3D Face Pose Tracking From an Uncalibrated Monocular Camera

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, which most of the existing techniques do, 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. Furthermore, detected eyes allow us to automatically construct the initial face model using anthropometric proportions. Finally, the face model is dynamically updated to account for variations in face appearances under different face orientations.

Compared with the existing 3D face pose tracking techniques, our technique enjoys simplicity (due to the use of planar face model), allowing large out-of-plane face rotations and working with a single uncalibrated camera. It is also accurate, robust and less sensitive to illumination changes due to the use of active IR, and face detection is done automatically.

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 [5, 6, 7, 8, 9, 10] and model-based approaches [1, 2, 3, 4]. 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, though several efforts, such as wavelet transform, 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 [2, 3, 1]. So, for most of model-based approaches, pose estimation is done after face feature tracking. Though simple to be implemented, robust and accurate facial feature tracking is often a significant challenge. Others [11, 12, 13, 14, 15, 16, 17] have proposed to model face explicitly and perform pose estimation using the model. Ji et al. [17] 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. Black and Yacoob [11] have developed a regularized optical flow method that uses an eight parameter 2D model of optical flow, which generates good results. This method will fail when presented with large head rotations. Sumit Basu et al [15] 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. La Cascia et al [18] present a 3D head tracking system that is robust to changing illumination conditions. In their technique, the head is modelled as a texture mapped cylinder, and the tracking is formulated as an image registration problem in the cylinder's texture mapped images, which are generated by different face poses. Because the human head is not truly cylindrical, the accuracy of the system, as explained by Lisa M Brown [19], degenerates under some conditions, such as large head rotations around the vertical axis and large frame to frame head pose changes. Also, Lisa M Brown [19] found that it is unable to distinguish rotations around the horizontal axis and vertical translations or similarly rotations around the vertical axis and the horizontal translations. In order to improve
the accuracy of the head tracker, Lisa M Brown [19] presented some methods to overcome these problems.

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. While intuitively natural, the main problem with all these approaches is that face detection and face pose estimation are carried 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. The current techniques in face detection and tracking, however, are usually carried out solely in the 2D image based on face color, texture, or motion [20, 21, 22, 23, 24] independent from the 3D face poses. The detected 2D image faces (or facial features) are then used for face pose estimation, despite the errors of the detected face. Furthermore, the 3D face pose estimation from 2D image is an inverse process. The solutions are not unique and are susceptible to errors of the detected face. 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.

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.

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, and the face detection and face pose estimation are synchronized and kept consistent with each other. 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.

Basic projection equation of weak perspective camera model for planar 3D object points can be expressed in terms of relative coordinates as follows:

$$
\begin{pmatrix}
    c - c_1 \\
    r - r_1
\end{pmatrix} =
\begin{pmatrix}
    m_{11} & m_{12} \\
    m_{21} & m_{22}
\end{pmatrix}
\begin{pmatrix}
    x - x_1 \\
    y - y_1
\end{pmatrix}
$$

(1)

$m_{ij}$ are elements of the projection matrix and $(c_1, r_1)$ is the image projection of a 3D object point $(x_1, y_1, z_1)$.

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 $\kappa$. Elements of the projection matrix can be parameterized as

$$
\begin{align*}
    m_{11} &= \lambda \cos \phi \cos \kappa \\
    m_{12} &= \lambda \cos \phi \sin \kappa \\
    m_{21} &= \lambda (\sin \phi \sin \cos \kappa - \sin \kappa \cos \omega) \\
    m_{22} &= \lambda (\sin \phi \sin \sin \kappa + \cos \kappa \cos \omega)
\end{align*}
$$

where $\lambda$ is a scale factor, representing the distance from face to camera.

The 3D pose of a face can therefore be characterized by the three Euler angles and the scalar $\lambda$. 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.
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 Face Image Projection Via Weak-Perspective Model

4.1.1 3D Face Model

We assume that the face is a minimum rectangle that includes major facial features: eyes, nose, and mouth, which are symmetrically located within the rectangle. In 3D, given the positions of two eyes, we can use some anthropometric proportions to roughly obtain the location and scope of the face rectangle. The ideal frontal face region should contain all the salient facial features such as eyes, eye brows, nose and mouth. However, it should exclude hair and cheek as much as possible. Specifically, given $d_{eyes}$, which is the distance between two eyes, the dimensions of the face region ($W_{face}, H_{face}$) are approximately determined as follows

$$W_{face} = d_{eyes}/0.618$$
$$H_{face} = d_{eyes}/0.618$$

The golden ratio 0.618 is used here, because the results of ($W_{face}, H_{face}$) using 0.618 are approximately true for most people. Further, the center of the face region ($X_c, Y_c$) is determined by

$$X_c = (X_{lefteye} + X_{righteye})/2$$
$$Y_c = Y_{lefteye} + H_{face} * 0.25$$

where ($X_{lefteye}, Y_{lefteye}$) and ($X_{righteye}, Y_{righteye}$) are the left and right eyes positions. Figure 2 shows the face model and the associated ratios used to determine the scope and location of a face given its two eyes.

![Figure 2. Face model ratios with eyes locations.](image)

When subjects are facing directly to the camera (frontal-parallel face pose), these ratios are preserved in the image under weak perspective projection. Given the detected eyes positions, we can use the frontal-parallel image of the face as the initial 3D face model, as shown in Figure 3.

![Figure 3. The initial face model](image)

The face pose parameters for the initial face model are

$$\begin{align*}
\omega &= 0^\circ \\
\phi &= 0^\circ \\
\kappa &= 0^\circ \\
\lambda &= 1
\end{align*}$$

where \(\lambda\), without loss of generality, is normalized to 1. The corresponding projection matrix \(M\) is:

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

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 \((c_1, r_c)\) in the row-column image plane.

4.1.2 Image Homography

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 equation 1 yields

$$\begin{pmatrix}
c_2 - c_2 \\
r_2 - r_2
\end{pmatrix} = M_2 * M_1^{-1} * \begin{pmatrix}
c_1 - c_1 \\
r_1 - r_1
\end{pmatrix}$$

where \(M_1\) and \(M_2\) are the projection matrices for image \(I_1\) and \(I_2\) respectively, and \((c_1, r_1)\) and \((c_2, r_2)\) 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 5 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 5, 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.

4.2 Face Pose Tracking Algorithm

4.2.1 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 4. The eye tracking technique is then activated to detect eyes. After detecting the eyes, the first step is to compute the
distance $d_{\text{eyes}}$ 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 4.

![Figure 4. Detected eyes are marked with white rectangles and the face region with black rectangle. Also, eye centers and face center are marked by white crosses.](image)

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.

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 fronto-parallel face pose and to perturbations of initial positions of the face.

### 4.2.2 Tracking Algorithm

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 [26], the system can therefore be modelled as

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

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

   Given the system model, $X_{t+1}^*$, the state vector at $t + 1$, can be predicted by

   $$X_{t+1}^* = \Phi X_t + w_t$$  

   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 $X_{\text{e}}^{t+1}$. Then the final predicted face pose should be based on combining the one from Kalman with the one from eyes, i.e.,

   $$X_{t+1}^* = X_{t+1}^* + \Sigma_{t+1}^{-1}(X_{t+1}^* - X_{\text{e}}^{t+1})$$  

   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 4.3. Let the measurement model in the form needed by the Kalman filtering be

   $$z_{t+1} = H X_{t+1} + m_{t+1}$$  

   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 $X_{t+1}$ and its covariance matrix $\Sigma_{t+1}$

   $$X_{t+1} = X_{t+1}^* + \Sigma_{t+1}(z_{t+1} - H X_{t+1}^*)$$  


and
\[ \Sigma_{t+1} = (I - K_{t+1}H)\Sigma_{t+1}^{*} \]  
where \( K_{t+1} \) is the Kalman gain matrix and can be computed from
\[ K_{t+1} = \frac{\Sigma_{t+1}^{*}H^{T}}{HS_{t+1}^{*}H^{T} + S} \]  

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 \( t + 1 \) is updated based on successful face pose estimation for frame \( t + 1 \). Our study shows that it is important to update the face model periodically (every 10 frames) to account for the significant aspect changes under different face orientations.

4.3 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. \( \Sigma_{t+1}^{*} \) is 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 \( x_{t+1} \) 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 \( \Sigma_{t+1}^{*} \) over all the image pixels within the region of interest:
\[ E_{\text{matching}} = \sum_{i=1}^{N} (I_{p}(i) - I_{c}(i))^{2} \]  
where, \( I_{p}(i) \) is the pixel value of \( i \)th pixel in the reconstructed face view image \( I_{p} \), \( I_{c}(i) \) is the pixel value of the face image in the current image frame, and \( N \) the total pixel number of the reconstructed face view image. \( I_{c}(i) \) is projected from the reference image \( I_{c}(c, r) \) via a mapping function \( f \):
\[ I_{p} = f(I_{c}, \alpha) \]
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:
\[ \alpha_{n}^{*} = \arg \min_{\alpha} E_{\text{matching}} \]

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

4.3.1 Regularization of Minimization Criteria by Eyes Positions
Since different sets of pose parameters may produce the same image, yielding the same \( E_{\text{matching}} \), the minimization procedure may converge to a wrong place if it is left unconstrained. Therefore, 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 \( E_{\text{eyes}} \) between the detected eyes and the projected eyes can be expressed as
\[ E_{\text{eyes}} = E_{\text{Left}} + E_{\text{Right}} \]  
where \( E_{\text{Left}} \) is the Euclidean distance between the detected left eye and the projected left eye, and \( E_{\text{Right}} \) 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 13 and 14. Therefore, the criteria for the image matching minimization can be expressed as the sum of both terms
\[ E = \beta E_{\text{matching}} + (1 - \beta)E_{\text{eyes}} \]  
where \( \beta \) is a scale factor determining the relative importance of two terms. It is determined empirically.

5 Experimental Results

A series of experiments involving real image sequences are conducted to characterize the performance of our face pose estimation technique. First, we study the sensitivity of our algorithm to perturbations with initial face pose and placement. We then demonstrate the effectiveness of face model updating for accurate face pose tracking. Finally, we present results to show the robustness and accuracy of our algorithm for different individuals and under significant face rotations and movements.

The first experiment is designed to test sensitivity of the tracker to the initial face size for the face model. One sequence that contains a person rotating his face naturally in front of the camera is chosen for this experiment. We perturb the size of face, which is represented by the face width, by \( \pm 5 \) and \( \pm 10 \) percent of the estimated face width. Figure 5 shows the 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.

The second experiment is designed to test the sensitivity of the tracker to the initial face pose for the face model. The same sequence is used. We simulate the perturbations to the frontal-parallel pose of the face model by choosing the initial face model from
Figure 5. 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.

the image frame, which contains the face not under the 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: \((0^\circ, 0^\circ, 0^\circ), (-0.4^\circ, -0.5^\circ, -1^\circ), (-8.8^\circ, -10.9^\circ, -2^\circ)\) and \((-11.7^\circ, -12.8^\circ, -2.7^\circ)\). Figure 6 shows the tracking results.

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

5.1 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, or face expression, the frontal-parallel face model is not suitable for measuring the similarity between these images and the face model any more. Figure 7 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 7.

5.2 Convergence and Speed

In our implementation, tracking speed averages about 2 frames a second using the Powell minimization method in a computer with an Intel Pentium 3 processor and 256M 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.

5.3 Accuracy of Face Tracking and Face Pose Determination

The proposed algorithm is tested with numerous image sequences of different people. The first image sequence includes a person rotating his head before an un-calibrated camera, which is approximately 1.5 meter from the person. Figure 8 show some tracking results under different face rotations. The second image sequence includes a person rotating his head before another un-calibrated camera at the same distance. Tracking results are summarized in Figures 9. The figures show that the estimated pose is very visually convincing over a large range of head orientations and changing distance between the face and camera. Plots of three angles \(\omega, \phi, \kappa\) are shown in Figure 10, from which we can see the three angles are changed consistently with the smooth head rotations. Video demos of our system can be found at CVPR website.

Figure 10. The results of pose tracking. The plots show the sequences of rotation angles through the image sequence.

6 Conclusion

In this paper, we present a technique for simultaneous 3D face pose tracking and image face detection. Unlike the 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. The tracking 3D pose using Kalman Filtering effectively converts the 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.

Experimental results show robustness and good accuracy of our technique to track the face and face pose under different face orientations and changing distance from the face to the camera.

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. 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; and 5) not very sensitive to illumination due to the use of IR illumination, IR camera and dynamic face model updating. These benefits are resulted from the use of active IR for robust and accurate eyes tracking, which significantly improve the face detection accuracy and which subsequently can be used to constrain the face pose estimation. Further, the use of dynamic face updating ensures that the face model is consistent with the face images, which leads to robust and accurate tracking even under significant illumination changes and facial expression changes. Finally, the simultaneous face detection and face pose estimation is another reason for the improvements.

References

[10] Igor Elagin, Johannes Steffens, and Harriau Neven, “Automatic pose estimation system for human faces based on bunch
Figure 8. Face and face pose tracking result images taken from first 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 the estimated three face pose angles.

Figure 9. 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.


