3D TRACKING OF FACIAL FEATURES FOR AUGMENTED REALITY APPLICATIONS

Vlado Kitanovski, Ebroul Izquierdo

Multimedia and Vision Research Group, Queen Mary, University of London, UK
{vlado.kitanovski, ebroul.izquierdo}@eecs.qmul.ac.uk

ABSTRACT

We present an algorithm for feature-based real-time 3D tracking of facial features and its application for visualization of virtual facial modifications. A non-linear Kalman-based estimator is used for 3D head pose calculation and accurate facial features localization. The 3D face model used is adapted to the particular user's face by utilizing active shape model for facial landmarks detection, followed by *z*-depth progressive refining. Virtual facial modifications are performed by user-driven 3D-aware 2D warping of the image sequence. The evaluation of our system shows that the tracker is robust to moderate head movements, occlusion and facial animation, which results in quite realistically-looking virtual facial modifications.

Index Terms— real-time facial features tracking, augmented reality, image warping

1. INTRODUCTION

Whenever we work on designing real-time application that includes facial analysis/synthesis, we have to model the face and track the facial features through the sequence. What are usually modelled are the face 3D shape and/or the face 2D/3D appearance. Depending on the concrete application's requirements, this model can be simple (e.g. low-vertex model) or complex (e.g., thousands of 3D vertices, or appearance model trained using huge number of images showing faces under different poses, facial expressions or illumination). Tracking of facial features refers to the process of obtaining the model's parameters for each frame in the sequence that best fit to the actual face appearance. This includes both the global model parameters (3D pose) and the local model parameters (facial features location, facial expression). The connection between the 3D tracking method and the face modelling approach is obvious and essential for every facial analysis based application.

In this paper, we present a real-time method for 3D tracking of facial features. The accent was put to finding a good compromise between accurate tracking, highly non-intrusive behavior, and real-time performance. We used the tracker in an interactive application for visualizing virtual face modifications. However, the tracker can be used for other real-time augmented reality applications as well.

Many authors have addressed this issue and proposed face trackers of various types. They differ in many aspects, like e.g.: how they model the face, the way they handle occlusions or changes in illumination, susceptibility to drifting, achievement of real-time performance, manual/automatic initialisation, or monocular/stereo/multi-view based techniques. In our work we focus on face tracking and modelling from monocular image sequences. In general, facial features trackers can be separated into two groups: *feature-based trackers*, where features like colour, edges, or templates are robustly tracked and matched to obtain the 3D face model parameters, and *appearance-based trackers* where entire facial appearance and shape is matched with the input image to find the best-fitting 3D model parameters.

Our review of feature-based trackers starts with the tracker proposed by Storm [1] who used Kalman filter to obtain the pose of generic face model. The model used was rigid and was manually initialized. Vacchetti et al. proposed real-time rigid-face tracker which included training phase to perfectly align the 3D model with the images for different poses [2]. Terissi and Gomez proposed a method for face tracking, in which features location obtained using pattern-matching techniques are refined using independent component analysis [3]. Chen and Davione used generic 3D face model that is trained using images showing faces in different poses and expressions [4]. For each frame, model's parameters are calculated by maximizing a similarity Gaussian-based function.

Appearance-based trackers were firstly introduced by Cootes [5]. Since then, a variety of trackers using this approach were proposed, including constrained appearance and shape models [6], online appearance models [7], dynamic active models [8] or 3D active appearance models [9]. What is common for these trackers is that the models are built using extensive training; the trackers are more robust to drift but may suffer from inaccurate tracking of facial expressions, and vulnerability to changing illumination, or to fair amount of out-of-plane head rotations.

In our approach, facial features are tracked using user-customized 3D face model which is generated online, in a

non-intrusive way as explained in section 2. Head movements are used to progressively refine the 3D face model for even more accurate performance. The details of the tracking process are explained in section 3. We implement this tracker in an augmented reality application where user-desired facial modifications are performed by means of 2D warping, as explained in section 4. Section 5 presents evaluation results, followed by concluding remarks and future research directions in section 6.

2. 3D FACE MODEL INITIALIZATION

The 3D face model used is initialized in the beginning of the tracking process, when the person places his face in a frontal position with respect to the camera. Unlike the uncomfortable manual landmarks/3D model initialization found in [1], [10-11] we use the Appearance Shape Model (ASM) trained on frontal faces [12] to automatically detect 68 facial landmarks. Additional 49 points are added using interpolation while preserving facial symmetry. All of the points are then assigned predefined z coordinates to initialize our 117-vertices user-customized 3D face model shown in Fig. 1. Relation between the 3D world and 2D image is established using the camera perspective model, with the coordinate origin set to be in the pinhole camera, the x-y plane parallel to the image plane and the z axis passing through the image centre. In this way, only one intrinsic parameter is needed – the focal distance f whose value is roughly approximated and then refined during tracking, as explained in the next section.

3. REAL-TIME FACIAL FEATURES TRACKING

In the variety of different approaches, we chose to use feature-based approach as we preferred to have more control in terms of tracker's deterministic behavior. Extended Kalman Filter (EKF) is fed with template matching results to estimate the state **b** of our face model - the pose (rotation and translation) and the facial animation parameters $\boldsymbol{\tau}_a$:

$$\mathbf{b} = [\mathbf{r}^T, \mathbf{t}^T, \boldsymbol{\tau}_a^T]^T \quad (1)$$

Similar approach can be found in [1], [13], where authors are estimating only the pose and the rigid 3D face model. Estimating the z-depth of the 3D face model, however, is not too critical for accurate tracking and shouldn't be performed all the time during tracking. A second EKF is used in the beginning phase only to refine the initialized z-depth of our model. After certain criterion is met, the second EKF is simply turned off. The EKF is initialized with the frontal face pose $\mathbf{r} = [0,\ 0,\ 0]^T$, the translations along x and y axis are calculated as the distance between model's and image centers, while the z-translation is roughly approximated. The animation parameters vector is initialized to zero – assuming normal face expression during tracker initialisation.

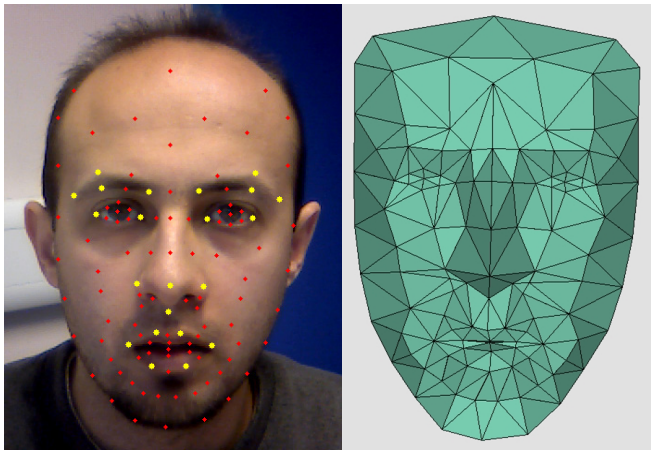

Figure 1: Frontal face with facial landmarks (left); Corresponding 3D model (right).

For each frame, the 3D model is rendered first using the estimated parameters **b** and the first-frame texture, and then square templates are extracted around the N=22 tracked points (shown in yellow in Fig. 1). In order to improve performance under changing lighting conditions or facial animation, these templates are alpha-blended with corresponding templates extracted from the previous frame to produce the final templates. The latter ones are used for matching within a search window in the next frame. Zero-mean normalized cross-correlation is used to calculate the matching score. The estimation of point's position in the new frame is performed using a second-order linear model which takes into account the current spatial speed and the acceleration of the tracked point. Considering that the facial movements at uniform high frame rate are smooth, this approach is quite accurate and requires considerably less calculations than using the EKF for the same purpose (which would require several matrix multiplications for predicting the state **b** and projecting the tracked points afterwards). The measurement noise covariance matrix, **R**, is used to manipulate the EKF on which tracked points should be considered more when estimating the state for the incoming frame. "Greater" covariance value makes the EKF to "less" consider that particular measurement when estimating the state. This matrix is diagonal and for each tracked point i, it is updated with the following value:

$$R(i,i) = \frac{1 + Cd}{M_c} \cdot W(\mathbf{r}, i) \quad (2)$$

In the last equation, C is a constant; d is the distance between the estimated centre and the actual best match; M_c is zero-mean normalized cross correlation between the matching templates, while $W(\mathbf{r},i)$ are experimentally obtained weighting factors used to suppress points that, according to the face pose **r**, might be occluded. For example, as the head is turning to the left side (according to the r_y value), tracked points i on the face left half are being

assigned larger weighting factors; It means that these points are more likely to be occluded which leads to possibly incorrect measurements, so the value of the "measurement noise" is artificially increased for those points. This rough estimation of the measurement error makes the EKF "aware" about potential occlusions under head rotation, or eventual bad matches. The relation matrix **H**, that relates the 2D tracked points with the state **b**, is linearized around the current state parameters using the camera model. The other covariance matrices are initialized to the identity matrix.

In the beginning stage of the tracking, a second EKF is used to refine the model's z-depth to fit better the tracked face. Estimating the z coordinate for all vertices using EKF would be impossible in real-time, so only those k vertices that correspond to the tracked points around the facial features on the left half of the face are estimated (we assume that the face is symmetric in the z direction). The z coordinates of all other vertices are corrected accordingly using linear interpolation. The focal distance f is also estimated, so the state vector is $\mathbf{s} = [z_1, z_2, \ldots z_k, f]^T$. As z-depth and focal distance don't vary in time, they are estimated until the total relative change $\Delta\mathbf{s}$ of the state vector in the last consecutive F frames falls below experimentally obtained threshold – when the second EKF is turned off and the tracker continues to use the corrected 3D vertices and focal distance.

4. REAL-TIME FACE MODIFICATIONS

We use the presented tracker to build an augmented reality application, where the user can experiment with the appearance of various virtual shape modifications of his/her face. In order to get very high realistic impression, 2D warping of the original camera video is used. The warping is performed according to the user's desire for slight modification of his/her lips, nose or eyebrows. Each of the facial modifications is controlled by a separate pair modification unit–modification control vector, and they are performed on the static 3D face model together with the facial animation. After the modified face model is obtained it is then back-projected on the image plane to form 2D mesh. This back-projected mesh, together with the one obtained without applying any modification define a triangular mesh warp that establishes affine mapping from each triangle in the source (unmodified) mesh to the corresponding triangle in the destination mesh. Apparently, this approach cannot achieve realistic face rendering under arbitrary rotations of the head. However, the warping used has several advantages, e.g. GPU exploitation and spatially-controlled warping – triangles are independently warped without being influenced of other non-neighboring triangles [14]. The main disadvantage – C^0 continuity at the edges, is not a serious problem for our application as the facial modification performed is usually minimal, and always within allowed range.

5. EVALUATION RESULTS

In this section we present the evaluation results of our face tracker. Implemented using non-optimized C/C++ code, it runs at 25 fps on Pentium D 3.2 GHz PC with a webcam and onboard graphic card, and with the CPU not being used 100% during tracking. Intel's OpenCV library is used for handling the webcam input, OpenGL library for accessing the functionalities of the GPU and the ASM Library [12] for automatic landmarks detection in frontal faces.

Examples images showing the tracked points for different head positions are shown in Fig. 2. Figure 3 shows examples of 3D face model drawn on top of the tracked face. The tracker is robust to head rotations within the following ranges: ±55°, ±35° and ±30° for rotations around x, y and z axis, respectively. However, if the head is moved too quickly the track may be lost so the tracker will have to be re-initialised in frontal pose. Figure 4 shows examples of performed nose and lips modification. It can be seen that the new rendered faces don't suffer visible distortions as long as the tracking is accurate and the warping is within certain limits that correspond to feasible facial alterations.

6. CONCLUSION

In this paper, we have presented our real-time facial features tracker. The 3D face model used is customized to each tracked face by employing active shape models for landmark detection in frontal poses, and z-depth progressive refining under head motion in the beginning phase of tracking. Face pose and local facial actions are estimated using non-linear Kalman filter, which is made aware of the global head pose to combat occlusion during head rotation.

As the tracked facial points are stable and accurate, we use the tracker in an augmented reality application which allows users to make attentive and precise modifications of their face. Evaluation results show that the tracker is robust to fair amount of head rotations, which can be also verified by the realism of the visualized virtual face alterations. Besides trading tracker robustness for complexity, other space for possible further improvements is development of method for automatic estimation of the 3D initial head pose, so that tracker initialization can be performed accurately on arbitrary nearly-frontal face poses.

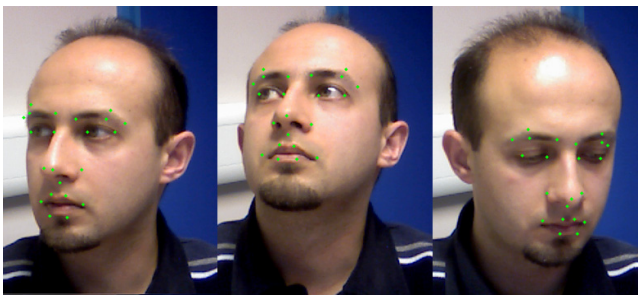

Figure 2: Tracked points for different head poses.

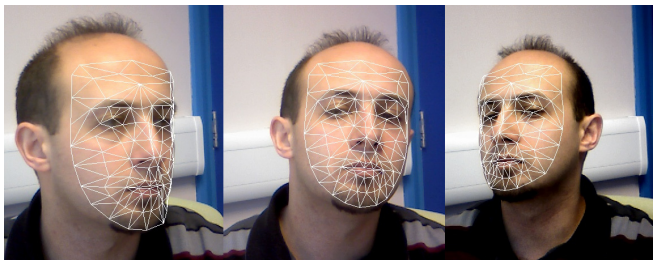

Figure 3: 3D mesh drawn on top of the tracked face.

7. ACKNOWLEDGEMENTS

This work is supported by the EU project 3DLife (Grant No. 247688).

8. REFERENCES

[1] J. Storm, "Model-based Real-Time Head Tracking", *EURASIP Journal on Appl. Signal Proc.*No. 1, January 2002.

[2] L. Vacchetti, V. Lepetit, and P. Fua. "Stable Real-Time 3D Tracking Using Online and Offline Information". *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 10 pp.1385-1391 October 2004.

[3] L. Terissi, J. G´omez, "Facial motion tracking and animation: An ICA-based approach" *in Proc. EUSIPCO 2007*; pp292–296; Poznan, Poland, 2007.

[4] Y. Chen, F. Davoine, "Simultaneous tracking of rigid head motion and non-rigid facial animation by analyzing local features statistically", *in Proc. BMVC* 2006.

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[6] D. Cristinacce, T. F. Cootes, "Feature Detection and Tracking with Constrained Local Models" *in Proc. BMVA 2006*, pp. 929-938, Edinburgh, September, 2006.

[7] F. Dornaika and F. Davoine, "On appearance based face and facial action tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 9, pp. 1107–1124, Sep. 2006.

[8] D. Zhou, P. Horain, "Robust 3D Face Tracking on Unknown Users with Dynamical Active Models", *in Proc. 15th Multimedia Modelling Conference*, pp. 74-84, Berlin 2008.

[9] C. Chen, C. Wang, "3D Active Appearance Model for Aligning Faces in 2D Images", *in Proc. Intelligent Robots and Systems*, Sept. 2008.

[10] M. Chaumont and B. Beaumesnil, "Robust and Real-Time 3D-Face Model Extraction" *in proc. ICIP 2005*, Genova, 2005.

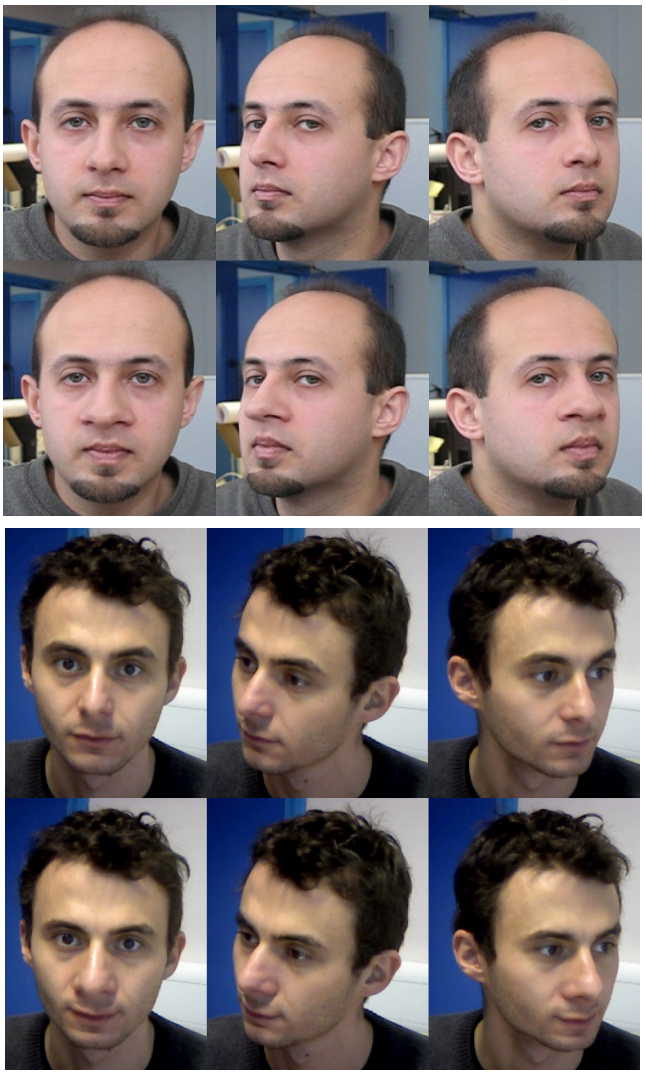

Figure 4: Rendered virtual facial modifications. Top rows: Original faces; Bottom rows: Modified faces.

[11] F. Dornaika and B. Raducanu, "Three-Dimensional Face Pose Detection and Tracking Using Monocular Videos: Tool and Application" *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 39, No. 4, August 2009.

[12] Y. Wei, "Research on Facial Expression Recognition and Synthesis", *Master thesis*, Nanjing University, February 2009.

[13] J. Ahlberg and F. Dornaika, "Parametric Face Modeling and Tracking" *chap in Handbook of Face Recognition*, Springer, 2005.

[14] S. Melacci, L. Sarti, M. Maggini, and M. Gori, "A Template-based Approach to Automatic Face Enhancement" *Journal of Pattern Analysis & Applications*, vol. 13, Issue 3, August 2010.