Real Time Head Pose Tracking from Multiple Cameras with a Generic Model

Q. Cai
Microsoft Research
1 Microsoft Way, Redmond, WA
qincai@microsoft.com

A. Sankaranarayanan
University of Maryland
College Park, MD
saswin@gmail.com

Q. Zhang
The University of Kentucky
1 Quality Street, Lexington, KY
zhangqing06@gmail.com

Z. Zhang and Z. Liu
Microsoft Research
1 Microsoft Way, Redmond, WA
zhang, zliu@microsoft.com

Abstract

We present a robust approach to real-time 3D head pose tracking using multiple cameras with unknown camera placements. Many important applications do not want prior multi-camera calibration. We exploit a generic face model to overcome the difficulties due to the lack of prior knowledge of camera placement and the severe head appearance difference across cameras. We propose a fast drift-free solution based on feature point tracking using reference frames of high confidence over the temporal and spatial domains. Our algorithm tracks feature points from Harris feature detector, but not necessarily points of face landmarks. The relative camera placement is refined progressively at the same time as the user's head pose is resolved. Compared to single camera tracking, the use of multiple cameras increases the reliability of tracking by covering a wider range of poses as well as providing more accurate head pose estimation. We have tested the algorithm on many subjects in a variety of environments. A live demonstration will be shown at the conference.

1. Introduction

3D head pose information is very important for many applications. It can be used, for example, in video-driven avatar animation for entertainment or for model-based low-bandwidth teleconferencing. It is also useful for face recognition, face relighting, gaze correction and adaptive interface. In this paper, we present a real time head pose tracking algorithm using multiple cameras, with unknown placement geometry.

Tracking of the human head pose has been studied extensively in existing literature. Model-based head pose tracking algorithms (for a survey, cf. [6]) can be clustered in terms of the models used to represent the human head, ranging from simple geometric shapes such as cylinders [1] and ellipsoids [10], to more complicated generic and morphable 3D face models. However, the cylindrical and ellipsoidal model are at best a coarse approximation of the face and show significant deviations at extreme poses. Generic face models are more suitable for most of the applications using the head pose without extreme precision of face matching.

A real-time monocular pose tracking algorithm in [15] combines information from a few keyframes (learnt a priori) and preceding frames to track the object. In particular, feature correspondences are established across both short and wide baselines. The use of keyframes also reduces the effect of drift. Wang et al. [17] formulate the pose tracking problem as a Bayesian inference incorporating the correspondences from the previous frames as well as the keyframes together. A stochastic sampling method is used to sample the posterior density, and the maximum a posteriori estimate of the pose is computed. Morphable models are learnt by first identifying corresponding points (usually manually) on multiple images of the face and adapting the parameters of the generic face model to fit the labeled point correspondences [4].

However, monocular pose tracking algorithms for the human face inherently suffer when the face is far away from frontal because the face exhibits few reliable features that are trackable, and the mismatch due to modeling errors becomes pronounced. Towards making tracking performance reliable, observing the face from multiple views [8] allows robustly trackable features to be observable at any given time instant. Multi-camera based pose tracking has been studied in various contexts in existing literature. Tariq and Dellaert [14] present a head-pose tracker by rigidly mounting multiple cameras on the subject's head. The problem of head pose estimation is mapped to that of ego-motion estimation. We avoid such an approach that requires mounting devices on the subject because it is usually not preferred. Ruddarraju et al. [12] propose a head pose tracking algorithm using multiple calibrated (both external and internal parameters) cameras. Their approach relies on tracking face landmarks such as eye and mouth corners at each camera and triangulating their features. The robustness of this algorithm is limited due to the modeling of the face as a plane as well as the use of a small set of features.

Ba and Odobez [2] quantize the orientation of the head and obtain multiple exemplars of the face. A particle filter is used to track over the state space of the location of the head and the discretized pose. Voit and Stiefelhagen [16] use neural networks to obtain the pose at each camera from a template of the face, and use a Bayesian filter to fuse the individual estimates. Ng and Gong [11] use SVM to model changes on facial images with pose and use multi-view inputs to detect and estimate the head pose. In general, methods performing a discretization of the pose space are more suitable in low resolution imagery conditions in which 3D model fails because there are no face landmarks that can be reliably matched. Meanwhile, machine learning based approaches for pose estimation require a large amount of training data, which inherently discretizes the pose space and thus limits the accuracy of the pose estimates.

In this paper, we present a multi-camera based real-time pose tracking system without any prior knowledge of the camera placements. This is extremely useful for many applications wherein usability and ease-of-use often dictate that the system uses minimal prior calibration. We believe the work described in this paper is one of the first on multi-view head pose estimation using a generic face model with the following properties:

- it does not require prior calibration of relative camera geometry;
- it is not required, although beneficial when possible, to obtain explicit correspondences across cameras;
- it tracks feature points which are not just restricted to face landmarks.

The estimation of the relative camera geometry along with that of the head pose is done by aggregating the information from all the cameras, and is refined over time through optimization. In the remainder of the paper, we formulate the problem in section 2, describe our tracking framework in detail in section 3, then present experimental results in section 4 and conclusion in the end.

2. Problem Formulation

Assume the position of an object is represented by rotation $\mathbf{R}$ and translation $\mathbf{t}$, described by matrix $\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$, Given N multiple static cameras with unknown relative geometries $\mathbf{P}_{ij}$ $(i, j = 1, \ldots, N; i \neq j)$, we are interested in developing a robust framework for real time tracking on the common head pose of a subject in world coordinate at time t, i.e., $\mathbf{P}_w^t$, from the image sequence $\mathbf{I}_i^t$ captured by camera i using a 3D generic face model. Figure 1 shows the pose relationship between multiple cameras, the head model and world coordinate system, where $\mathbf{P}_i^t$ is the head pose at time t viewed from camera i, $\mathbf{P}_{wi}$ is the transformation from world to camera i, i.e. $\mathbf{P}_i^t = \mathbf{P}_{wi}\mathbf{P}_w^t$. One of the cameras c is chosen as the world reference, where c could change over time. Therefore $\mathbf{P}_i^t = \mathbf{P}_{ci}\mathbf{P}_c^t$. Our pose tracking problem becomes finding $\mathbf{P}_c^t$ and $\mathbf{P}_{ci}$.

2.1. Single Camera Tracking

Spatial Consistency: We start with single camera tracking by exploring the 2D-3D spatial consistency of feature points from camera i. We need to estimate $\mathbf{P}_i^t$ given input images $\mathbf{I}_i^t$ and feature points $k = 1, \ldots, K$ from the head model. From 2D-3D correspondences under the perspective projection, $\mathbf{P}_i^t$ is estimated by minimizing

$$e_i(t) = \sum_{k=1}^{K} \rho(\mathbf{u}_{i,k}^t - \phi(\mathbf{P}_i^t \mathbf{U}_k)) \quad (1)$$

where $\phi(.)$ is the perspective projection of a 3D point from the model in homogeneous coordinate $\mathbf{U}_k$ to its 2D correspondence $\mathbf{u}_{i,k}^t$ and $\rho(.)$ is an M-estimator chosen to alleviate gross noise interference. We use the POSIT iterative algorithm [3] as a pre-processing step to obtain the initial pose estimation at time t. Note we only apply a global scaling in X and Y to the 3D points in a generic face model to obtain an approximate personalized model at this moment. In the future, different scaling could be applied to various parts of the head model and make the model more accurate to a particular subject.

Temporal Smoothing: We apply an aging factor α $(0 < \alpha < 1)$ up to preceding τ frames to smooth pose estimation

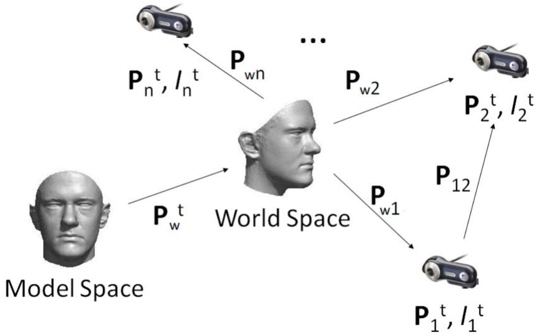


Figure 1. The relationship of poses between the model, the world, and local camera coordinate system.

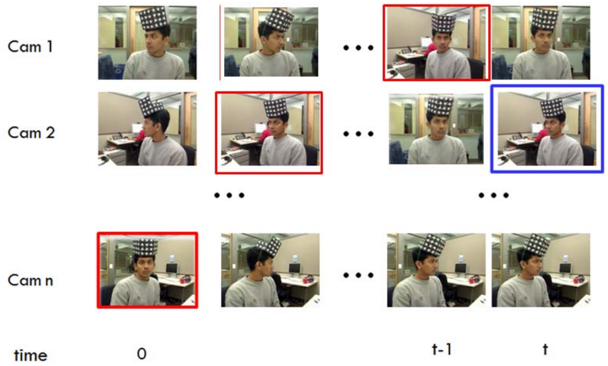


Figure 2. Robustness of a tracking algorithm using reference frames (highlighted in red between $t-1,\ldots,0$, the current frame is highlighted in blue rectangle) temporally and spatially across cameras.

temporally, thus the above minimization becomes

$$e_i^t = \sum_{n=0}^{\tau-1} \alpha^n e_i(t-n) \qquad (2)$$

where τ is set to 3 in our implementation.

Reference Frames: The above two constraints cannot prevent tracking from drifting caused by error propagation. A typical approach is to use reference frames from different time instants with a variety of poses. In a multi-camera tracking system described later, we can extend the reference frames across cameras. Figure 2 shows our generic framework of using reference images for tracking temporally and spatially. Thus we generalize Eqs. (1) and (2) into

$$\sum_r \omega_r e_r^t = \sum_r \omega_r \sum_k \rho(\mathbf{u}_{i,k}^t - \phi(\Delta\mathbf{P}_{ri}^t \mathbf{P}_r^t \mathbf{U}_k)) \qquad (3)$$

where r represents a reference frame which could be either a preceding frame up to time $t-\tau$ or a keyframe, ω_r is the weight for the reference frame based on timing window or tracking confidence, and $\Delta\mathbf{P}_{ri}^t$ is the pose difference from the reference frame to the current one.

2.2. Multiple Camera Tracking

As it is not guaranteed that a single camera will be able to track the face all the time, we extend the above formulation to multi-cameras, and the summarization of $e_i(t)$ in Eq. (1) becomes

$$\sum_{k=1}^{K} \rho(\mathbf{u}_{c,k}^t - \phi(\mathbf{P}_c^t \mathbf{U}_k)) + \sum_{i=1, i\neq c}^{N} \sum_{k=1}^{K} \rho(\mathbf{u}_{i,k}^t - \phi(\mathbf{P}_{ci}\mathbf{P}_c^t \mathbf{U}_k)) \qquad (4)$$

with the addition of camera index c and i, where c serves as the reference. The first term is for the reference camera, and the second term is for all other cameras.

Comparing the second term in Eq. (4) to Eq. (3), we notice that they are essentially similar in that $\Delta\mathbf{P}_{ri}^t$ is the difference from the reference frame to the current frame, and $\mathbf{P}_{ci}$ is the transform from the reference camera to the current camera. The only difference is that $\Delta\mathbf{P}_{ri}^t$ changes over time and $\mathbf{P}_{ci}$ addresses the static spatial transform since cameras are fixed most of the time. In the next section, we will generalize (4) within a Bayesian framework with certain constraints on $\mathbf{P}_{ci}$.

2.3. Bayesian Framework

We consider head poses at one time instant as a state in a dynamic system. Let $\mathbf{X}^t$ denote the poses to estimate from all the cameras, i.e., $\mathbf{X}^t = \{\mathbf{P}_i^t | i = 1,\ldots,N\}$. Let $\mathbf{X}^{t-} = \{\mathbf{X}^{t-n} | n = 1,\ldots,\tau\}$ represent the poses of the previous frames, and $\mathbf{Y}^t$ be the poses of keyframes up to time t. The observation input to the system is the image sequences from the cameras, denoted by $\mathbf{I}^t$. Extending the Bayesian framework for single-camera pose tracking [17], we conduct multi-camera pose tracking by maximizing the posterior probability $P(\mathbf{X}^t|\mathbf{X}^{t-}, \mathbf{Y}^t, \mathbf{I}^t)$. Since keyframes are retrieved through a re-detection process independently from the main tracking thread, we further simplify the problem to

$$P(\mathbf{X}^t|\mathbf{X}^{t-}, \mathbf{Y}^t, \mathbf{I}^t) \propto P(\mathbf{X}^t|\mathbf{X}^{t-}, \mathbf{I}^t) P(\mathbf{X}^t|\mathbf{Y}^t, \mathbf{I}^t) \quad (5)$$

Assuming each $P(.)$ is a Gaussian distribution and each feature point matching is independent, we derive the following equation for 2D-3D correspondences from the reference camera c using the reference frame r to

$$\max P(\mathbf{X}^t|\mathbf{Z}_r^t, \mathbf{I}^t) \propto \min \sum_k (\mathbf{u}_{c,k}^t - \hat{\mathbf{u}}_{c,r,k}^t)^T \Sigma_{c,r,k}^{-1} (\mathbf{u}_{c,k}^t - \hat{\mathbf{u}}_{c,r,k}^t) \qquad (6)$$

where $\mathbf{Z}_r^t$ denotes the pose of a generic reference frame at time t, $\Sigma_{c,r,k}$ is the covariance matrix for feature estimation at point k with reference frame r from camera c, and $\hat{\mathbf{u}}_{c,r,k}^t = \phi(\Delta\mathbf{P}_{rc}^t \mathbf{P}_r^t \mathbf{U}_k)$ as defined similarly in Eq. (3). This essentially addresses the first term of Eq. (4). The second term follows the similar fashion except that the mean becomes $\hat{\mathbf{u}}_{i,c,k}^t = \phi(\mathbf{P}_{ci}\mathbf{P}_c^t \mathbf{U}_k)$ for camera i and $i \neq c$.

Meanwhile, we also estimate the relative geometry $\mathbf{P}_{ij}$ between camera i and j for the benefits of leveraging tracking from different cameras. Although we could use $\mathbf{P}_{ij} = \mathbf{P}_j \mathbf{P}_i^{-1}$ to obtain the relative pose at each time instant, this could result in unwanted divergence of the invariant $\mathbf{P}_{ij}$. Recall that the cameras remain fixed, we add a smoothness term $||\mathbf{P}_{ij} - \mathbf{P}_{ij}^{old}||_M^2$ to Eq. (6), where $\mathbf{P}_{ij}^{old}$ is the estimate from the last instance or the initial guess, and $||\cdot||_M$ denotes the Mahalanobis distance given by

$$||\mathbf{P}_{ij} - \mathbf{P}_{ij}^{old}||_M^2 = (\mathbf{P}_{ij} - \mathbf{P}_{ij}^{old})^T \Lambda_{ij}^{-1} (\mathbf{P}_{ij} - \mathbf{P}_{ij}^{old}). \quad (7)$$

where the covariance matrix of Λ_{ij} controls the adaptation of relative geometry between camera i and j learnt during optimization. In the end, our total cost for minimization is to combine (4), (6) and (7) with associate weights based on aging factor for previous frames or robustness of the feature matching [19].

$$
\begin{aligned}
e^t = & \sum_r \omega_r \sum_k (\mathbf{u}^t_{c,k} - \hat{\mathbf{u}}^t_{c,r,k})^T \Sigma^{-1}_{c,r,k} (\mathbf{u}^t_{c,k} - \hat{\mathbf{u}}^t_{c,r,k}) \\
& + \sum_{i,i\neq c} \omega_i \sum_k (\mathbf{u}^t_{i,k} - \hat{\mathbf{u}}^t_{i,c,k})^T \Sigma^{-1}_{i,c,k} (\mathbf{u}^t_{i,k} - \hat{\mathbf{u}}^t_{i,c,k}) \\
& + \sum_{i,j,i\neq j} \omega_{ij} ||\mathbf{P}_{ij} - \mathbf{P}^{old}_{ij}||^2_M \qquad (8)
\end{aligned}
$$

3. Tracking Algorithm

The main steps of the tracking algorithm is highlighted in this section. Given all the reference frames their associated poses, we select features on the face from the Harris detector for tracking from each camera. The KLT algorithm [9] is used to track the features onto the frame at time t. This provides a set of 2D-3D correspondences at each view temporally. To make the algorithm less prone to error drift, we retain keyframes accumulated over time for additional matches. Both key and previously tracked frames serve as reference for establishing 2D-3D correspondences. Finally, the smoothness constraints for camera geometry is added to the cost function (8) for optimization.

3.1. Initialization and Feature Point Matching

Assume that we know the intrinsic parameters of each camera by calibration or supplied from the manufacture. The assumption that the intrinsic parameters do not change over time is reasonable since there is not much need of PTZ cameras for a user sitting in front of the desktop. Tracking from each camera starts with face detection [18] which identifies a face rectangle. Then we run 2D mesh alignment [7] to get 2D feature points for initial tracking. Basically, the 3D model is oriented and projected to the individual camera coordinate in the face region so that we obtain an initial guess of the camera geometry relative to world coordinate. At each camera, we select point features over the face region based on the KLT selection criterion [13] and back-project these points onto the mesh model to obtain the 3D point locations on the model denoted as $\mathbf{U}^r_k$ for camera r at point k. Moving on to the next time instant, we use the KLT feature point tracking algorithm [9] to track the selected 2D features and obtain the features on the new frames $\mathbf{u}^t_{r,k}$. Using the Perspective-n-Point (PnP) formulation, a rigid pose is determined by minimizing the sum of projection errors given multiple 2D-3D corresponding points. The use of POSIT algorithm [3] allows us for a quick coarse pose estimation and inlier verification to achieve real-time processing. Most of the above procedures are depicted in Figure 3. The advantage of the above algorithm is that it is extremely fast, allowing for real time solutions even in the multi-camera scenario. However, feature point trackers are susceptible to drift. If the error is uncorrected, it leads to the divergence of the tracking algorithm as the 2D-3D correspondence is no longer valid. In order to make the algorithm robust to drift, we expand the selection of reference tracking to include keyframes.

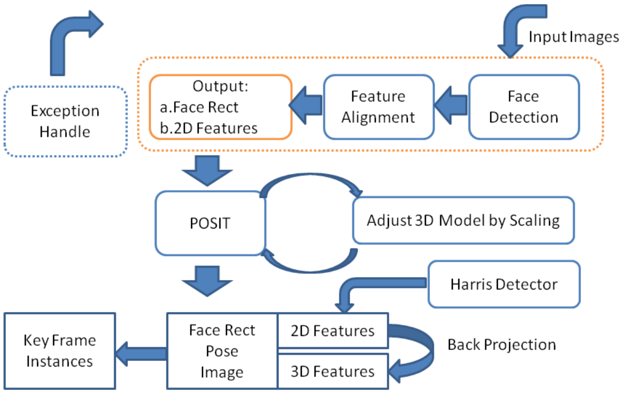


Figure 3. Tracking initialization and key frame insertion.

3.2. Keyframe Addition as Reference

Our keyframe comprises of an image and the pose of the head observed in the image. This provides an absolute reference to restrict drift. Over the time of tracking, we maintain a set of keyframes extracted from an independent initialization thread similar to the process in Figure 3. We then proceed to obtain 2D-3D correspondences between the current image and the keyframe. Keyframes are considered as one type of reference frames which are always available for matching with the current frame, as emphasized in (3).

Since the pose of the reference frame could differ significantly from the query pose (i.e., the latest pose), we warp the face from its pose in the reference frame to that of the query. This is done by aligning the feature points in 2D image from reference pose to the 3D face model, and rendering it onto the image based on the query pose. As a result, the warped face image and head pose from the reference frame are expected to be in a reasonable proximity of the current image and pose. Referencing on both keyframes and previous frames corrects drift during tracking and allow us to apply the same simple feature point correspondence between the 3D model and its 2D projection. Thus the POSIT algorithm can be applied again for fast pose estimation.

Ideally, the keyframe storage should contain a variety of poses viewed from all cameras with high confidence of tracking, along with its corresponding images. This pool is common to all cameras, and shared across views. However, using reference frames across camera views can be affected adversely by illumination effects. It is important to keep this in mind, and use suitable illumination normalization in matching feature points, or select appropriate keyframes under similar illumination condition. Also, a frame is added as a new keyframe only if the pose is sufficiently far away from all existing ones to ensure the diversity of key poses to achieve better tracking at various poses. In video conferencing applications, the poses close to frontal form an important type of keyframes [15] as a user most likely talks directly to the cameras. To handle such preferred poses, we add the initializing frame for each camera as a keyframe when the face is detected and aligned.

3.3. Joint Pose Estimation via Optimization

Pose estimation includes both the common pose of the head $\mathbf{P}^t$ in world coordinate and the relative poses $\mathbf{P}_{ij}$ between camera i and j $(i, j = 1, \ldots, N)$. They are estimated together by minimizing (8). In our implementation, the Levenburg-Marquardt (LM) technique is used. Besides the pose estimates, their uncertainty (covariance matrix) is also computed. More technical details are provided in the following sections.

Pose From Each Camera: For each camera j where tracking is successful, we use the 2D-3D correspondence $(\mathbf{u}^t_{j,k}, \mathbf{U}_k)$ between the current image and all possible reference frames, apply the RANSAC robust estimation technique with the POSIT algorithm to discard outliers, and get an approximate pose estimation $\mathbf{P}^t_j$ with the remaining good correspondences. Among all cameras being tracked, we choose the one with the highest matching fidelity, which is usually the one closer to the frontal pose, as a reference camera c. Then we use $\mathbf{P}^t_j = \mathbf{P}_{cj}\mathbf{P}^t_c$ to estimate poses from all other cameras.

Relative Geometry: As described above, relative geometry $\mathbf{P}_{ij}$ between camera i and j is needed during the joint pose estimation. The initial estimate of $\mathbf{P}_{ij}$ could be either the estimation at the previous instance if tracked, or calculated as

$$\mathbf{P}_{ij} = \begin{cases} \mathbf{P}_j\mathbf{P}_i^{-1} & \text{if } i = r \\ \mathbf{P}_{cj}\mathbf{P}_{ci}^{-1} = \mathbf{P}_{cj}\mathbf{P}_{ic} & \text{if } i \neq r \end{cases} \tag{9}$$

If $\mathbf{P}_{ij}$ is never initialized, its corresponding covariance matrix is set to a very large value, indicating poor knowledge of the relative geometry between camera i and j. Since the cameras are static, the incorporation of the Mahalanobis distance $||\mathbf{P}_{ij} - \mathbf{P}^{old}_{ij}||^2_M$ from Eq. (7), imposes a constraint that the relative camera geometry $\mathbf{P}_{ij}$ at time t is expected not to be too different from the previous estimate $\mathbf{P}^{old}_{ij}$ if multi-camera tracking is getting accurate.

Dynamic Switching of the Reference Camera: Note that we cannot pre-select a camera as reference because there is no guarantee that the face will be viewed clearly from this particular camera during the course of tracking. The advantage of using multiple cameras is that at least one of the cameras in the system will most likely have a good view of the face so that the pose estimation for other views could leverage from that. Thus, we usually select a camera close to frontal view of the face and allow dynamic switching of the reference camera for better tracking result.

LM Optimization: At the end, let the reference camera be c, and we group all the parameters to be determined as the union of $\mathbf{P}^t_c$ and $\mathbf{P}_{cj}$ $(j = 1, \ldots, N; j \neq c)$ and apply the Levenburg-Marquardt (LM) method to minimize the overall cost in Eq. (8). Once $\mathbf{P}^t_c$ and $\mathbf{P}_{cj}$ are solved, we can easily estimate the pose for a camera which has lost tracking due to occlusion or poor angle of the face using the pose from reference camera.

Due to dynamic switching of the reference camera, the covariance matrix $\mathbf{\Lambda}_{ij}$ is directly updated during optimization only if camera i or j is selected as reference. If we need to update $\mathbf{\Lambda}_{ij}$ when neither camera i or j is selected as a reference camera, we use the first order Taylor expansion and compute it with the following formula

$$\begin{aligned}\mathbf{\Lambda}_{ij} = {} & \frac{\partial(\mathbf{P}_{cj}\mathbf{P}_{ic})}{\partial\mathbf{P}_{ic}}\mathbf{\Lambda}_{ic}[\frac{\partial(\mathbf{P}_{cj}\mathbf{P}_{ic})}{\partial\mathbf{P}_{ic}}]^T \\ & + \frac{\partial(\mathbf{P}_{cj}\mathbf{P}_{ic})}{\partial\mathbf{P}_{cj}}\mathbf{\Lambda}_{cj}[\frac{\partial(\mathbf{P}_{cj}\mathbf{P}_{ic})}{\partial\mathbf{P}_{cj}}]^T \end{aligned} \tag{10}$$

where partial derivative $\partial(.)$ is approximated using the difference between the estimated poses at time t and $t-1$.

3.4. Tracking Process

The overall tracking process is illustrated in pesudo code in the algorithm shown in the next page.

4. Experimental Results

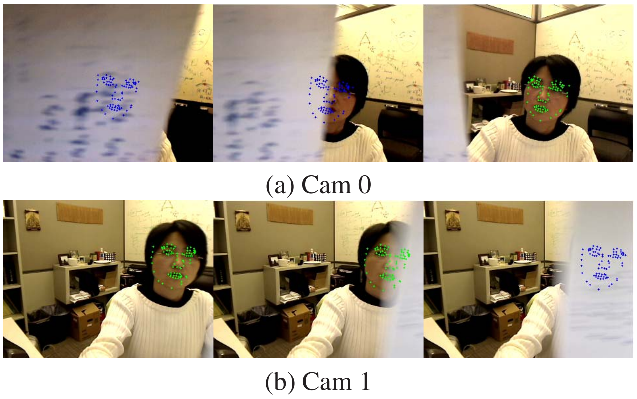
We implemented our tracking framework in a system with a dual core 3 GHz processor PC and multiple Logitech VX9000 web-cameras with image size of 320x240. The generic face model we use for face alignment is based on [5]'s face model with 83 landmarks. Our head pose tracking system runs at an average frame rate of 15 fps when three cameras are involved. We demonstrated our tracking algorithm with tens of subjects in various environments such as indoor offices, conference rooms and busy demo floors for big events with satisfying results, i.e., subjects saw the 3D head model points overlaid on the video sequence with face landmarks closely matched. Due to space constraints, we only show some examples of tracking results in Figure 4, 5 and 6.

To compare the tracking result to the ground truth, we captured a one-minute long sequence from three cameras on a subject wearing a hexagonal crown (see Figure 5) at 30 frames per second. The ground truth is estimated by calibration at each view independently. The tracking results ob-

```
Algorithm: Multi-Camera 3D Head Pose Tracking
Input: images at time $t$, $\mathbf{I}^t$, the previous frames $\mathbf{X}^{t-n}$
    ($n = 1, \ldots, \tau$), and the keyframe set $\mathbf{Y}^t$ from
    all cameras
Output: Refined pose $\mathbf{P}^t$
1. Select reference frame $\mathbf{P}_r^{t-1}$ from each camera.
2. Initialize the relative pose $\mathbf{P}_{ci}$ to $\mathbf{P}_{ci}^{old}$. If it is never
initialized, use Eq. (9) with a large uncertainty.
foreach camera $i$ do
    foreach frame $\mathbf{Z}_r^t \in \mathbf{X}^{t-n} \cup \mathbf{Y}^t$ do
        3. Find feature matches $(\hat{\mathbf{u}}, \mathbf{u})$ between image
        $I_i^t$ and the image $I_r^t$ and back-project features
        $(\mathbf{u})$ in $I_r^t$ onto the 3D model to get $(\mathbf{U})$.
        4. Discard outliers by applying RANSAC.
    end
    5. Compute an approximate pose estimate by
    applying POSIT to the inlier feature subset.
end
6. Select a new reference camera $c$ if necessary.
7. Minimizing the cost function (8) using LM
algorithm to optimize the pose from the reference
camera $\mathbf{P}_c^t$ and relative poses $\mathbf{P}_{ci}$.
8. Update the common pose and relative poses.
```

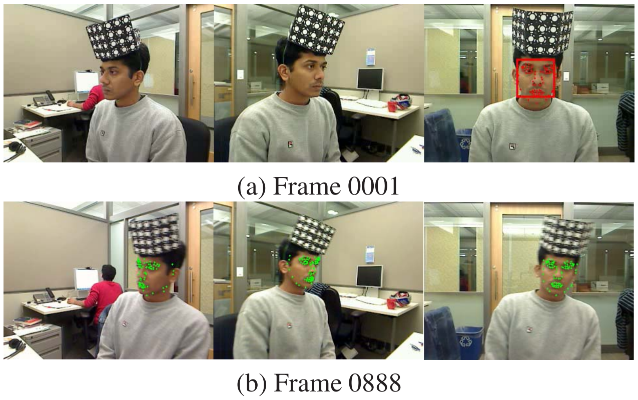
tained from the multi-camera tracker was matched against the ground truth allowing for a scaled Euclidean transformation between the two sets of estimates. Figure 6 shows the tracking results overlaid on ground truth estimates for both the rotation (in terms of three Euler angles: roll in red , pitch in green and yaw in blue) and translation (x, y and z components in red, green, and blue, respectively). Note that the mismatch between ground truth and the tracker occurs only at extreme translations where even the ground truth might be incorrect.

Figure 7 shows the comparison results of estimated camera relative pose and the ground truth of essential matrices between cameras using the same data as in Figure 6. Since the three cameras are static, we should expect flat lines for both the rotation and translation in 3D. We notice that the


(a) Cam 0

(b) Cam 1

Figure 4. A sub-sampled tracking sequence from two cameras which leverage each other during occlusion.


(a) Frame 0001

(b) Frame 0888

Figure 5. Multi-camera tracking results with three cameras. The hexagonal cap was used to obtain ground truth for the head pose as well as the camera placement.

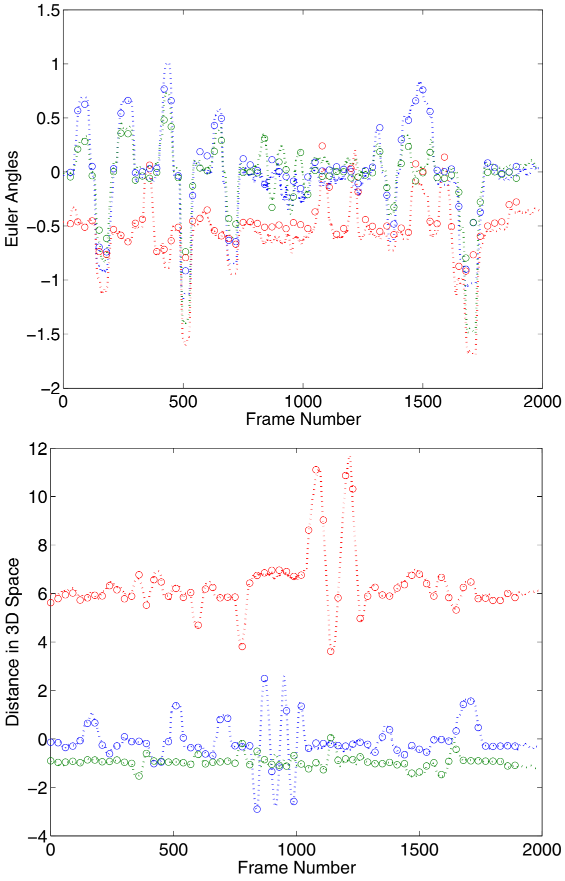


Figure 6. Comparison of head pose from multi-camera pose tracking (shown in dotted line) against manually labeled ground truth (shown in circles). They coincide well as the dotted lines mostly pass through the circles. The results were obtained over the video showed in Figure 5.

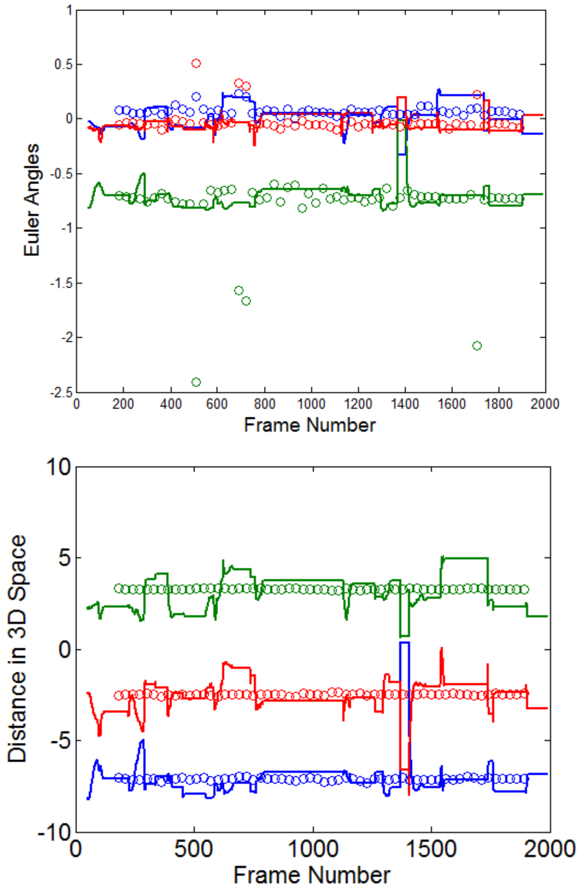


Figure 7. Comparison of relative geometry between camera 1 and 2 from multi-camera pose tracking (shown in dotted line) against manually labeled ground truth (shown in circles).

estimated rotation angles are even better than the ground truth in certain instances when there is clearly outlier during calibration.

To compare with the monocular pose tracking, we tested another half minute long video with occlusions captured from three cameras at 30 fps. Figure 8 shows a time instant with severe occlusions from two of the three views. In this case, single camera tracking fails in the two views, but multi-camera tracking is still able to track from all three views by leveraging the reference camera with little occlusion. The plot in the figure shows the estimated 3D translation of the head with single-camera (thicker line) and multi-camera tracking (thin line). The flat thicker line shows the instances when single camera tracking fails, while the multi-camera tracking still succeeds.

Further comparison on tracking between a single camera and two-camera system is shown in Figure 9 using another half minute long video sequence shown in Figure 4, where the interframe differences are computed for rotation and translation. For illustration purpose, we use a pre-selected value $\pi/2$ for rotation and 10.0 for translation when loss of tracking happens. Note that the single camera tracking failed quite a number of times.

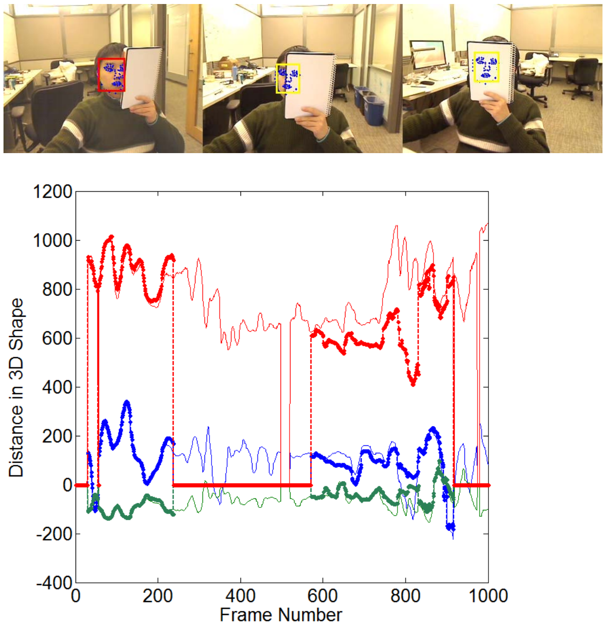


Figure 8. A comparison of single-camera and multi-camera tracking in a severe occlusion case. The Red, Green and Blue thick lines represent the x, y and z components of the head translation estimated by the single-camera tracking, while those thin lines, by the multi-camera tracking.

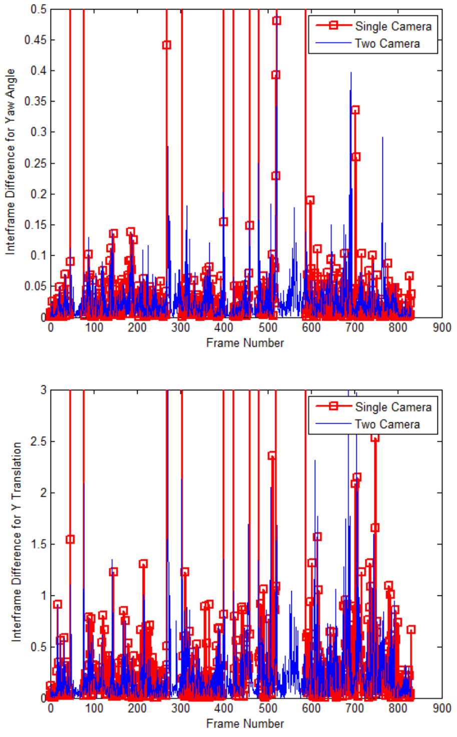


Figure 9. Performance Comparison on tracking using one camera and two cameras.

Comparison of tracking performance between single and multi-camera are data dependent. We could choose a sequence of data with frequent occlusions to favor the multi-camera tracking. On the other hand, if cameras are placed close to each other and faces are mostly frontal, we shall not see much difference in performance between the two. The following table shows the performance evaluation of using a single camera, two cameras and three cameras, compared to a three camera tracking system with pre-calibrated camera geometry, using the image sequence described in Figure 5. We intentionally avoid occlusion in the video for a fair comparison. We tested our tracking algorithm by assuming there are one, two or all three cameras in the system respectively. The rotation error is measured in angular degrees, and translation is in millimeters. As we expected, the average re-projection decreases with the increase of the number of cameras. We also notice that the tracking performance is almost equivalent compared to the case where camera placements are pre-calibrated (shown in the last row of the table). We also notice that single camera tracking has larger mean rotation error in degree. This is mainly due to loss of tracking for profile poses where we just use the frontal pose as default. On the other hand, multi-camera tracking is able to estimate a pose most of the time using a reference camera dynamically, resulting in smaller mean errors.

# of Cams	Median Ang err	Median Tran err	Mean Ang err	Mean Tran err
1	5.9754	0.4138	19.0132	0.5504
2	5.4878	0.3481	11.8039	0.4885
3	4.4671	0.3404	11.0915	0.4634
3 (calib)	4.0827	0.3132	11.0577	0.4540

5. Conclusion

In this paper, we described a real time head pose tracking framework with multiple cameras and unknown placement parameters. The proposed algorithm achieves real time tracking requirements as well as the ability to estimate and refine the camera placement parameters. Furthermore, by using generalized reference frames in both temporal and spatial directions, the algorithm remains drift free within a large pose range. Future work in this regard involves adapting the face model parameters to customize the 3D model to the individual being tracked. Also of interest is multiple object tracking that could make cross-camera correspondences more robust.

References

[1] G. Aggarwal, A. Veeraraghavan, and R. Chellappa. 3D Facial Pose Tracking in Uncalibrated Videos. *Lecture Notes in Computer Science*, 3776:515, 2005.

[2] S. O. Ba and J.-M. Odobez. Probabilistic head pose tracking evaluation in single and multiple camera setups. In *Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR 2007 and RT 2007*, pages 276–286. Springer-Verlag, Berlin, Heidelberg, 2008.

[3] D. Dementhon and L. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1):123–141, 1995.

[4] P. Fua. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. *International Journal of Computer Vision*, 38(2):153–171, 2000.

[5] F. Jiao, S. Li, H.-Y. Shum, and D. Schuurmans. Face alignment using statistical models and wavelet features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:321, 2003.

[6] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, October 2005.

[7] L. Liang, R. Xiao, F. Wen, and J. Sun. Face alignment via component-based discriminative search, 2008. ECCV (2).

[8] T. D. Louis-Philippe Morency, Ali Rahimi. Adaptive view based Appearance Model. In *Computer Vision and Pattern Recognition, 2003. Proceedings CVPR'03, 1996 IEEE Computer Society Conference on*, volume 1, 2003.

[9] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (ijcai). In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, April 1981.

[10] H. Moon, R. Chellappa, and A. Rosenfeld. 3d object tracking using shape-encoded particle propagation. In *IEEE International Conference on Computer Vision*, 2001.

[11] J. Ng and S. Gong. Multi-view face detection and pose estimation using a composite support vector machine across the view sphere. In *Proc. Int. Workshop Recogn., Anal., Tracking Faces Gestures Real-Time Syst*, pages 14–21, 1999.

[12] R. Ruddarraju and I. A. Essa. Fast multiple camera head pose tracking. In *In Proceedings, Vision Interface 2003*, 2003.

[13] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[14] S. Tariq and F. Dellaert. A Multi-Camera 6-DOF Pose Tracker. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, volume 3, 2004.

[15] L. Vacchetti, V. Lepetit, and P. Fua. Stable Real-Time 3D Tracking Using Online and Offline Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1385–1391, 2004.

[16] M. Voit and R. Stiefelhagen. Tracking head pose and focus of attention with multiple far-field cameras. In *Proceedings of the 8th international conference on Multimodal interfaces*, pages 281–286. ACM New York, NY, USA, 2006.

[17] Q. Wang, W. Zhang, X. Tang, and H. Shum. Real-Time Bayesian 3-D Pose Tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(12):1533, 2006.

[18] R. Xiao, L. Zhu, and H. Zhang. Boosting Chain Learning for Object Detection. In *Proc. ICCV*, volume 1, pages 709–715, 2003.

[19] Z. Zhang. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.