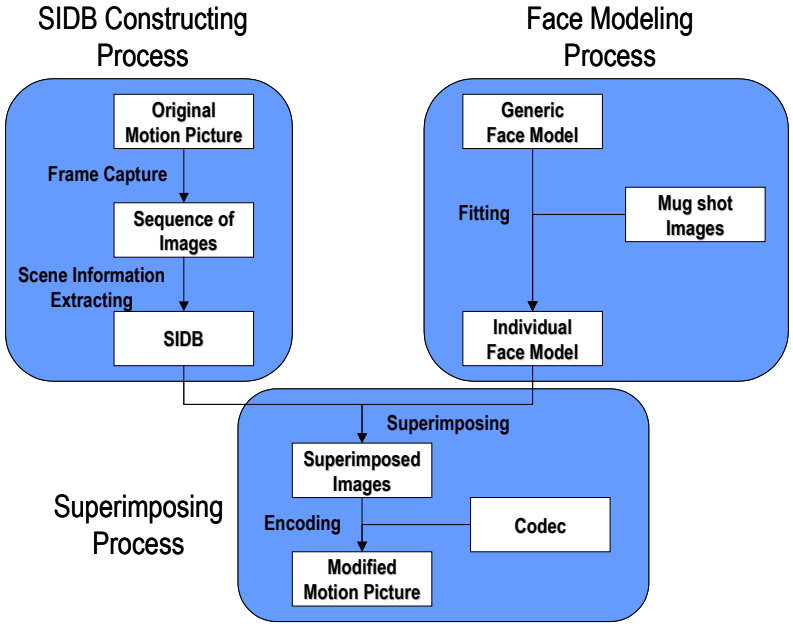

Figure 1. Overall system diagram

2.2.3. Scene Information Extracting tool: To obtain basic information of the SIDB from the sequence of image frame, we could consider automatic approach such as face detection, face tracking or manual approach using adequate tool. Many researchers have been suggested good algorithm for face detection and tracking from image sequence automatically [2][3][7]. Unfortunately, we did not find optimal method for our system since they require for front viewing face image for initializing and does not support rapid or wide-angle head motion. So we dose not use fully automatic approach to get basic information at present. For manual approach, we made the GUI based SIDB constructing program named SIEtool as shown in Figure 2. With the SIEtool, user can set face and light information per frame using GUI. In fact, though carefully designed functionality such as initial face position setting, key-frame interpolating, and frame copy/paste, are supported in the SIEtool, it is somewhat labor-intensive work to make the SIDB of long time motion picture. To alleviate hand job, semi-automatic approach will be combined into the SIEtool soon.

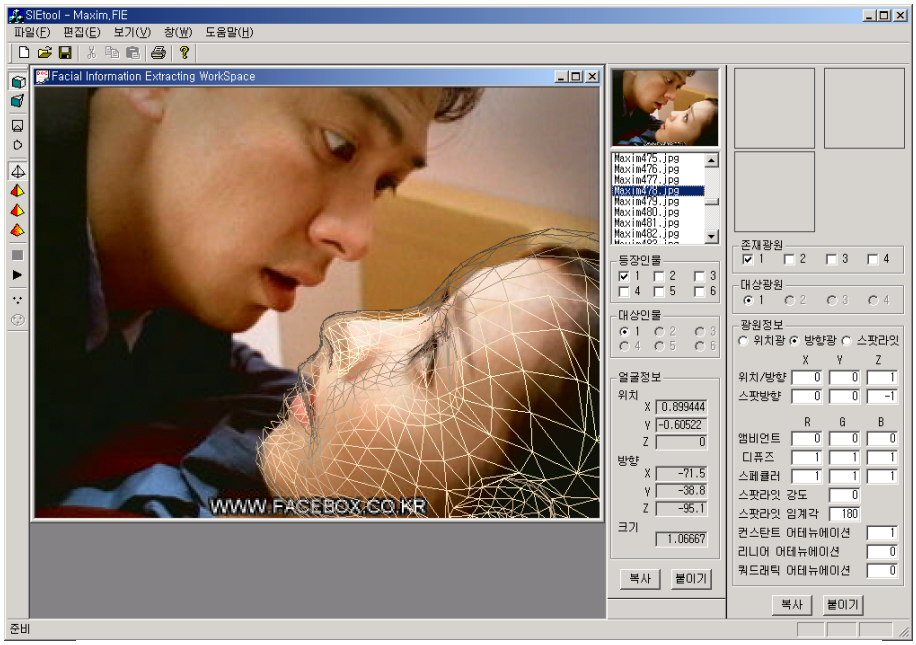

Figure 2. Execution example of the SIEtool

2.2. 3D Face Modeling

In face modeling process, we make the 3D face model for a specific user by deforming the generic face model with feature points [1][5][6]. This face model will be superimposed over actor's one lately. The overall process is shown in Figure 3. At first, we take the mug shot image for a specific user and locate predefined feature points on each image. The CFBOX™ provides GUI for this process. Then fitting algorithm based on radially symmetric basis function generates specific user's 3D face model.

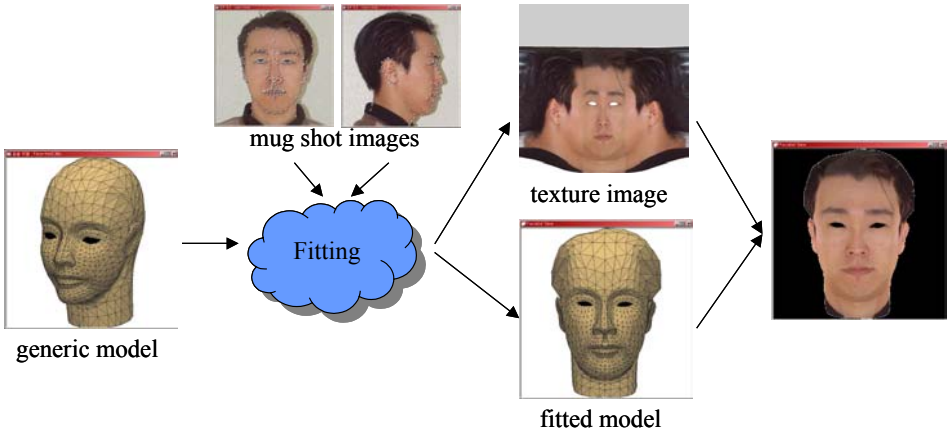

Figure 3. Face modeling process

The displacement vector of feature vertex is obtained by vector from the generic position to the located position and its normalized displacement vector is obtained by solving the following simultaneous Equation 1:

$$\begin{aligned} U_f^1 &= \sum_{k=1}^{n} f(r_{k1}) u_f^k, \\ U_f^2 &= \sum_{k=1}^{n} f(r_{k2}) u_f^k, \\ &\vdots \\ U_f^n &= \sum_{k=1}^{n} f(r_{kn}) u_f^k \end{aligned} \qquad (1)$$

where $U_f^l, \{l = 1,2,...,n\}$ is the displacement vector of feature vertex l, r_{ij} is the distance between feature vertex i and feature vertex j, $f(r)$ is a radially symmetric basis function that provides a large value for small r, n is the number of the feature vertices, and $u_f^j, \{j = 1,2,...,n\}$ is the normalized displacement vector for feature vertex j.

All other vertices, called non-feature vertices, are positioned by interpolating the normalized displacement vector of the feature vertices. The displacement vector u_{nf}^i of non-feature vertex i is calculated by Equation 2:

$$u_{nf}^i = \sum_{j=1}^{n} f(r_{ij}) u_f^j \qquad (2)$$

Since fitting algorithm is processed for front and side view image separately, full 3D face model is obtained by combining the result of both images. After we get user's face model, we generate cylindrical texture map by projecting this face model on both mug shot images. To join both images smoothly, [1] used blending both image and [5] used blending with multiresolution image mosaic. We use blending with cylindrical normal map. Each pixel of cylindrical texture map is calculated by the weighting of its normal from cylindrical normal map and both texture

map from front and side image. To obtain cylindrical normal map, we use OpenGL rendering by setting vertex normal value for vertex color. Although result normal map has quantized value because RGB color depth could have up to 8bit each in OpenGL, this technique is very simple and quick method to get cylindrical normal map and provides sufficient result for our system.

2.3. Superimposition

In superimposing process, system makes a result motion picture with superimposing for each frame and encoding them together. Superimposing is performed by three times rendering. At first, background image is rendered with minimum depth value in blank image. Then masking information is set to only depth buffer with maximum depth value. With masking information, we could define area whether rendered or not as shown in Figure 4. Currently we make masking information by using graphic editing tool. In OpenGL rendering system, stencil buffer may be used instead of depth buffer, but it provided slow performance in our experiments. Finally, individual's face model is rendered. For better superimposing result, we could consider image retouching or enhancement techniques. In most case, image enhancement techniques have much processing time. Superimposition and playback has to be performed on real-time, so time-consuming image enhancement method are not adequate for the CFBOX™. Nowadays, most current available graphic accelerator supports anti-aliasing and anti-aliasing does not have much effect on rendering speed. We use only anti-aliasing technique for image enhancement during the superimposing process.

After superimposing for each frame is done, system encodes all image frame and make an AVI format file. We currently use the Indeo® video 5 codec as encoder for making motion picture. Since Indeo® is a software encoder, encoding speed is not fast. We will change it for hardware mpeg encoder in the next version the CFBOX™.

(a) (b) (c)

Figure 4. (a) : an original image, (b) : a superimposed image, (c) : a superimposed with masking

3. Implementation and Experiments

Recently we make the prototype of the CFBOX™ system and open it to the public in SEE-KAIST '2001 exhibition. The CFBOX™ is a kind of personal commercial film studio that replaces the face of the person in an existing CF with that of the modeled face of a user by utilizing 3D face on & off integration technology on real-time.

The CFBOX™ is implemented on a Pentium III 866MHz running under Millennium Edition with nVidia GeForce2 GTS 3D graphics accelerator. The program is written entirely in C++. To capture user's photo online, we use two CCD cameras one for front and other for side each. Currently, we adopt joystick and button for user input device, but it will be changed for touch

Figure 5. SEE-KAIST'2001 Exhibition

screen interface in next version. The CFBOX™ provides stepwise GUI menu and speech guidance to help novice user. User can decorate their cylindrical texture map with funny icons. This functionality enjoys general popularity in the exhibition. At last menu stage, user can select what he/she wants to get for an output. The CFBOX™ provides user with a print or a CDR for an output. A print has some frames of predefined image set of superimposed motion picture. A CDR contains both original and superimposed motion picture in AVI file format. The CFBOX™ also sends superimposed motion picture to other via email when user inputs email address.

Average SIDB size for 30 seconds CF (about 900 frames) has 13.5 MB for image frame itself, 0.45MB for basic information, and 0.5MB for masking information. Average one full operation takes about 5 minutes. Superimposing and playback for all image sequence are performed in video frame rate (=30fps) during the operation. The sample superimposed image sequence is shown in Figure 6. Some result movie can be downloaded at http://vr.kaist.ac.kr/~solee/cfbox/movie/.

4. Conclusions and Future Work

In this paper we have presented a standalone kiosk named CFBOX™ that can perform taking user's photograph, modeling 3D face model, superimposing user's face over actor's one on motion picture, and printing out results such as a photo sticker or a CDR. We opened our system to the public in exhibition and received many useful feedbacks from novice users.

Nevertheless, the CFBOX™ can still be improved on several fronts. For example, since the dynamic feature of face motion is not considered during the SIDB construction, result face motion may have some jerky face motion. To overcome this problem and alleviate labor-intensive manual job with the SIEtool, we are actively continuing to adopt an extended kalman filter to track the face [2]. With face information from manual method, we make the initial position and region of face and then use it for the part of measurement in kalman filter framework to track face motion more efficiently. Using filtered face motion data from kalman filter, color histogram, and optical flow, we could find face-occluding region and obtain masking information automatically.

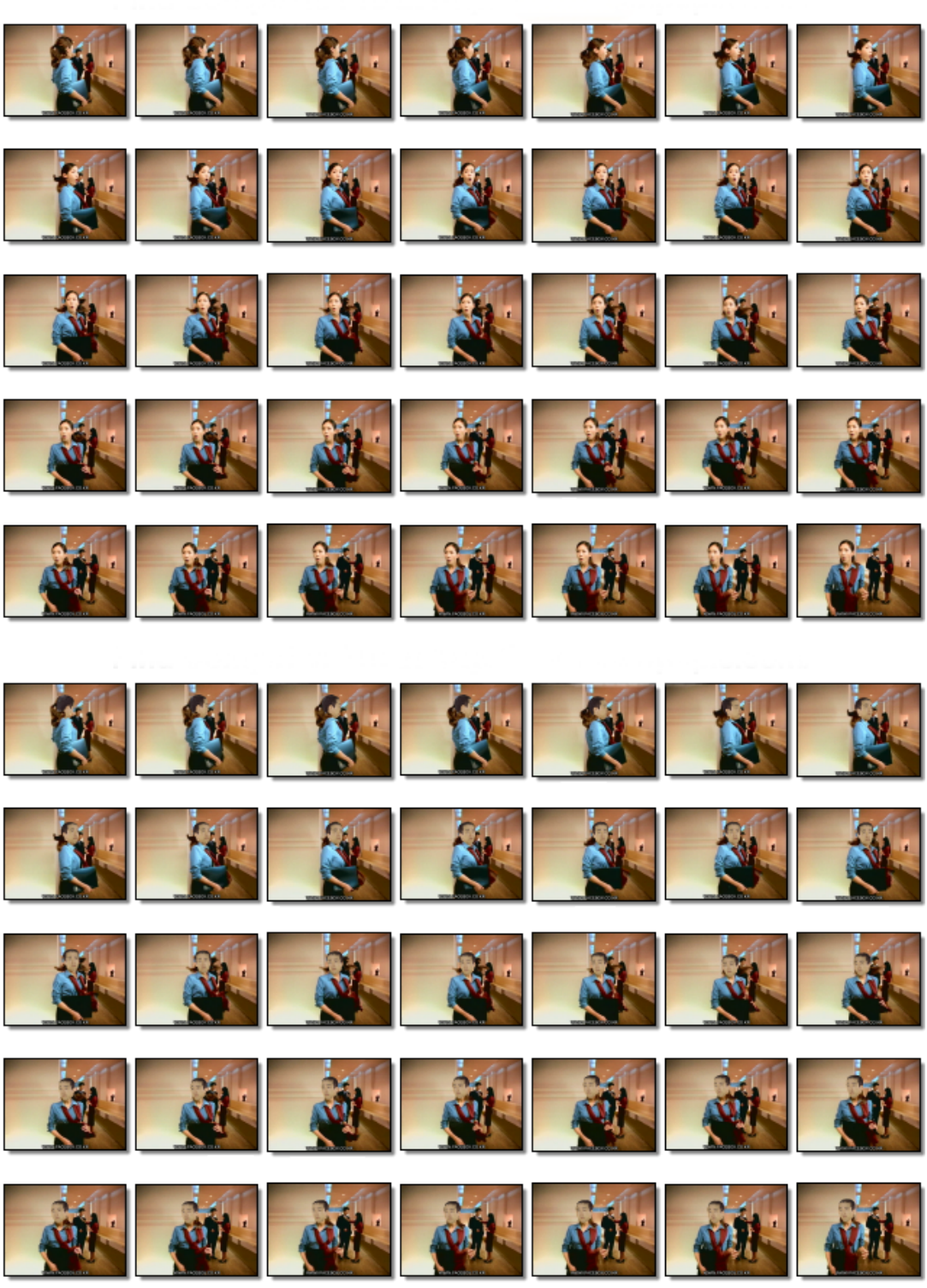

Figure 6. Superimposed result
(Upper 35 frames are original and lower 35 frames are superimposed image sequences)

5. Acknowledgement

This work was supported in part by Digitalion, Inc. and Virtual Reality Research Center.

6. References

[1] T. Akimoto, Y. Suenaga, and R.S. Wallace, “Automatic Creation of 3D Facial Models”, IEEE Computer Graphics and Application, Vol.13, No. 5, pp. 16-22, 1993.

[2] A. Azarbayejani, T. Starner, B. Horowits, and A. Pentland, “Visually controlled Graphics”, IEEE PAMI, Vol. 15, No. 6, pp. 602-605, 1993

[3] M.L. Cascia, S. Sclaroff, and V. Athitsos, “Fast, Reliable Head Tracking under Varying Illumination: An Approach Based on Registration of Texture-Mapped 3D Models”, IEEE PAMI, Vol. 22, No. 4, pp. 322-336, 2000.

[4] P.E. Devevec, “Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography”, Proc. SIGGRAPH’98, pp.189-198, 1998.

[5] W. Lee and N.M. Thalmann, “Head Modeling from Pictures and Morphing in 3D with Image Metamorphosis Based on Triangulation”, Proc. International Workshop, CAPTECH’98, pp. 254-267, 1998.

[6] F. Pighin, J. Hecker, D. Lischinski, R. Szleliski, and D. Salesin, “Synthesizing Realistic Facial Expression from Photographs”, Proc. SIGGRAPH’98, pp. 75-84”, 1998.

[7] H.A. Rowley, S. Baluja, and T. Kanade, “Neural Network-Based Face Detection”, IEEE PAMI, Vol. 20, No. 1, pp. 23-28, 1998.

[8] I. Sato, Y. Sato, and K. Ikeuchi, “Acquiring a Radiance Distribution to Superimpose Virtual Objects onto a Real Scene, IEEE Transaction on Visualization and Computer Graphics, Vol. 5, No. 1, Jan.-Mar., 1999, pp. 1-12