# CREATING 3-D VIRTUAL HEADS FROM VIDEO SEQUENCES : A RECURSIVE APPROACH BY COMBINING EKF AND DFFD

KyoungHo Choi, Ying Luo, Jenq-Neng Hwang
Information Processing Lab.
Department of Electrical Engineering
University of Washington, Box #352500
Seattle, WA 98195-2500, USA
email : {khchoi, luoying, hwang}@ee.washington.edu

## ABSTRACT

An automatic system for creating a virtual head that is compatible with MPEG-4 facial object specification is presented. Color classification and a valley detection filter are performed to find face and Facial Definition Points (FDPs) at the initialization stage. Extracted FDPs are tracked by normalized correlation and their trajectories are fed into an extended Kalman filter (EKF) to recover camera geometry, facial orientation, and depth of selected FDPs. Based on a recovered point-wise 3-D structure, Dirichlet Free-Form Deformations (DFFD) is applied to deform a generic 3-D model. Once a virtual head is created, the head can be used to track FDPs for large out-of-plane rotations and to update the head model continuously based on refined depth information. A complete texture map is created by mixing frontal and rotated faces based on the recovered face orientation.

## 1. INTRODUCTION

Virtual heads have an important role in multimedia applications such as collaborative virtual environments, virtual games and virtual conference systems [1][2][3]. Automatic creation of virtual heads from video sequences is a challenging task, because of its difficulty in establishing correspondences and an unknown facial structure and camera geometry.

A recent work of Lee and Thalmann showed orthogonal images can be used to build 3-D animation models [5]. A point-wise 3-D structure can be determined from corresponding point pairs on orthogonal images and can be used to reconstruct 3-D models by using Dirichlet Free-Form Deformations (DFFD). The only limitation of this method is that it requires manual intervention to locate corresponding point pairs in orthogonal images.

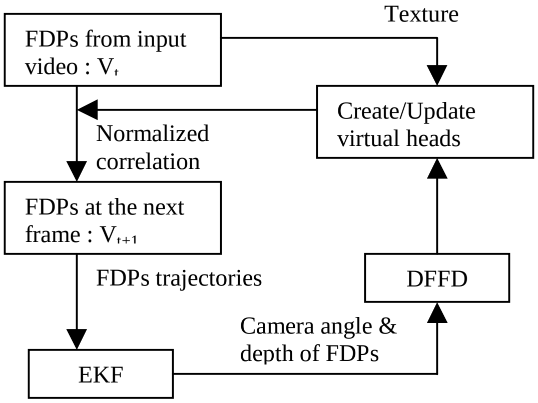

Texture
FDPs from input video : $V_t$
Create/Update virtual heads
Normalized correlation
FDPs at the next frame : $V_{t+1}$
FDPs trajectories
DFFD
Camera angle & depth of FDPs
EKF

Fig. 1. A block diagram for an automated system that create virtual heads

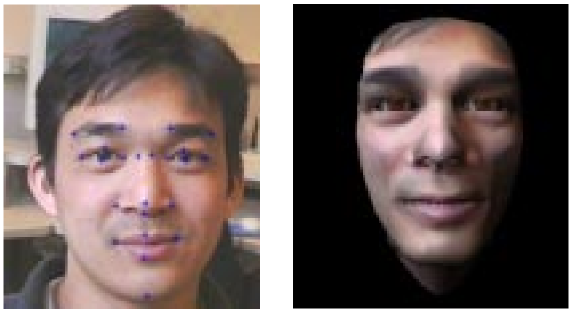
To overcome this limitation, an EKF-based approach can be applied. A research from Strom and Pentland showed an EKF-based structure from motion (SfM) algorithm can recover camera geometry as well as a point-wise 3-D structure by tracking selected feature points [6][7][8].

In this paper, an automatic system that can create a realistic virtual face is described. The basic idea is that we can combine a point-wise structure estimated from an EKF-based SfM algorithm with a DFFD method, which can be used to deform a 3-D model, to create virtual heads without any user intervention. Figure 1 shows a block diagram of our proposed system. The extracted FDPs, $V_{t,}$, at time $t$ can be tracked in the next frame using normalized correlation and their trajectories are fed into an EKF-based structure from motion algorithm to recover camera geometry and a point-wise structure. Based on the recovered point-wise structure, DFFD can be applied to deform a generic 3-D model. Selected FDPs are used for control points to deform other vertices of the 3-D model. After creating a 3-D model based on the recovered point-wise structure, the model can be used to guide a search area for 2-D correlation matching in the subsequent tracking stage. Constrained search area can guarantee more accurate tracking results and produce more precise depth information that allows us to refine the created virtual head after every iteration. The work described here is similar to that proposed in [8], in addition we combine their method with DFFD to create 3-D models automatically. To successfully combine these methods, we incorporate the comparison between feature points on the video sequence and projected feature points after rendering based on an estimated camera angle and depth information.

The organization of this paper is as follows. In Section 2, we explain a process of initialization and tracking of facial feature points. In Section 3, a procedure to combine point-wise 3-D structure with DFFD is explained. Experimental results are followed in Section 4. Finally, conclusion is given in Section 5.

## 2. INITIALIZING AND TRACKING

Our system is initialized by finding FDPs in a frontal face. To detect face and facial feature components such as eyes and mouth automatically, color classification can be used, followed by a valley detection filter and a profile scanning method. The valley detection filter is an appropriate choice for detecting facial features because facial feature regions such as eyes, eyebrows and the mouth have a deep valley for their luminance distribution. Details of the detection method can be found in [9].



(a) Feature Detection in a frontal image

(b) Initial 3D

Fig. 2. Initialization

After finding the FDPs of the frontal view, we begin with a generic 3-D model of a face. The generic 3-D model points are modified so that their horizontal and vertical positions correspond to the feature points found in the frontal view. The input face and the created 3-D model are quite different, because depth information from the generic model is not the same as the input face. Therefore the depth of each feature point is refined after each new frame is analyzed and more depth information is returned from the shape from motion algorithm.

In tracking the FDPs, we used an algorithm similar to the algorithm used in [7]. In our implementation however we used edge in addition to luminance information. A *7x7* pixel patch around each detected feature point was selected for use in locating the feature point in the subsequent frame. Normalized correlation between the selected patch and a patch from input video in a rectangular *22x11* search window was performed. The choice of non-symmetric search window was because natural head movements are generally involved side-to-side motion rather than up-and-down. In our normalized correlation, a patch from video at time *t+1* that maximizes the following equation was chosen.

$$\rho = \frac{1}{2}\left(\frac{\hat{a}_l \cdot \hat{b}_l}{\|\hat{a}_l\|\|\hat{b}_l\|}\right) + \frac{1}{2}\left(\frac{\hat{a}_e \cdot \hat{b}_e}{\|\hat{a}_e\|\|\hat{b}_e\|}\right),$$

where, $\hat{a}_l = a_l - \mu_l a_l$ and $\hat{b}_l = b_l - \mu_l b_l$. $a_l$ and $b_l$ are luminance vectors and $a_e$ and $b_e$ denote edge vectors. $\mu_l$ is the average of luminance value for a selected patch. $a_l$ and $a_e$ are vectors at time *t* and $b_l$ and $b_e$ are vectors at time *t+1*. $\rho$ is such that the more closely the two patches match one another, the closer $\rho$ is to 1.

## 3. COMBINING A POINT-WISE STRUCTURE WITH DFFD

It is important that we can reliably recover depth information for FDPs in order to generate accurate 3-D models. The EKF-based shape from motion approach of Pentland has shown its robustness in tracking facial features from video. Our system uses the same EKF-based approach to recover 3-D structure of selected FDPs.

After getting a point-wise 3-D structure, corresponding points in a 3-D model can be updated after measuring the distance between a feature vector from input video and a feature vector from a newly rendered 3-D model. We define the $L_2$ metric $d_2(\mathrm{V}_t, \mathrm{M}_t)$ to decide whether the new point-wise structure from the EKF is appropriate to update the 3-D model or not.

$$d_2(\mathrm{V}_\mathrm{t}, \mathrm{M}_\mathrm{t}) = \left( \sum_{i=1}^{N} (V_{ti} - M_{ti})^2 \right)^{1/2},$$

where $N$ is the number of feature points; $\mathbf{V_t}$ is a feature vector from video sequence at time $t$, which contains $(x,y)$ locations of selected feature points, and $\mathbf{M_t}$ denotes a feature vector that is projected and overlapped onto the input video after rendering the 3-D model by using the new point-wise 3-D structure and camera angle. The Distance $d_2(\mathrm{V}_\mathrm{t}, \mathrm{M}_\mathrm{t})$ is compared with $d_2(\mathrm{V}_\mathrm{t}, \mathrm{M}_{\mathrm{t-1}})$. If the distance $d_2(\mathrm{V}_\mathrm{t}, \mathrm{M}_\mathrm{t})$ is smaller than $d_2(\mathrm{V}_\mathrm{t}, \mathrm{M}_{\mathrm{t-1}})$, which means the newly rendered 3-D model is closer than the current 3-D model, we update the 3-D model. These updated points can be considered as control points to deform other vertices of the 3-D model. To reconstruct the 3-D model by using DFFD, 27 points surrounding the 3-D model are added automatically to adjust other vertices in the 3-D model. With this methodology a model for a human face can be constructed automatically given a video sequence of the face. By using the automatically constructed 3-D model, future attempts to locate the FDPs can be enhanced by predicting the location of the FDPs by using the updated 3-D face. The whole procedure of combining extended Kalman filter (EKF) and DFFD to build a virtual face is shown in Figure 3.

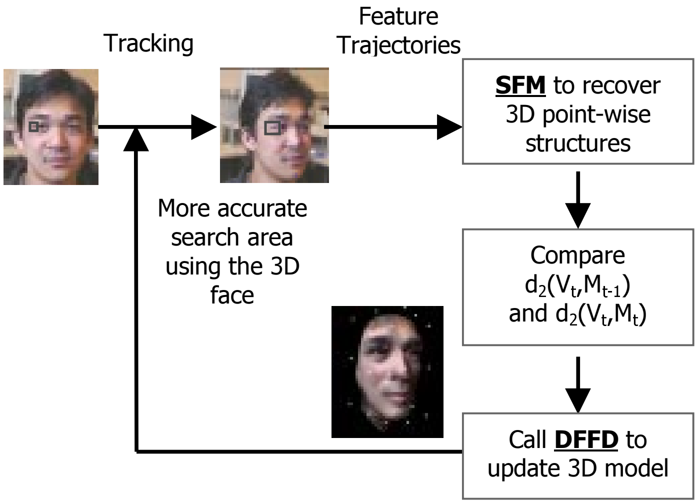

Tracking
Feature Trajectories
SFM to recover 3D point-wise structures
More accurate search area using the 3D face
Compare $d_2(V_t,M_{t-1})$ and $d_2(V_t,M_t)$
Call DFFD to update 3D model

Fig. 3. A block diagram for updating a virtual face in a recursive manner.

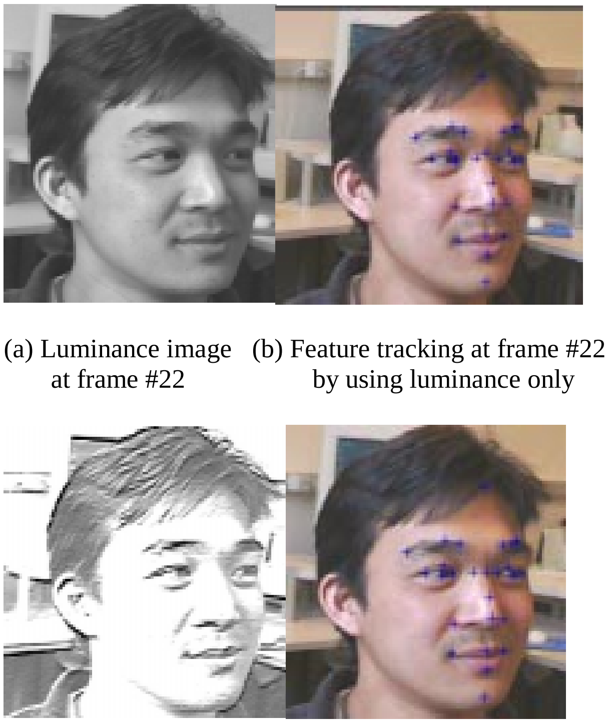
## 4. EXPERIMENTS

Our system was implemented in Pentium-III 600 MHz PC. A valley detection filter and a profile scanning method are used to detect feature points. We set initial feature locations manually in this simulation to extract more accurate texture information. Right after extracting facial feature points, an initial 3-D model was created based on the extracted feature points. This initial model has temporary depth information of the generic face model. Then, 2-D feature tracking was started and their trajectories were fed into EKF to estimate camera parameters and facial structure. Figure 4 (b) and (d) show feature tracking results of frame #22. By comparing tracked results of (b) and (d) around eyes, we can conclude that edge information can be used to improve tracking accuracy. Figure 5 shows two created models. Figure 5 (a) is a model created based on the depth from a generic model and (b) is the updated 3-D model based on the proposed automatic procedure. Because of limited feature points that can be used as control points for DFFD, the created 3-D model is not exactly the same as the input face but looks much more similar after updating depth information. Because of the inherited limitation of 2-D matching for large rotations, tracking is only successful within a rotation angle less than 30 degrees.

Our preliminary experimental result shows that the proposed system can create 3-D virtual faces automatically, except for the initialization to increase accuracy, without requiring the manual creation of the 3-D model though manually created models may look more realistic. To create more realistic 3-D models automatically, more feature points are needed in the feature tracking process.

## 5. CONCLUSIONS

An automatic system that can create virtual faces without user intervention is developed. This system uses a recursive approach to incrementally refine created 3-D facial models. An EKF-based structure from motion algorithm and DFFD method are combined to build 3-D virtual faces automatically. A distance measure that compares extracted feature points with projected feature points after rendering based on an estimated camera angle and a point-wise structure is defined to update the 3-D model. For future research, other interpolation methods such as B-spline and radial basis functions can be considered to deform 3-D models.



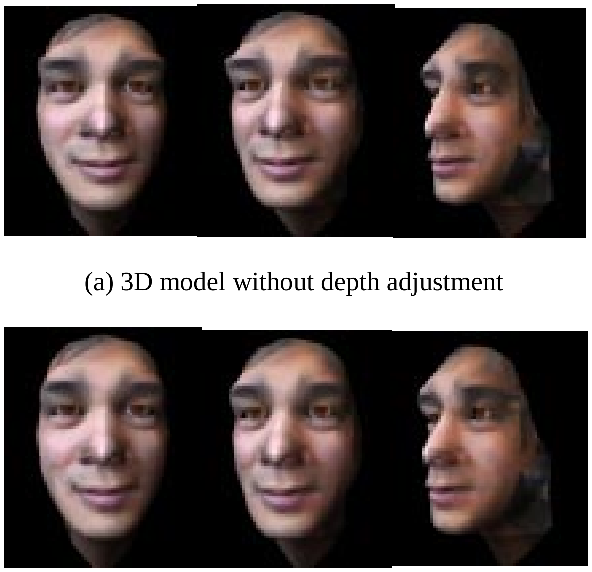
(a) Luminance image at frame #22

(b) Feature tracking at frame #22 by using luminance only

(c) Edge enhanced image at frame #22

(d) Feature tracking by combining luminance and edge

Fig. 4. An example of feature tracking



(a) 3D model without depth adjustment

(b) 3D model after updating depth information

Fig. 5. Comparison of created 3D models

## 6. REFERENCES

[1] Kiyokawa, K., Takemura, H., Yokoya, N., “SeamlessDesign: a face-to-face collaborative virtual/augmented environment for rapid prototyping of geometrically constrained 3-D objects,” *IEEE International Conference on Multimedia Computing and Systems* , Vol. 2, 1999, pp. 447-453.

[2] Yao-Jen Chang, Chih-Chung Chen, Jen-Chung Chou, Yung-Chang Chen, **“**Implementation of a virtual chat room for multimedia communications,*” 1999 IEEE 3rd Workshop on Multimedia Signal Processing*, 1999, pp.599-604.

[3] Yura, S., Usaka, T., Sakamura, K., **“** Video avatar: embedded video for collaborative virtual environment,” *IEEE International Conference on Multimedia Computing and Systems*, Vol. 2, 1999, pp. 433–438.

[4] Won-Sook Lee, N. Magnenat-Thalmann, "Fast Head Modeling for Animation," *Journal of Image and Vision Computing*, Volume 18, Number 4, 2000, pp.355-364.

[5] Won-Sook Lee, Marc Escher, Gael Sannier, Nadia Magnenat-Thalmann, "MPEG-4 Compatible Faces from Orthogonal Photos," *International Conference on Computer Animation*, 1999, pp.186-194.

[6] Ali Azarbayejani and Alex P. Pentland, “Recursive Estimation of Motion, Structure, and Focal Length,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, 1995, pp. 562-574.

[7] J Strom, T. Jebara, S. Basu, A. Pentland, “Real time tracking and modeling of faces: an EKF-based analysis by synthesis approach,” *Proceedings of IEEE International Workshop on Modelling People*, 1999 pp. 55 –61.

[8] T. Jebara and A. Pentland, “Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces,” *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 144-150.

[9] Ru-Shang Wang, Yao Wang, “Facial feature extraction and tracking in video sequences,” *IEEE International Workshop on Multimedia Signal Processing*, 1997, pp. 233 –238.