# Real Time 3D Face Pose Tracking From an Uncalibrated Camera

Zhiwei Zhu
Department of ECSE
Rensselaer Polytechnic Institute, Troy, NY
zhuz@rpi.edu

Qiang Ji
Department of ECSE
Rensselaer Polytechnic Institute, Troy, NY
qji@ecse.rpi.edu

## 1 Abstract

We propose a new near-real time technique for 3D face pose tracking from a monocular image sequence obtained from an uncalibrated camera. The basic idea behind our approach is that instead of treating 2D face detection and 3D face pose estimation separately, we perform simultaneous 2D face detection and 3D face pose tracking. Specifically, 3D face pose at a time instant is constrained by the face dynamics using Kalman Filtering and by the face appearance in the image. The use of Kalman Filtering limits possible 3D face poses to a small range while the best matching between the actual face image and the projected face image allows to pinpoint the exact 3D face pose. Face matching is formulated as an optimization problem so that the exact face location and 3D face pose can be estimated efficiently. Another major feature of our approach lies in the use of active IR illumination, which allows to robustly detect eyes. The detected eyes can in turn constrain the face in the image and regularize the 3D face pose, therefore the tracking drift issue can be avoided and the processing can speedup. Finally, the face model is dynamically updated to account for variations in face appearances caused by face pose, face expression, illumination and the combination of them.

Compared with the existing 3D face pose tracking techniques, our technique has the following benefits. First, our technique can track face and face pose simultaneously in real time, which has been implemented as a real time working system. Second, only one uncalibrated camera is needed for our technique, which will make our system very easy to set up. Third, our technique can handle facial expression change, face occlusion and illumination change, which will make our system work under real life conditions.

## 2 Introduction

Face pose tracking is very important in vision based applications such as HCI, face recognition, and virtual reality. Many techniques have been proposed for face pose estimation. Basically, face pose estimation techniques can be classified into two main categories: appearance-based approaches [1, 2, 3, 4, 5] and model-based approaches [6, 7, 8, 9]. Appearance-based approaches attempt to use holistic facial appearance, where face is treated as a two-dimensional pattern of intensity variations. They assume that there exists a mapping relationship between 3D face pose and certain properties of the facial image, which is constructed based on a large number of training images. The appearance-based approaches usually only provides a sparse set of face poses rather than continuous-valued face poses. In summary, the main benefit of these approaches is their simplicity, but there are several significant hurdles. First, despite efforts such as the use of wavelet transform to minimize the effect of some factors, the mapping function is still a function of many other factors including face poses, such as illuminations, camera parameters and etc. Second, these techniques often require a face detector first. They often rely on others' techniques to detect faces or crop the face regions by hand. Finally, face pose is estimated on the detected face region. In these techniques, face detection and face pose estimation is done separately, ignoring any dependence between them.

Model-based (or features-based) approaches usually assume a 3D model of the face and recover the face pose based on the assumed model. First, a set of $2D-3D$ feature correspondences are established. Then the face pose is estimated by using the conventional pose estimation techniques [7, 8, 6]. Ji et al. [9] propose a 3D face pose estimation by modelling the face as an ellipse and by using anthropometric statistics of the face. Face poses are recovered based on the distortion of the face images. So, for most of these approaches, pose estimation is done after face feature tracking. Though simple to implement, robust and accurate facial feature tracking is often a significant challenge due to face occlusion, face movement, facial expression change and illumination change.

We can conclude that most existing methods for face pose estimation follow the strategy of face detection in the image and face pose estimation from the detected face images. The main problem with all these approaches is that face detection and face pose estimation are carried

0-7695-2158-4/04 $20.00 (C) 2004 IEEE

out independently. There is no input from each other. But in reality, these two steps are very interrelated, and because the face location in the image is caused by face pose in 3D, they must be consistent with each other. Therefore, we propose to take full advantage of the interdependent relationship between face image and 3D face pose and perform face detection and face pose estimation simultaneously.

Several similar techniques [10, 11, 12, 13, 14] have been proposed recently to track the face in 3D space. Sumit Basu et al [13] presented a 3D head tracker by modelling the head as an ellipsoid. Experiments show that the algorithm is stable to extract 3D head information accurately, but it is sensitive to the motion being observed in the scene due to the use of optical flow regularization. Recently, La Cascia et al. [15] present a 3D head tracking system that is robust to varying illuminations. In their technique, the head is modeled as a texture mapped cylinder and head tracking is formulated as registering the face image with the cylinder's texture mapped images under different face poses. In other words, tracking is based on the minimization of the sum of squared differences between the incoming texture and a reference texture. Therefore, it is sensitive to the changes in facial expressions, illuminations and self-occlusion. Although an illumination corrector is used to avoid the illumination effect under the assumption of a Lambertian surface in the absence of self-shadowing , it can not work very well under the indoor environment because the face surface is not truly Lambertian nor is there an absence of self-shadowing. Alternatively, in this paper, we are going to introduce a face model updating strategy to account for these changes, which can improve the accuracy of the SSD based tracker dramatically under these changes. Also, because human is not truly cylindrical, the accuracy of their system, as explored by Brown [16], degenerates under some conditions, such as large rotations around the vertical axis and large frame to frame pose changes. Even, La Cascia's approach is unable to distinguish rotations around the horizontal axis and vertical translations or similar rotations around the vertical axis and the horizontal translations. In order to improve the accuracy of the head tracker, Brown [16] presents methods to overcome these problems. First, adaptive motion templates are used to handle the variable frame-to-frame motion changes. Second, 2D positional information of the neckline is utilized to constrain the 3D positional estimates. Third, additional motion templates are used to compensate for the large head rotations. But there are still several issues. Neckline is difficult to track robustly due to possible occlusions, and the strategy to use additional templates is not convincing. Even, the skin tone based face detector will fail under some situations such as poor illuminations. More recently, Thomas Vetter et al. [17] proposed an algorithm to fit the 2D face images with 3D Morphable Models such that the face pose can be estimated. Although the face pose can be estimated accurately, the average processing time for each frame is around 30 seconds, which is too slow for a real time face pose estimation system.

In this paper, we describe a new technique to perform the 2D face tracking and 3D pose estimation synchronously. In our method, 3D face pose is tracked using Kalman Filtering. The initial estimated 3D pose is then used to guide face tracking in the image, which is subsequently used to refine the 3D face pose estimation. Face detection and face pose estimation work together and benefit from each other. Weak perspective projection model is assumed so that face can be approximated as a planar object with facial features, such as eyes, nose and mouth, located symmetrically on the plane. Figure 1 summarizes our approach.

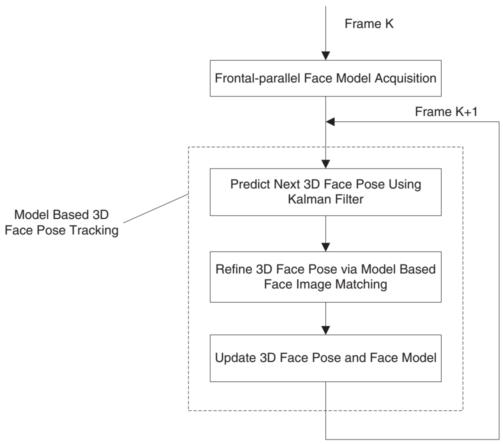

Frame K
Frontal-parallel Face Model Acquisition
Frame K+1
Model Based 3D
Face Pose Tracking
Predict Next 3D Face Pose Using
Kalman Filter
Refine 3D Face Pose via Model Based
Face Image Matching
Update 3D Face Pose and Face Model

**Figure 1.** The Flowchart of Face Pose Tracking

First, we automatically detect a frontal-parallel face view image based on the detected eyes and some simple anthropometric statistics. The detected face region is used as the initial 3D planar face model. The 3D face pose is then tracked starting from the frontal-parallel face pose. During tracking, the 3D face model is updated dynamically to account for the face appearance changes introduced by the illumination, facial expression, face pose, or the combinations of them, and the face detection and face pose estimation are synchronized and kept consistent with each other. Also, the use of the 2D eye location is used to constrain the face location in the image, which can avoid the tracking drift issue and speedup the processing. If the tracker fails, it will recover automatically by the robust recovery technique.

Because of these improvements, our technique can successfully track face pose under large out-of-plane face rotations. It is more tolerable to illumination and facial expression changes. Furthermore, face pose can be robustly tracked even under rapid head movements. Finally, automatic face detection is performed to initialize 3D face model.

## 3 Weak Perspective Model for Face Pose Estimation

We employ an object coordinate system affixed to user's face, with face normal (face pose) being the $Z$ axis of the object frame. Without loss of generality, face is assumed to be planar. Let $X = (x, y, 0)^t$ be the coordinate of a 3D point on the face relative to the object coordinate frame, and $p = (c, r)^t$ the coordinate of the corresponding projection image point in the row-column frame.

For each pixel $(c_1, r_1)$ in a given image frame $I_1$ and the corresponding image point $(c_2, r_2)$ in another image $I_2$, using basic projection equation of weak perspective camera model for planar 3D object points [18] yields

$$\begin{pmatrix} c_2 - c_{c2} \\ r_2 - r_{c2} \end{pmatrix} = M_2 * {M_1}^{-1} \begin{pmatrix} c_1 - c_{c1} \\ r_1 - r_{c1} \end{pmatrix} \tag{1}$$

where $M_1$ and $M_2$ are the projection matrices for image $I_1$ and $I_2$ respectively, and $(c_{c1}, r_{c1})$ and $(c_{c2}, r_{c2})$ are the projection points of the same reference point $(x_c, y_c, z_c)$ in the image $I_1$ and image $I_2$. Equation 1 is the fundamental weak perspective homographic projection equation that relates image projections of the same 3D points in two images with different face poses. The homographic matrix $P = M_2 * {M_1}^{-1}$ characterizes the relative orientation between the two face poses.

From equation 1, if we have one face image and its corresponding pose matrix, we can theoretically reconstruct all the other different face view images, which will have different pose matrices.

The face pose can be characterized by a rotation matrix $R$ resulted from successive Euler rotations of the camera frame around its $X$ axis by $\omega$, its once rotated $Y$ axis by $\phi$, and its twice rotated $Z$ axis by $\kappa$ [18]. Elements of the projection matrix is related with these three angles and a scale factor $\lambda$, representing the distance from face to camera. The 3D pose of a face can therefore be characterized by the three Euler angles and the scaler $\lambda$. While the three angles determine face orientation, $\lambda$ determines the distance from face to camera. Face pose estimation can be expressed as determination of these four parameters $(\omega, \phi, \kappa, \lambda)$.

## 4 Simultaneous 3D Face Pose Determination and 2D Face Tracking

We propose a novel technique to track 3D face pose and 2D face in the image synchronously as follows.

### 4.1 Automatic 3D Face Model and Pose Initialization

In our algorithm, we should have a fronto-parallel face to represent the initial face model. This initialization is automatically accomplished by using the eye tracking technique we have developed. Specifically, the subject starts in fronto-parallel face pose position with the face facing directly to the camera as shown in Figure 2. The eye tracking technique is then activated to detect eyes. After detecting the eyes, the first step is to compute the distance $d_{eyes}$ between two eyes. Then, the distance between the detected eyes, eyes locations and the anthropometric proportions are used to estimate the scope and the location of the face in the image automatically. Experiments show that our face detection method works well for all the faces we tested. Example of the detected frontal face region is shown in Figure 2. Once the face region is decided, we will treat it as our initial face model, whose pose parameters are used as initial face pose parameters.

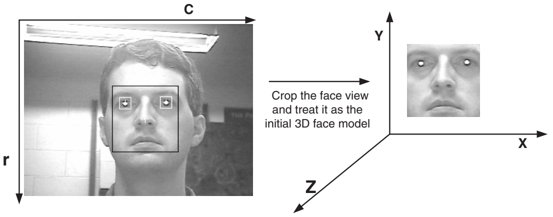


**Figure 2.** The initial face model

The face pose parameters $(\omega, \phi, \kappa, \lambda)$ for the initial face model are $(0°, 0°, 0°, 1)$, where $\lambda$, without loss of generality, is normalized to 1. The corresponding projection matrix $M$ is:

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2}$$

We use the center of the 3D face model as the reference point, which is represented as $(x_c, y_c, z_c)$. Furthermore, we assume its corresponding projection image point is $(c_c, r_c)$ in the row-column image plane.

Compared with the existing frontal face detection methods, ours takes full advantage of the detected eyes to guide the detection of the frontal face, and it is simple, robust and automatic. In section 5, we will also demonstrate the tolerance of our face initialization to slight deviations from the frontal-parallel face pose and to perturbations of initial positions of the face.

### 4.2 Face Pose Tracking Algorithm

#### 4.2.1 Kalman Filter with Eye Constraints

Given the initial face image and its pose in the first frame, the task of finding the face location and the face pose in subsequent frames can be implemented as simultaneous 3D face pose tracking and face detection.

Given the 2D face model obtained from the initialization, the current face pose parameters $X_t = (\omega_t, \phi_t, \kappa_t, \lambda_t)$ and the current face image location $(x_t, y_t)$, the Kalman Filtering based face pose tracking consists of the following steps.

1. Combined Prediction

   Let the state vector at time $t$ be represented as $X_t = (\omega_t \ \phi_t \ \kappa_t \ \lambda_t \ x_t \ y_t)^t$. According to the theory of Kalman Filtering [19], the system can therefore be modelled as

   $$X_{t+1} = \Phi X_t + \mathbf{w_t} \tag{3}$$

   where $\Phi$ is the state transition matrix, and $\mathbf{w_t}$ system perturbation.

   Given the system model, $X^-_{t+1}$, the state vector at $t+1$, can be predicted by

   $$X^-{}_{t+1} = \Phi X_t + \mathbf{w_t} \tag{4}$$

   along with its covariance matrix $\Sigma^-_{t+1}$ to characterize its uncertainty.

   The prediction based on Kalman Filtering assumes smooth face movement. The prediction will be off significantly if head undergoes a sudden rapid movement. In dealing with the problem, we propose to approximate the face movement with eyes movement since eyes can be reliably detected in each frame. Let the predicted face pose vector at $t+1$ based on eyes motion be $X^p_{t+1}$. Then the final predicted face pose should be based on combining the one from Kalman with the one from eyes, i.e.,

   $$X^*_{t+1} = X^-_{t+1} + \Sigma^{-1}_{t+1}(X^-_{t+1} - X^p_{t+1}) \tag{5}$$

   The simultaneous use of Kalman Filtering and eyes motion allows to perform accurate face pose prediction even under significant and rapid head movements. We can then derive a new covariance matrix $\Sigma^*_{t+1}$ for $X^*_{t+1}$ using the above equation to characterize its uncertainty.

2. Detection

   Given the predicted state vector $X^*_{t+1}$ at time $t+1$ and the prediction uncertainty $\Sigma^*_{t+1}$, we can perform a face pose estimation via face detection verification. Specifically, $X^*_{t+1}$ and $\Sigma^*_{t+1}$ form a local pose space at time t+1. The pose search in this local pose space will lead to the measured pose $z_{t+1}$. The search can be formulated as a minimization problem to be detailed in section 4.2.2. Let the measurement model in the form needed by the Kalman filtering be

   $$\mathbf{z}_{t+1} = HX_{t+1} + m_{t+1} \tag{6}$$

   where $m_{t+1}$ represents measurement uncertainty, normally distributed as $m_{t+1} \sim N(0, S)$, where $S$ is measurement error covariance matrix. The matrix $H$ relates the state $X_{t+1}$ to the measurement $z_{t+1}$.

3. Face Pose Updating

   Given the predicted face pose $X^*_{t+1}$, its covariance matrix $\Sigma^*_{t+1}$ and the measured face pose $z_{t+1}$, face pose updating can be performed to derive the final pose $X_{t+1}$

   $$X_{t+1} = X^*_{t+1} + K_{t+1}(z_{t+1} - HX^*_{t+1}) \tag{7}$$

   where $K_{t+1}$ is the Kalman gain matrix.

4. Update Face Model

   If current face pose aspect is significantly different from the face model aspect, the face model should be updated. The face model for frame $t+2$ is updated based on successful face pose estimation for frame $t+1$. Our study shows that it is important to update the face model dynamically to account for the significant aspect changes under different face orientations or facial expressions. Details on updating strategy will be discussed in Section 4.2.4.

### 4.2.2 Face Detection and Pose Estimation via Matching Optimization

The combined prediction from Kalman Filtering provides the predicted face position, 3D face pose, and the associated uncertainty $\Sigma^*_{t+1}$ for the next frame $t+1$. $\Sigma^*_{t+1}$ can be used to limit the search area for the face location and face pose at time $t+1$. Face detection and face pose estimation are to search for a face position and 3D face pose within the scope determined by $\Sigma^*_{t+1}$ such that the detected face view image can best match the projected face view image under the given face pose.

Mathematically, this is formulated as follows. Find the state vector $z_{t+1}$ within the scope determined by $\Sigma^*_{t+1}$ from $X^*_{t+1}$ such that the detected face image best matches the projected face image. We formulate the matching criterion as Sum of Squared Difference errors over all the image pixels within the region of interest:

$$E_{matching} = \sum_{i=1}^{N} (I_p(i) - I_c(i))^2 \tag{8}$$

where, $I_p(i)$ is the pixel value of $i$th pixel in the reconstructed face view image $I_p$, $I_c(i)$ the pixel value of the face image in the current image frame, and $N$ the total pixel number of the reconstructed face view image. $I_p(i)$ is projected from the reference image $I_r(c, r)$ via a mapping function $f$

$$I_p = f(I_r, \alpha)$$

where $\alpha = (\omega, \phi, \kappa, \lambda, x, y)$, which consists of the face pose parameters. We need to find the locally optimal face pose parameter set ${\alpha^*}_n$, which results in the projected face image that best matches the face in the current image frame

$$ {\alpha^*}_n = \arg\min_{\alpha}\ E_{matching} $$

$E_{matching}$ is minimized to solve for the 6 pose parameters, where $x^*_{t+1}$ serves as the initial value of $\alpha$.

#### 4.2.3 Regularization of Minimization Criteria by Eyes Positions

Since different sets of pose parameters may produce the same image, yielding the same $E_{matching}$, the minimization procedure may converge to a wrong place if it is left unconstrained. This problem is further exacerbated by the susceptibility of SSD to drifting. To overcome this, a penalty term is imposed to each SSD error corresponding to each set of pose parameters. Since we can accurately and independently detect the eyes position, the detected eyes positions can be used to constrain the 3D face pose and the 2D face image.

For each pair of the projected eyes in the projected face image and the detected eyes in the detected face image, the distance $E_{eyes}$ between the detected eyes and the projected eyes can be expressed as

$$ E_{eyes} = E_{Left} + E_{Right} \tag{9} $$

where $E_{Left}$ is the Euclidean distance between the detected left eye and the projected left eye, and $E_{Right}$ is the Euclidean distance between the detected right eye and the projected right eye. The correct face pose and face position should simultaneously minimize the results from equations 8 and 9. Therefore, the criteria for the image matching minimization can be expressed as the sum of both terms

$$ E = \beta E_{matching} + (1-\beta)E_{eyes} \tag{10} $$

where $\beta$ is a scale factor determining the relative importance of two terms. It is determined empirically.

#### 4.2.4 Dynamic Face Model Updating

In practice, during tracking, the human face usually undergoes different kinds of variations, most of which come from the pose, expression, illumination and the combination of them. In order for face templates to maintain adequate tracking performance while tracking face with time-varying appearance, it is necessary to dynamically adapt the face model to keep them consistent with the changing appearance of the face.

Specifically, our strategy for updating the face template is to include in each new face template with a portion of the initial template, which is assumed to have been chosen correctly, and thus contain the desired target. For example, for the $k^{th}$ frame, once the best match is found, with the matching confidence measurement above a certain threshold, the new face template $I_r(k+1)$ for frame $k+1$ is updated to be the combination of the initial face template $I_r(0)$ and the face model $I_g(k)$ generated from the image region $I_c(k)$ in the new frame that best matches the current template as follows:

$$ I_r(k+1) = \rho I_r(0) + (1-\rho)I_g(k) $$

where $I_g(k) = g(I_c(k), \alpha)$, and $\alpha$ is the detected face pose parameters related with frame $k$ and function $g$ is the inverse projection function of equation 9.

Furthermore, the constant $\rho \in [0,1]$ determines the contribution of the initial template to the new template. $\rho = 0$ is the case of fully updated templates and $\rho = 1$ gives standard templates.

By the dynamic face model updating and physical eye locations constraints, our tracker greatly improve the tracking accuracy.

### 4.3 Tracking Failure Detection and Recovery

Human eyes are the most significant features of the face, therefore, the location of the human's face can be determined once two eyes are detected [20]. Further, eyes from the same face are likely to blink at the same time and with the same frequency. They also move rigidly with the head, and the expected size of the face can be estimated from the distance between two eyes and the location of the face can be estimated from the eyes positions. Therefore, the eye's motion and position information can be combined easily to tell whether the face pose tracker fails or not. Once the face's motion and the eye's motion or the face's position and the eye's position are inconsistent with each other, we can conclude that the tracker fails.

Once the face pose tracker fails, a backup technique is proposed to recover automatically. First, based on the eye locations in the image, the new face center is estimated according to some anthropometric proportions. Second, 22 fiducial facial feature points are easily detected via the eye constrained facial feature tracker proposed in [21]. Based on the detected facial features, a set of rough face pose parameters can be estimated via the feature-based technique [6]. Finally, based on these new roughly estimated face pose parameters, they are further refined by a local search over the face pose parameter space to minimize the matching criteria 10.

Based on the above techniques, our face pose tracker can automatically detect the tracking failures and reinitialize the tracker when the tracker is lost.

## 5 Experimental Results

A series of experiments involving real image sequences are conducted to characterize the performance of our face pose estimation technique. First, the accuracy of the estimated face pose is analyzed. Then, we study the sensitivity of our algorithm to perturbations with initial face pose and placement. Further we demonstrate the effectiveness of face model updating for accurate face pose tracking.

### 5.1 Accuracy of the Face Pose Tracker

In order to measure the accuracy of the estimated face pose, several image sequences were collected. The ground truth for these sequences was simultaneously collected via "Flock of Birds" magnetic tracker.

In Figure 3, two estimated face pose angles, Pitch and Yaw, are shown together with the ground truth. The estimation error for Yaw is 2.9260 degrees and the estimation error for Pitch is 3.6174 degrees.

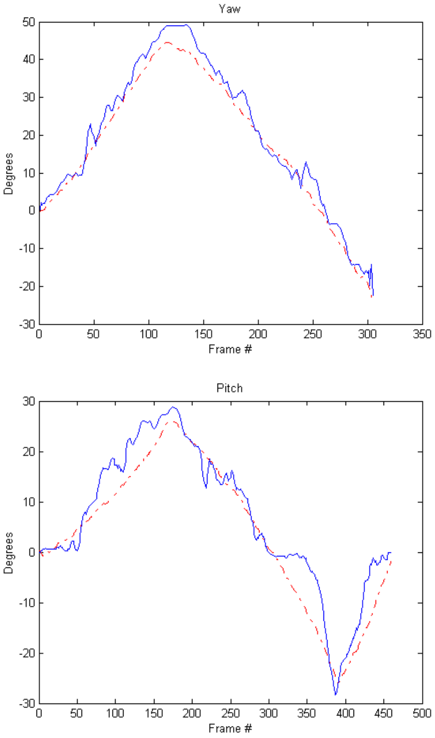


**Figure 3.** Comparison between the estimated face pose and the ground truth. In each graph, the dashed line depicts the ground truth and the solid line depicts the estimated face pose.

### 5.2 Sensitivity of the Face Pose Tracker

Two experiments are conducted to test sensitivity of the tracker to the initial face size and the initial face pose for the face model separately. One sequence that contains a person rotating his face naturally in front of the camera is chosen for this experiment. First, we perturb the size of face, which is represented by the face width, by ±5 and ±10 percent of the estimated face width. Figure 4 shows some tracking results corresponding to the different variations of different face sizes. We can see the tracking trajectories basically follow the same trend though there are some small deviations at certain frames. Second, we simulate the perturbations to the frontal-parallel pose of the face model by choosing the initial face model from the image frame, which contains the face under the non-frontal-parallel pose. Then the tracking is started from that image frame. We choose frames 1, 8, 12 and 14 as the starting frame for tracking respectively. They roughly correspond to the following face poses: $(0°, 0°, 0°)$, $(-0.4°, -0.5°, -1°)$, $(-8.8°, -10.9°, -2°)$ and $(-11.7°, -12.8°, -2.7°)$. Figure 4 shows some tracking results.

We can conclude that the tracker is not very sensitive to slight variations of initial face poses.

### 5.3 Face Model Updating

The initial face model is obtained from a frontal-parallel face image. When there are significant face appearance changes, either caused by face rotations, lighting changes, face expression, or the combinations of them, the frontal-parallel face model is not suitable for measuring the similarity between these images and the face model any more. Figure 5 shows that the face pose tracker will fail to track the face pose due to the significant face appearance changes when no face model updating is involved.

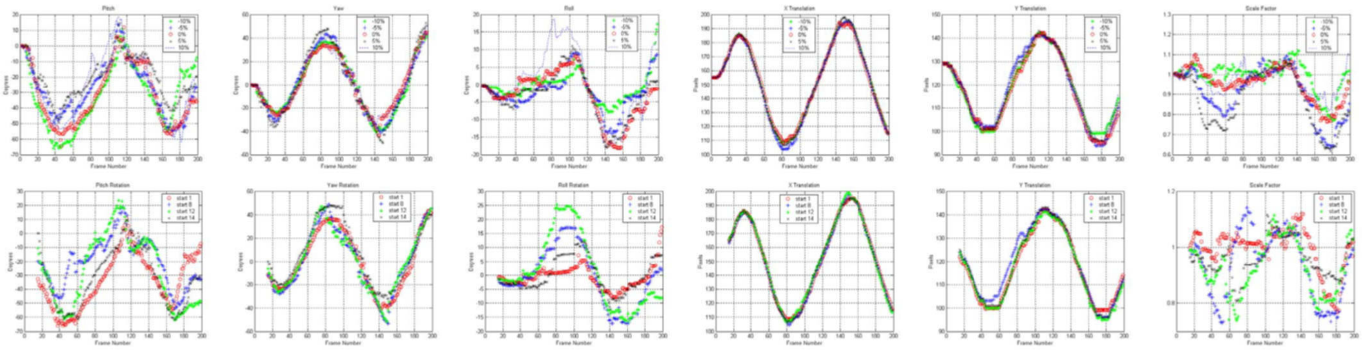
The face model is updated when significant aspect change has occurred between the face model and current face pose. Face model updating allows to successfully track face poses, which the tracker without face model updating will fail previously as demonstrated in Figure 5.

### 5.4 Convergence and Speed

In our implementation, tracking speed averages about 10 frames a second using the Powell minimization method in a computer with an Intel Pentium 4 processor and $512M$ memory. This performance number includes the time needed to extract images from the video sequence and save them back into the hard disk. Further speed gain can be obtained with a faster computer and with additional optimization of the codes. Tracking results of our system are shown in Figure 6. Video demos of our system may be found at http://www.ecse.rpi.edu/∼cvrl/Demo/demo.html.

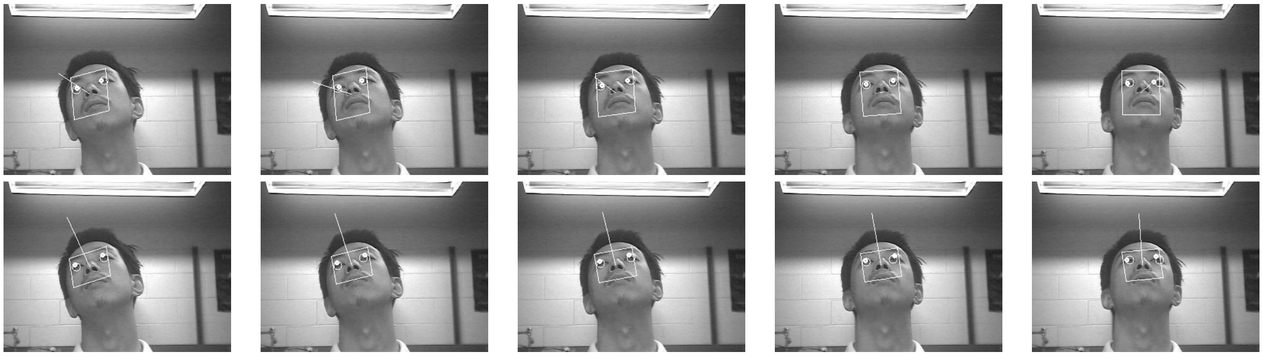
## 6 Advantages

Unlike conventional face pose estimation techniques, which often perform face detection and face pose estimation separately, our technique performs both concurrently, allowing one to complement another. Tracking 3D pose using Kalman Filtering effectively converts the



Pitch rotation Yaw rotation Roll rotation X translation Y translation Scale factor $\lambda$

**Figure 4.** (a) First row: sensitivity of the tracker to errors in estimating size of the initial face model. The face size is perturbed by ±5 and ±10 percent off the estimated face width. (b) Second row: sensitivity of the tracker to errors in the initial face pose of face model. In each graph, the curves correspond to image frames 1, 8, 12 and 14.



Frame 64 Frame 68 Frame 72 Frame 74 Frame 78

**Figure 5.** When significant face aspects occur, the face pose tracker without model updating will fail as shown in top row images. The proposed pose tracker still can track the face poses successfully under these circumstances, as shown in the images in the bottom row. In all images, white lines represent the face norms, as determined by the estimated poses. The two small circles represent the detected eyes, and the solid circles represent the projected eyes from the face model using the estimated pose parameters.

conventional inverse face pose estimation problem to a forward problem, therefore leading to a unique solution. Our method does not require tracking facial features. It performs face pose tracking from a monocular uncalibrated camera.

The pure planar tracker can accurately estimate the position of head, but tends to lose the target as soon as there is some significant out of plane rotations. This drift issue can be successfully handled by the detected eyes information. Our eyes detection and tracking combine active IR with mean-shift, therefore, it is very robust under various face rotations and illuminations. The detected locations of eyes can therefore constrain the face location in the image successfully. Therefore, our technique overcomes problems with the existing face pose estimation techniques based on planar models by allowing significantly out-of-plane rotations. In the meantime, our technique preserves its simplicity as contrasted with some other more complex face models.

Our face pose tracker is based on minimizing the sum-of-squared (SSD) between the projected face view image and the detected face view image in the input image. As reported in [22, 23, 24, 25], pure SSD based region tracking is sensitive to the appearance changes. Therefore, we have to account for the face appearance changes caused by face pose, facial expression, illumination and the combinations of them. The need to update face model dynamically is to ensure that the current face model is compatible with current face aspect. This is another critical point to the success of our system, which leads to robust and accurate tracking even under significant facial expression changes, large out-of-plane rotation and illumination changes.

Further, the use of IR sensing does not limit the mobility of the person. Most existing 3D face pose trackers are for stationary people (e.g. people in front of the computer) instead of a mobile person. So, in this regard, our system is applicable and is more robust because of IR

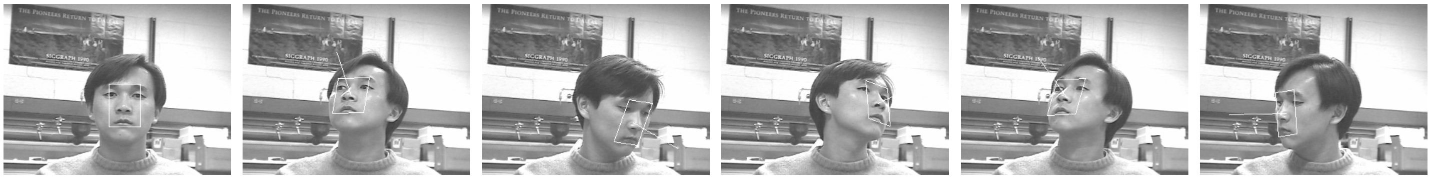


**Figure 6.** Face and face pose tracking result images taken from second video sequence from experiments, which are randomly selected. The white rectangle indicates the tracked face region and the white line is the norm of the face plane which is drawn according to those three angles.

sensing.

Simply, the benefits of our approach include: 1) working with a single uncalibrated camera; 2) allowing significant out-of-plane face rotations; 3) no need to track any facial features, 4) automatic face detection; 5) not very sensitive to face appearance changes introduced by pose, expression, illumination and the combination of them; 6) automatic failure recovery.

## 7 Conclusions

In this paper,we present a technique for simultaneous 3D face pose tracking and face detection under different face orientations, facial expressions and illumination changes in real time. Experimental results show how our technique greatly improves the standard pure planar face tracker, but still enjoys its simplicity. Based on the proposed techniques, we have built a real time working system that will start to track the face and estimate the 3D face pose as soon as the user is sitting in front of the camera. Our system is robust and accurate enough to track the face and estimate face pose, and it is suitable for the vision based applications such as HCI, face recognition and virtual reality.

## References

[1] S. McKenna and S. Gong, "Real-time face pose estimation," *Int. J. Real Time Imaging, Special Issue on Visual Monitoring and Inspection*, vol. 4, pp. 333–347, 1998.

[2] M. Motwani and Q. Ji, "3d face pose discrimination using wavelets," in *ICIP*, October 7-10,2001.

[3] S. Nayar, H. Murase, and S. Nene, "Parametric appearance repesentation," in *In Early Viual Learning. Oxford University Press*, 1996.

[4] J. Huang, X. Shao, and H. Wechsler, "Face pose discrimination using support vector machines (svm)," in *ICPR*, 1998.

[5] N. Kruger, M. Potzsch, , and C. v.d. Malsburg, "Determination of face positions and pose with a learned representation based on labeled graphs," *Image and Vision Computing*, August 1997.

[6] P. Yao, G. Evans, and A. Calway, "Using affine correspondence to estimate 3-d facial pose," in *ICIP*, 2001, pp. 919–922.

[7] A. Gee and R. Cipolla, "Determining the gaze of faces in images," *Image and Vision Computing*, vol. 12, pp. 639–948, 1994.

[8] T. Horprasert, Y. Yacoob, and L.S. Davis, "Computing 3-d head orientation from a monocular image sequence," in *International Conference on AFGR*, 1996.

[9] Q. Ji, "3d face pose estimation and tracking from a monocular camera," *Image and vision computing*, 2002.

[10] D. DeCarlo and D. Metaxas, "The integration of optical flow and deformable models with applications to human face shape and motion estimation," *CVPR*, 1996.

[11] T. S. Jebara and A. Pentland, "Parameterized structure from motion for 3d adaptive feedback tracking of faces," *CVPR*, 1997.

[12] A. Schdl, A. Haro, and I. Essa, "Head tracking using a textured polygonal model," *Proceedings Workshop on Perc. User Interfaces*, 1998.

[13] S. Basu, I. Essa, and A. Pentland, "Motion regularization for model-based head tracking," *ICPR*, 1996.

[14] A. Azarbayejani, T. Starner, B. Horowitz, and A.Pentland, "Visually controlled graphics," *PAMI*, 1993.

[15] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models," *PAMI*, 2000.

[16] L. Brown, "3d head tracking using motion adaptive texture-mapping," *CVPR*, 2001.

[17] Sami Romdhani and Thomas Vetter, "Efficient, robust and accurate fitting of a 3d morphable model," *International Conference in Computer Vision*, 2003.

[18] E. Trucco and A. Verri, "Introductory techniques for 3-d computer vision," *Prentice-Hall, New Jersey*, 1998.

[19] Kalman, Rudolph, and Emil, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[20] Carlos H. Morimoto and Myron Flicker, "Real-time multiple face detection using active illumination," *IEEE International Conference on AFGR, Grenoble, France*, March 2000.

[21] Haisong Gu, Qiang Ji, and Zhiwei Zhu, "Active facial tracking for fatigue detection," 2002.

[22] Gregory D. Hager and Peter N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *PAMI, Vol. 20, No. 10*, October 1998.

[23] J. Shi and C. Tomisito, "Good features to track," *CVPR*, 1994.

[24] N. P. Papanikolopoulos, "selection of features and evaluation of visual measurements during robotic visual servoing tasks," *Journal of Intelligent and Robotic System, 13(3), 279-304*, 1995.

[25] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *IJCV, 2(3), 283-310*, 1989.

0-7695-2158-4/04 $20.00 (C) 2004 IEEE