Robust 3D Head Tracking by Online Feature Registration

Jun-Su Jang and Takeo Kanade
Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
{jsjang, tk}@cs.cmu.edu

Abstract

This paper presents a robust method for tracking the position and orientation of a head in videos. The proposed method can overcome occlusions and divergence problems. We introduce an online registration technique to detect and register feature point of the head while tracking. A set of point features is registered and updated for each reference pose serving a multi-view head detector. The online feature registration rectifies error accumulation and provides fast recovery after occlusion has ended, while preventing divergence problem which frequently occurs in conventional frame-to-frame tracking methods. The robustness of the proposed tracker is experimentally shown with video sequences that include occlusions and large pose variations.

1. Introduction

3D head tracking is more than tracking a face in video. It estimates 3 rotation parameters and 3 translation parameters of the head.

Selecting a geometric model to represent the head is important. The complexity of the model affects the working range of the tracker, ease of initialization, and degree of computation. A planar model is simple, but not covering large rotations [1–3], it works properly only when the head rotation was around the frontal view. To obtain larger working ranges and more accurate motion, an ellipsoidal model [4], cylinder models [5–8] and more sophisticated models [9, 10] have been studied. Complicated models can provide more accurate motion; however, they generally require careful initialization as well as more computation. We have chosen a cylinder model to represent the 3D shape of a head. The cylinder model includes both circular and elliptical cylinder. The simplicity of the cylinder model provides robustness from initialization error compared to other more sophisticated models.

A good tracking method should be able to deal with varying head poses, occlusions, illumination changes and facial expressions. Many methods were proposed by using template update and registration. Cascia et al. formulated an image registration problem in the cylinder's texture map [5]. They used a linear combination of texture-warped templates and illumination templates to handle illumination changes in tracking. Brown improved the texture-mapped cylinder approach by proposing adaptive motion templates to enhance the motion between successive frames and additional templates to cover large head rotations [6]. Xiao et al. applied a dynamic template technique in order to accommodate gradual changes in lighting and self-occlusion [8]. Some frames associated certain head poses were stored as references to prevent error accumulation due to the dynamic template.

It is obvious that two conflicting strategies, updating templates and keeping reference templates, should be balanced. Updating templates can cause error accumulation and divergence in tracking, while keeping reference templates cannot accommodate appearance changes. Although template update methods are used in many studies [3, 7, 8, 11], it is difficult to obtain both adaptability and stability in tracking performance. One easy way to avoid the divergence problem is tracking-by-detection [12, 13]. A detector can be applied to each individual frame to prevent drift and divergence, which may occur in conventional frame-to-frame tracking using the template update technique. However, in the 3D head tracking problem, there is difficulty in making a universal detector, which can cover appearance differences among individuals, wide out-of-plane rotation, illumination, etc.

In this paper, we propose a cylinder model-based 3D head tracker using the online feature registration. The cylinder model covers a wide range of head motions and the online feature registration deals with the tracking-by-detection issue mentioned above. To avoid making a generic head detector, we only focus on the current individual in tracking sequence, since it makes detection problems much easier. The online feature registration technique stores the feature points of a head for each reference pose while tracking.

The overall tracking system is shown in Figure 1. An initial estimation of head pose uses Bayesian tangent shape model (BTSM) face alignment method [15]. The BTSM

978-1-4244-2154-1/08/$25.00

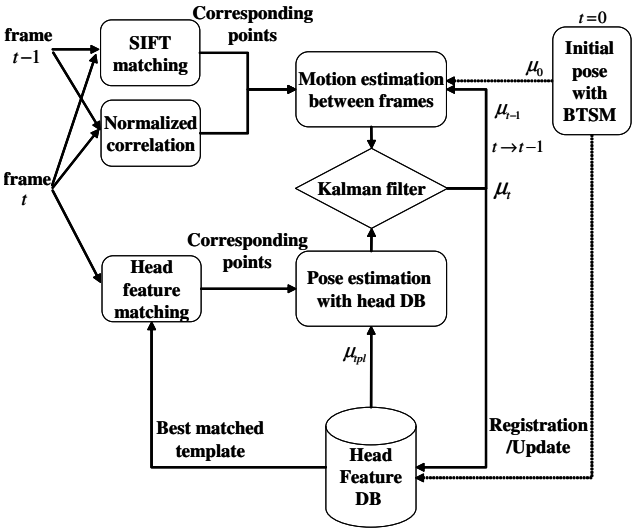


Figure 1. Overall architecture of the 3D head tracker

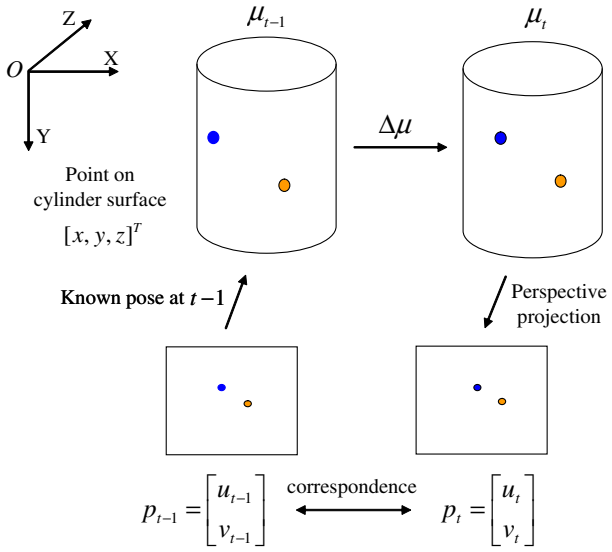


Figure 2. Cylinder motion estimation using known corresponding point pairs

face alignment gives a set of facial points, so that we can estimate the 3D pose of the head. Even though it only works properly for frontal face, it is more informative to initialize the 3D head pose than general face detectors which give a bounding box. Once a frontal head is initialized, our tracker works automatically. The scale invariant feature transform (SIFT) [14] is used to extract and match feature points. The set of SIFT feature points forms a view-based head feature database (DB), which provides robust performance in occlusions. Normalize correlation method is used to find corresponding points between successive frames together with SIFT. Kalman filter is then applied to combine the estimated motion between successive frames and the estimated pose with head feature DB.

2. Cylinder Motion Estimation

2.1. Rigid motion under perspective projection

This section presents a method that estimates a rigid cylinder motion, $\Delta\mu = [\Delta\theta_x, \Delta\theta_y, \Delta\theta_z, \Delta x, \Delta y, \Delta z]^T$, where $\Delta\theta_x, \Delta\theta_y, \Delta\theta_z$ represent 3 rotations (pitch, yaw, and roll) and $\Delta x, \Delta y, \Delta z$ represent 3 translations. Let a point in an image at time t be $p_t = [u_t, v_t]^T$. Given a known pose μ_{t-1} of a cylinder at time $t-1$, we can calculate 3D point $X_{t-1} = [x_{t-1}, y_{t-1}, z_{t-1}]^T$ in the world coordinate by assuming that the point p_{t-1} is on the cylinder surface. The motion between X_{t-1} and X_t can be represented by using twist representation [16]:

$$\begin{bmatrix} x_t \\ y_t \\ z_t \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -\Delta\theta_z & \Delta\theta_y & \Delta t_x \\ \Delta\theta_z & 1 & -\Delta\theta_x & \Delta t_y \\ -\Delta\theta_y & \Delta\theta_x & 1 & \Delta t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \\ 1 \end{bmatrix} \quad (1)$$

An expected projection point, p'_t, is calculated by using the 3D point and motion vector $\Delta\mu$.

$$p'_t = \begin{bmatrix} x_{t-1} - y_{t-1}\Delta\theta_z + z_{t-1}\Delta\theta_y + \Delta t_x \\ x_{t-1}\Delta\theta_z + y_{t-1} - z_{t-1}\Delta\theta_x + \Delta t_y \end{bmatrix} \cdot \frac{f}{-x_{t-1}\Delta\theta_y + y_{t-1}\Delta\theta_x + z_{t-1} + \Delta t_z}, \quad (2)$$

where f is camera focal length. We assume that the focal length is unknown. If the depth variation of a cylinder is relatively smaller than the distance between the cylinder and the camera, the unknown focal length does not cause a large error in pose estimation [7].

The equation (2) maps p_{t-1} to new location p'_t. Assuming that the corresponding point pair, p_{t-1} and p_t, is found by image observation, we can compute the motion vector to minimize the sum of distance error, e_t, between expected and observed locations of corresponding point pairs:

$$e_t = \sum_{i=1}^{N} \|p'_{i,t} - p_{i,t}\| , \quad (3)$$

where $N \geq 3$. Figure 2 shows cylinder motion estimation method using known corresponding point pairs.

We use weighted least squares (WLS) estimation to find motion vector. The WLS deals with outliers which can be obtained in the process of finding corresponding point pairs. The weight for each point is updated by using the distance error of the point:

$$w_{i,t} \leftarrow w_{i,t} \cdot \exp(-c \cdot e_{i,t}) \ , \tag{4}$$

where c is a positive constant and $e_{i,t} = \|p'_{i,t} - p_{i,t}\|$. Every point has a weight value which indicates how much the point is consistent to the cylinder motion. The WLS is iteratively applied until convergence.

2.2. Feature matching

To find the corresponding point pairs between two images, two kinds of feature matching approaches are used. First, SIFT is used to obtain distinctive feature points in images. It is invariant under scaling, rotation and limited view change. Each feature point has a 128-dimensional descriptor for matching. The advantage of SIFT feature is that it provides wide baseline matching with low false positive rate. Therefore, it is suitable to make up a head feature DB (see section 3), which is to detect head feature points regardless of the pose difference between two images.

Second, we generate regularly placed feature points inside of the head region based on current pose estimation. Normalized correlation method is applied to find matching points. A rectangle region centered on a grid-type point is extracted to compute normalized correlation to adjacent regions. A point which has the maximum correlation value (larger than proper threshold) is chosen as a corresponding point by searching over the adjacent regions of each point. We consider this feature as a complementary feature to SIFT. Because SIFT feature points may not uniformly appear inside of the head region and the number of SIFT feature points varies with image quality. We can control the number and the location of grid-type feature points. Normalized correlation works when motion between two images is relatively small, so that candidate search regions should be assigned properly.

Both kinds of features are used to find corresponding points between successive frames. SIFT features are also used to find corresponding points between current frame and head feature DB presented in the following section.

3. Online Feature Registration

The proposed tracker gathers head feature points from a past sequence to improve its tracking performance in the future sequence. 2D image observation of the head region varies a lot when the head moves with large rotation, especially out-of-plane rotation. Use of an initial reference template throughout the whole tracking sequence is not recommended. For example, typical head tracking starts with the frontal face region as a reference template. When the head rotates about axis Y, one of the eyes becomes invisible and the profile area of the head appears. The reference template does not cover the new appearing region, which may contain useful features to help tracking.

(a) SIFT feature points (b) Regularly placed feature points

Figure 3. Two kinds of feature points

We use SIFT features to make up a head feature DB. Although we successfully obtain the SIFT feature points which indicate the same 3D points between two images, the descriptor may be not matched well when out-of-plane rotation exceeds some bounds. To make the head feature DB cover a large range of out-of-plane rotation, multi-view approach is considered. Basically, SIFT features and an associate head pose are stored when current estimation of head pose comes close to one of certain reference poses. Reference poses are decided in out-of-plane rotations (pitch and yaw), because in-plane rotation (roll) is covered by SIFT features which is invariant to that rotation.

A head feature DB consists of many view-based templates, and each view template contains a set of SIFT features and head pose. Generated head feature DB is used to estimate poses in the remaining frames. When input frame comes into tracker, a template which has the most number of matched feature points is selected as the best matched template. The head pose of input is estimated by using the best matched template in the same way as described in section 2. An example of a head feature DB obtained in a real tracking sequence is shown in Figure 4. The locations of the feature points are displayed with certain views of the head. It is not necessary for feature points to have corresponding points among templates.

Each feature point in the DB has an accumulated weight as a confidence value. When a point is matched between the current frame and head feature DB, the accumulation of weight is

$$w^{acc}_{i,t} = w^{acc}_{i,t-1} + w_{i,t} \ , \tag{5}$$

where $w_{i,t}$ is a weight from the WLS in (4). A point with high weight means that it was matched frequently and moved consistently with head motion. The accumulated

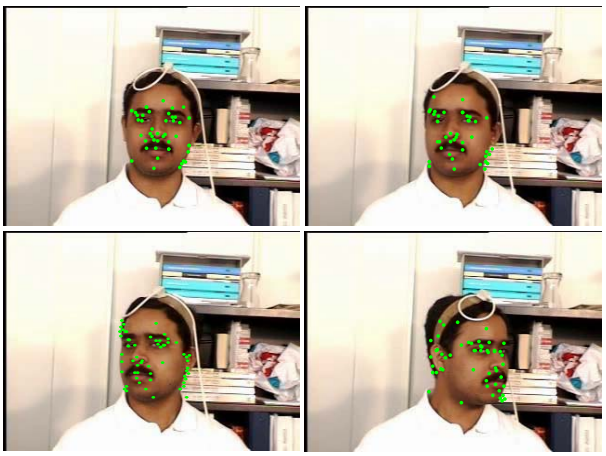


Figure 4. Example of a head feature DB

weight is used to assign the initial weight in the WLS iterations, therefore, the WLS gets rid of outliers at an earlier iteration. Feature points in the newly generated template inherit the accumulated weights from feature points in the existing neighbor view template, if they are matched.

Once the online feature registration method makes a head feature DB to cover a large range of view, the obtained DB can be seen as a multi-view head detector for the current individual. Our approach is different from the previous studies [6, 8], that store view-based templates to cover a large range of rotation. In their methods, both a candidate head region in the input frame and the closest template must be selected properly, which is difficult when the head pose in the current frame and that of the selected template are quite different. Especially, divergence in tracking occurs frequently when the candidate head region is selected inaccurately. To recover divergence in tracking, their methods need to pick a starting frame of divergence and re-initialize by using general detectors. On the other hand, our individual-specific head detector works regardless of the pose difference. SIFT feature points are matched in the whole input frame so that a candidate head region is not needed. Therefore, it fundamentally avoids divergence in tracking problems.

4. Tracking with Kalman Filter

There are two ways of estimating the current pose of a head in our method. The first is from accumulating motions between successive frames and the other is from estimating pose of the current frame using head feature DB. Kalman Filter is applied to combine the two kinds of information. Let u be a pose difference between successive frames; it is regarded as a control input in Kalman filter framework. The state transition equation is derived as

$$\mu_t = \mu_{t-1} + u_{t-1} + \alpha_{t-1} \ , \tag{6}$$

where α_t represents the process noise. This equation stands for the transition of state vector, μ. Many studies for tracking problems use dynamics based on smooth movement, which makes the prediction stage fail when a sudden rapid movement occurs. We do not assume a smooth head movement, so that the state transition equation contains the control input u. The noise α_t is assumed to be normally distributed as, $\alpha_t \sim N(0, Q_t)$. The covariance matrix Q_t is assumed by a diagonal matrix whose elements are set by using root mean square error (RMSE) of distance errors $e_{i,t}$ in (4).

Let h be a head pose calculated with the head feature DB. Then the observation equation is derived as

$$h_t = \mu_t + \beta_t \ , \tag{7}$$

where β_t represents the observation noise with $\beta_t \sim N(0, R_t)$. Similar to Q_t, R_t is set by using RMSE in the pose estimation process from the current frame and the head feature DB. The outputs of Kalman filter are estimated pose μ_t and covariance matrix P_t as a confidence measure of current estimation. The head feature DB is updated using P_t. The current view template replaces an existing view template, if the current estimated pose is close to the pose of the existing template and the covariance matrices satisfy following equation:

$$\|P_t\| < \|P_{tpl}\| \ , \tag{8}$$

where $\|P_{tpl}\|$ means covariance matrix of the existing template. P_t is stored as P_{tpl} after template replacement. The accumulated weights of feature points are inherited from the old template for matched points.

5. Experimental Results

We tested the proposed tracking system in three experiments. Throughout the experiments, an elliptical cylinder with a radius ratio of 1.3 was used to cover the side regions of the head, including ears.

5.1. Sequences with ground truth

The first experiment was done using Boston University dataset which provided the ground truth of the 3D pose [5]. We compared the pitch, yaw, and roll estimated by our tracker to the ground truth. Figure 5 shows the rotation parameters for two different sequences. We tested 45 sequences in the dataset; the average estimation errors for pitch, yaw, and roll were $3.7°, 4.6°$ and $2.1°$, respectively.

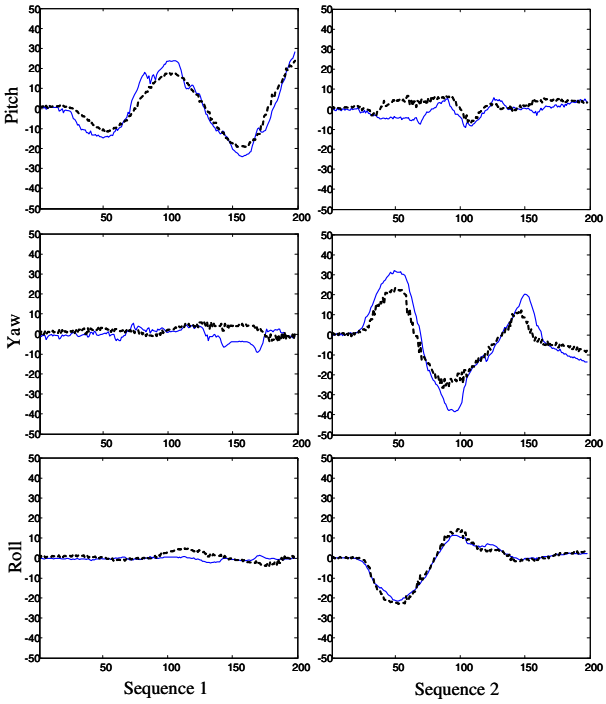


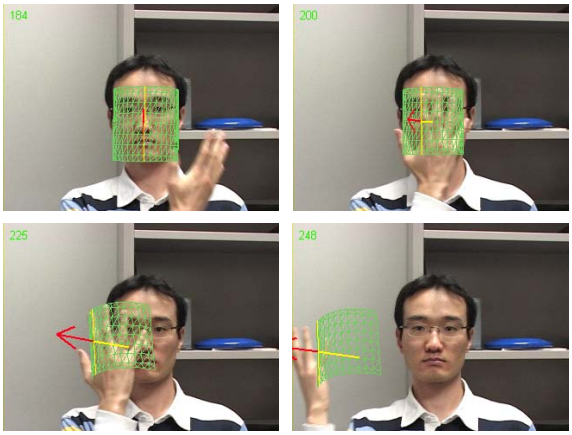
Figure 5. Comparison with the ground truth. Each column shows 3 rotation parameters from a sequence. Blue sold lines indicate estimated results, and black dashed lines indicate the ground truth.

5.2. Comparison with texture-mapped tracker

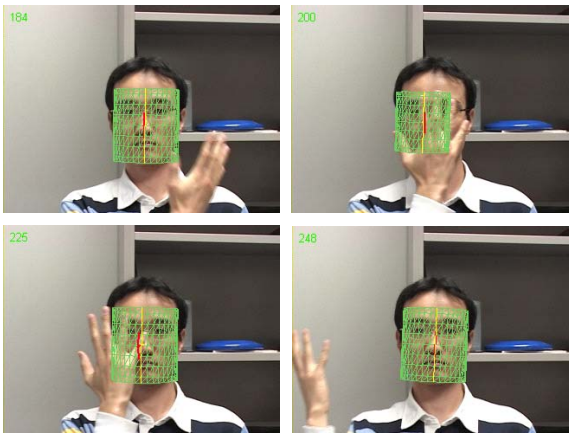
For the second experiment, we compared our method to a texture-mapped cylinder tracker which incorporated a dynamic template and multi-view template registration [8]. Optical flow method was used to track cylinder surface regions. The dynamic template dealt with the appearance changes, however, it caused the divergence in tracking. As shown in Figure 6(a), the tracker chased a hand after the hand occluded the head region. Their method needed re-initialization by using a face/head detector to recover the pose after occlusion had ended, although multi-view templates were prepared. Because view-based texture templates were only available on the assumption that the current candidate head region was extracted successfully. Our method overcame the occlusion and divergence problem as shown in Figure 6(b). Once the head region reappeared enough to match input feature points from the registered feature DB, our tracker immediately recovered the pose.

5.3. Sequences with large motion and occlusions

As the third experiment, the proposed tracker was tested with 40 real sequences that contained large head rotations, partial occlusions and complete occlusions. Figure 7 shows some examples. In the first frame 7(a), initial head pose



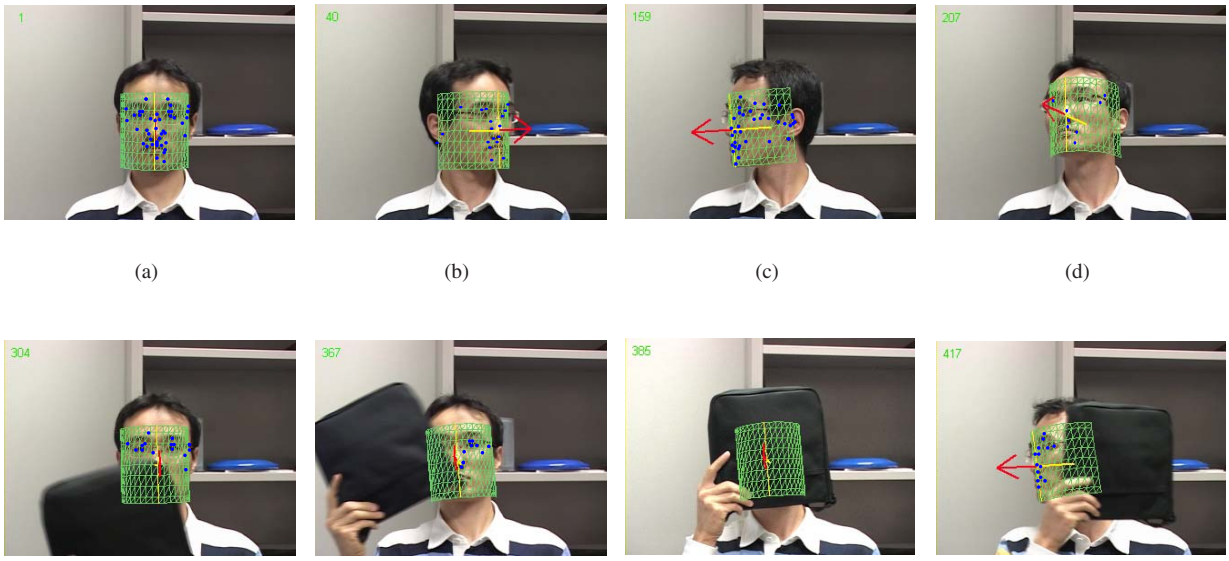
(a) Texture-mapped cylinder tracker



(b) Proposed tracker

Figure 6. Comparison between the texture-mapped cylinder tracker and the proposed tracker

was estimated by using BTSM face alignment. The tracker covered wide ranges of rotations, 7(b), 7(c), 7(d), and the online feature registration technique generated a head feature DB to make an individual-specific head detector. This detector covered a large range of views observed in previous frames in the sequence. The tracker showed robustness to partial occlusion in 7(e), 7(f). When the head was completely occluded 7(g), the tracker lost the head and held the last successfully estimated pose. In 7(h), the tracker recovered the head pose immediately when the head region started to show partially. It should be noted that the tracker rapidly recovered the head pose, even though the head reappeared with a largely rotated view. The tracker started with only frontal head information, after that, it learned the other views of the head in tracking sequence and constructed a multi-view head detector to improve the tracking perfor-



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7. Tracking results under large rotations and occlusions. Blue points indicate matched points from the head feature DB.

mance.

6. Conclusions

In this paper, we presented a robust 3D head tracking system using online feature registration. The proposed method incorporates motion estimation between successive frames with pose estimation from the head feature DB. The WLS method was used to reject outlier feature points. After observing the current individual's head movement, our tracker generated an individual-specific head detector. The obtained detector prevented tracking error accumulation and divergence, and recovered head pose rapidly when occlusion ended.

For future work, we plan to research head tracking with non-rigid motion. Analyzing the multiple observations of the same feature points is needed to discriminate the non-rigid points from rigid points.

References

[1] M. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *IJCV,* vol. 25, no. 1, pp. 23-48, 1997.

[2] G.D. Hager and P.N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Trans. PAMI,* vol. 20, no. 10, pp. 1025-1039, 1998.

[3] Z. Zhu and Q. Ji, "Real time 3D face pose tracking from an uncalibrated camera," in *CVPRW,* pp. 73, 2004.

[4] S. Basu, I. Essa and A. Pentland, "Motion regularization for model-based head tracking," in *ICPR,* pp. 611-616, 1996.

[5] 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," *IEEE Trans. PAMI,* 2000.

[6] L. Brown, "3D head tracking using motion adaptive texturemapping," in *CVPR,* 2001.

[7] G. Aggarwal, A. Veeraraghavan, and R. Chellappa. "3D facial pose tracking in uncalibrated videos," in *PRMI,* pp. 515-520, 2005.

[8] J. Xiao, T. Kanade and J. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," in *FG,* pp. 156-162, 2002.

[9] L. Lu, X.-T. Dai, G. Hager, "A particle filter without dynamics for robust 3D face tracking," in *CVPRW,* pp. 70, 2004

[10] M. Malciu and F. Preteux, "A robust model-based approach for 3D head tracking in video sequences," in *FG,* pp. 169-174, 2000.

[11] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. PAMI,* vol. 25, no. 10, pp. 1296-1311, 2003.

[12] M. Özuysal, V. Lepetit, F. Fleuret and P. Fua, "Feature harvesting for tracking-by-detection," in *ECCV,* pp. 592-605, 2006

[13] M Grabner, H Grabner and H Bischof, "Learning features for tracking," in *CVPR,* 2007.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV,* vol. 2, no. 60, pp. 91-110, 2004.

[15] Y. Zhou, L. Gu and H.-J. Zhang, "Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference," in *CVPR,* pp. 109-116, 2003.

[16] R.M. Murray, Z. Li, and S.S. Sastry, *A Mathematical introduction to robotic manipulation,* CRC Press, 1994.