Full-Motion Recovery from Multiple Video Cameras Applied to Face Tracking and Recognition

Josh Harguess, Changbo Hu, J. K. Aggarwal
Computer & Vision Research Center / Department of ECE
The University of Texas at Austin
harguess@utexas.edu, changbo.hu@gmail.com, aggarwaljk@mail.utexas.edu

Abstract

Robust object tracking still remains a difficult problem in computer vision research and surveillance applications. One promising development in this area is the increased availability of surveillance cameras with overlapping views. Intuitively, these overlapping views may lead to more robust object tracking and recognition. However, combining the information from the multiple cameras in a meaningful way is challenging. Our contribution in this work is a novel approach to object tracking by robustly and accurately recovering the full motion of the object from multiple cameras. This is accomplished by explicitly fusing the information from multiple cameras into a joint 3D motion calculation. We apply this approach to the tracking of faces in multiple video cameras and utilize the 3D cylinder model to realize the motion calculation. The method is demonstrated on a sequence of real data for pose estimation of the face. Also, the 3D cylinder texture map from the tracking result is used in face recognition. The performance of full-motion recovery from multiple cameras is shown to produce a significant increase in the accuracy of face pose estimation and results in a higher face recognition rate than from a single camera. Our approach may be applied to other types of object tracking such as vehicles.

1. Introduction

The full-motion recovery of an object for object tracking remains a persistent and difficult problem in computer vision search and surveillance applications. The full motion of a rigid object is described by three rotations and three translations in a three dimensional (3D) space and is essential for many computer vision tasks such as human computer interaction and visual surveillance. Fortunately, it is common to have multiple cameras available in today's many computer vision applications. Therefore, we may be able to utilize the information present from multiple cameras in a surveillance scene to enhance object tracking and recognition. However, it is not abundantly clear how this is accomplished. Our contribution is a novel solution to object tracking from multiple video cameras by explicitly fusing the information from the cameras into a joint 3D motion calculation of the object.

One important class of object tracking research is face (or head) tracking. Most face recognition and facial expression analysis methods require the motion of faces to be known so that faces may be aligned, implicitly or explicitly. One application of face (and object) tracking is pose estimation, which is a very large and diverse research topic, as depicted in [19]. These methods may be classified as feature-based approaches or model-based approaches. However, only a small portion of these algorithms recover the 3D face motion. The authors in [14, 16, 17] track image features to recover 3D poses. Feature-based approaches are flexible but performance depends heavily on the availability of good features. Since human heads are similarly shaped, many model-based approaches have been proposed to take advantage of this similarity. In [1, 6, 7, 20, 22], the head is treated as a cylinder and the head motion is recovered using a cylinder model. Some researchers apply ellipsoid models instead [2, 8, 9]. Model-based approaches are more reliable considering that the motion recovery is obtained from the whole face region. Yet, the whole face region must be visible to perform the motion calculation. Therefore, large nonfrontal poses of the face still challenge the effectiveness of the model-based method. However, by utilizing multiple surveillance cameras with overlapping views, we may improve the model-based approach by fusing the motion from the multiple views into a joint 3D motion estimation.

Previous researchers in multi-camera face tracking have used stereo information to increase tracking robustness, such as in [15]. In contrast to stereo, we desire the integration of multiple camera information to obtain as many observations as possible to recover the head motion. The robustness and accuracy of head motion tracking may be increased by integrating the motion information from mul-

tiple cameras. Additionally, an individual camera may recover the pose correctly if the tracking is lost in that camera's view with assistance from other cameras in a multiple camera setting. However, how does one combine the motion from multiple cameras effectively to improve the face tracking result? We present a complete derivation for explicitly fusing the motion from multiple cameras into a joint 3D (three rotations and three translations) motion estimation of the face. This work is a continuation of our previous work in multi-camera face tracking [12] and multi-camera face recognition [11]. In our previous multi-camera face tracking work [12], the focus is on detecting self-occlusion and camera occlusion in a multi-camera system with an application to face tracking and pose estimation of the face. In our previous multi-camera face recognition work [11], we fuse the recognition results from multiple independent cameras to improve face recognition performance. The focus of the work in this paper is to apply the full-motion recovery from multiple cameras model to face recognition.

To realize this design, we employ a cylinder head model. A cylinder model is generic and easily adapted to a specific person, while the relative error between the cylinder model and the real geometry of a face is small [6]. The head motion recovery is driven by the observations from multiple views. In this sense, our approach shares same reasoning of [13], which uses 3D active appearance model and is more person specific. However, we have chosen to use a more generic model so that it is easy to initialize while still remaining a close approximation to the head. Our work is most similar to that of Cai *et al*. [5], where they propose a method that integrates feature tracking of the face from multiple cameras by adapting a generic head model. In addition to using a more generic model that is easier to initialize, our method also produces texture of the face that is appropriate for face recognition. Feature-based tracking relies on features of the face being present in every frame, whereas our use of model-based tracking does not have this limitation.

We demonstrate the proposed approach using video sequences with ground truth of 3D head motion from real data. Comparing the motion recovery from multiple cameras with that of a single camera, we show that the accuracy of pose estimation has been significantly increased. We also find that the motion-free texture of the face generated from the cylinder model with the multiple camera tracking produces higher recognition rates compared with the single camera case. We utilize a system to perform face recognition from video based on our earlier work in this area [11].

The remaining sections are organized as follows. Section 2 outlines the full-motion recovery from multiple cameras method in detail. In Section 3 we present experimental results on face pose estimation and face recognition. A discussion of our results is presented in Section 4 and we conclude the paper in Section 5.

2. Full-Motion Recovery from Multiple Cameras

In this section we introduce the multiple camera full-motion recovery model. Without loss of generality, we specify one of the cameras as the world coordinates for our system. We call this camera the "first" camera. This is done because the derivation for the first camera motion is unique from the derivation of the other cameras in the system, since the motion of the 3D rigid object will be calculated in the first camera's view. Incidentally, the derivations for a single camera motion and the first camera in the multiple camera motion are essentially the same. The derivations for the first camera and the extension to multiple cameras follows.

2.1. First Camera Motion

In a model-based head tracking approach, a basic assumption is made that the head (and thus the face) may be treated as a 3D rigid object. Therefore, only six parameters (three translations and three rotations) are needed to describe the motion performed by the head. The motion of the 3D points w.r.t. the first camera coordinates may then be described as

$$\boldsymbol{X}(t+1) = \boldsymbol{M} * \boldsymbol{X}(t) \tag{1}$$

$$\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{\mu}) = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{T}, \tag{2}$$

where $\boldsymbol{X}(t+1)$ are the new coordinates of the 3D points $\boldsymbol{X}(t)$ after a motion of $\boldsymbol{M}$ has been applied where $\boldsymbol{\mu}$ is a six element vector representing rigid motion, including 3D rotation (w_x, w_y, w_z) and translation (t_x, t_y, t_z), $\boldsymbol{x} = (x, y, z)^T$ is a 3D coordinate of a point on the surface of the object, $\boldsymbol{M}$ is the function of the rigid transformation, $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{T}$ is the translation vector. We denote the rigid motion of the head from time t to time $t+1$ as $\Delta\boldsymbol{\mu}$. If $\boldsymbol{p}_t = (u, v)$ is the projection point in the image plane $\boldsymbol{I_t}$ of point $\boldsymbol{x}$ on the 3D object , then the new location of point $\boldsymbol{p}_{t+1}$ in the next frame $\boldsymbol{I}_{t+1}$ is estimated as

$$\boldsymbol{p}_{t+1} = \boldsymbol{F}(\boldsymbol{p}_t, \Delta\boldsymbol{\mu}). \tag{3}$$

The next image frame may then be computed by

$$\boldsymbol{I}_{t+1}(\boldsymbol{F}(\boldsymbol{p_t}, \Delta\boldsymbol{\mu})) = \boldsymbol{I}_t(\boldsymbol{p}_t), \tag{4}$$

where $\boldsymbol{F}$ is the 2D parametric motion function of $\boldsymbol{p_t}$. A necessary and reasonable assumption is made that the illumination does not change and that movement is small between frames, so the pixel intensities between the two frames are consistent.

To compute the change in rigid motion vector $\Delta\boldsymbol{\mu}$, the error between two successive image frames is minimized. This is solved by using the Lucas-Kanade image alignment algorithm [18]. The result is

$$\Delta\boldsymbol{\mu} = -(\sum_{\Omega} \boldsymbol{G}^T\boldsymbol{G})^{-1} \sum_{\Omega} \boldsymbol{I_t}\boldsymbol{G} \tag{5}$$

where

$$\boldsymbol{G} = \boldsymbol{I_p}\boldsymbol{F_\mu} \tag{6}$$

and where $\boldsymbol{\Omega}$ is the region of overlapping pixels between the two frames, $\boldsymbol{F_\mu}$ is the partial derivative of $\boldsymbol{F}$ w.r.t. the rigid motion vector, and $\boldsymbol{I_p}$ and $\boldsymbol{I_t}$ are the spatial and temporal image gradients, respectively.

Then, under the assumption that the perspective projection only depends on the focal length, then the derivative of $\boldsymbol{F}$ w.r.t. the rigid motion vector at $\boldsymbol{\mu} = 0$ is [22] Assuming that the camera projection matrix depends only on the focal length,

$$\boldsymbol{F_\mu}\Big|_{\Delta\boldsymbol{\mu}=0} = \begin{bmatrix} -xy & x^2+z^2 & -yz & z & 0 & -x \\ -(y^2+z^2) & xy & xz & 0 & z & -y \end{bmatrix} \frac{f}{z^2}, \tag{7}$$

where x, y and z are the 3D coordinates of the object and f is the focal length of the camera. For single camera tracking, the rigid head motion vector $\Delta\boldsymbol{\mu}$ is recovered by substituting the result of (7) into equation (6). Then, using Rodrigues' transformation formula, $\boldsymbol{M}$ is calculated from $\Delta\boldsymbol{\mu}$ and applied in equation (1) to recover $\boldsymbol{X}$ at time $t+1$.

2.2. Multiple Camera Motion

The most natural way to extend the full-motion recovery model to multiple cameras is to allow each camera to track the face independently. However, if any of the independent cameras lose track of the face due to large nonfrontal poses of the face, it may not be able recover the track and pose of the face. Also, by combining the motion information from multiple cameras simultaneously, we may improve the robustness of the overall motion estimation of the face. Since each of the cameras are viewing the face of the same individual, the 3D motion of the face must be the same w.r.t. the world coordinates. Therefore, the motion that is described in each of the cameras may be used to estimate the motion of the face more accurately and precisely. We take advantage of this observation by calculating a joint change in motion from the cameras.

Using a similar notation and methodology as that explained in the first camera section, we recover the full motion of the face from multiple cameras in the following manner. Please note that the following derivation could be applied to any number of cameras that refer back to the first camera as the world coordinate system. In the first camera's view, we have

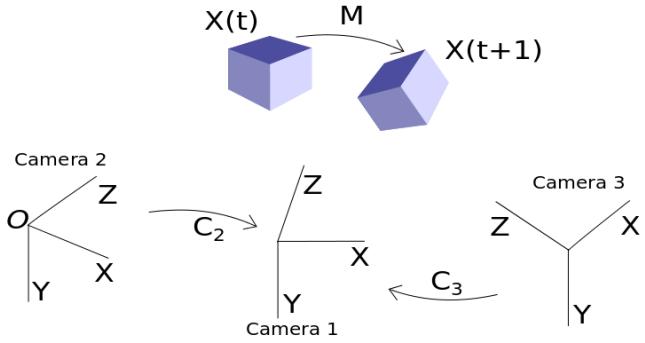


Figure 1. Example of a Three Camera System

$$\boldsymbol{p}_1(t+1) = \boldsymbol{K}_1 * \boldsymbol{X}_1(t+1) = \boldsymbol{K}_1 * \boldsymbol{M} * \boldsymbol{X}_1(t) \tag{8}$$

where $\boldsymbol{p}_1(t+1)$ is the projection of those 3D points to the first camera's image plane obtained through the multiplication with intrinsic camera matrix $\boldsymbol{K}_1$. Recall that the motion of the 3D object in the first camera's view is shown in equation (1). To relate the ith camera with the first camera,

$$\boldsymbol{X}_1(t) = \boldsymbol{C}_i * \boldsymbol{X}_i(t), \tag{9}$$

where

$$\boldsymbol{C}_i = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \tau_x \\ r_{21} & r_{22} & r_{23} & \tau_y \\ r_{31} & r_{32} & r_{33} & \tau_z \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \tag{10}$$

is the 4×4 matrix representing the rotation and translation between the ith camera's coordinate system and the world coordinate system (which is the first camera in our case), and $\boldsymbol{X}_i(t)$ are the 3D points w.r.t. the ith camera's coordinate system.

In the ith camera's view

$$\boldsymbol{X}_i(t+1) = \boldsymbol{C}_i^{-1} * \boldsymbol{X}_1(t+1) = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{X}_1(t) \tag{11}$$

$$\boldsymbol{X}_i(t+1) = \boldsymbol{M}_i * \boldsymbol{X}_i(t) \tag{12}$$

$$\boldsymbol{p}_i(t+1) = \boldsymbol{K}_i * \boldsymbol{M}_i * \boldsymbol{X}_i(t) \tag{13}$$

where $\boldsymbol{p}_i(t)$ are the image coordinates in the ith camera's view after a projection with camera matrix $\boldsymbol{K}_i$. A simulated environment with three cameras viewing the motion ($\boldsymbol{M}$) of a cube is shown in Figure 1.

Full motion recovery from multiple cameras is accomplished by relating the motion in each camera's view back to the motion in the first camera, $\boldsymbol{M}$. This may be done by using equations (11) and (12) in the following manner.

$$\boldsymbol{X}_i(t+1) = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{X}_1(t) = \boldsymbol{M}_i * \boldsymbol{X}_i(t). \quad (14)$$

Therefore,

$$\boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{X}_1(t) = \boldsymbol{M}_i * \boldsymbol{C}_i^{-1} * \boldsymbol{X}_1(t) \quad (15)$$

$$\boldsymbol{M}_i = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{C}_i \quad (16)$$

and we may rewrite the equation for motion of the ith camera as

$$\boldsymbol{X}_i(t+1) = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{C}_i * \boldsymbol{X}_i(t). \quad (17)$$

The reason the motion of the 3D points in the ith camera's view are represented this way is to exploit the idea that the motion of the 3D points is the same between multiple views of the moving object. We may now explicitly solve for the full-motion recovery of the face in both camera views and compute $\boldsymbol{\Delta\mu}$ from the information present in all cameras.

The crucial difference between calculating the motion in the ith camera's view as compared to the first camera's view is that the rotation and translation between the two cameras' coordinate systems (equation (9)) must be taken into account. Approximating the motion $\boldsymbol{M}$ by the twist representation [4] we have

$$\boldsymbol{M} = \begin{bmatrix} 1 & -w_z & w_y & t_x \\ w_z & 1 & -w_x & t_y \\ -w_y & w_x & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (18)$$

which may also be thought of as a small angle approximation to the real rotation and translation. In the following equations, entries of the inverse matrix $\boldsymbol{C}_i^{-1}$ are denoted as r'_{jk} and τ'_j. Carrying through the projection in equation (13),

$$\boldsymbol{p}_i(t+1) = \frac{f_i}{z'} \begin{bmatrix} x' \\ y' \end{bmatrix} \quad (19)$$

where

$$\begin{aligned} x' &= x(r_{11} + r_{12}w_z - r_{13}w_y) + y(-r_{11}w_z + r_{12} + r_{13}w_x) \\ &\quad + z(r_{11}w_y - r_{12}w_x + r_{13}) + r_{11}t_x + r_{12}t_y + r_{13}t_z + \tau_x \\ y' &= x(r_{21} + r_{22}w_z - r_{23}w_y) + y(-r_{21}w_z + r_{22} + r_{23}w_x) \\ &\quad + z(r_{21}w_y - r_{22}w_x + r_{23}) + r_{21}t_x + r_{22}t_y + r_{23}t_z + \tau_y \\ z' &= x(r_{31} + r_{32}w_z - r_{33}w_y) + y(-r_{31}w_z + r_{32} + r_{33}w_x) \\ &\quad + z(r_{31}w_y - r_{32}w_x + r_{33}) + r_{31}t_x + r_{32}t_y + r_{33}t_z + \tau_z \end{aligned}$$

and where x', y' and z' are the coordinates of the 3D object after the motion described in equation (17) in the ith camera's view, and f_i is the focal length of the ith camera. Just as in the single camera case, we wish to compute the entries of $\boldsymbol{F_{i\mu}}\Big|_{\Delta\boldsymbol{\mu}=0}$ (which will be referred to as $\boldsymbol{F_{i\mu}}$ from this point on):

$$\boldsymbol{F_{i\mu}} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{bmatrix} \frac{f_i}{(z'_\mu)^2} \quad (20)$$

where $(z'_\mu)^2$ represents the derivative of z' w.r.t. $\boldsymbol{\mu}$ evaluated at $\boldsymbol{\mu} = 0$ and u_i and v_i are the derivatives of x' and y' w.r.t. the parameters of $\boldsymbol{\mu}$. The form of $\boldsymbol{F_{i\mu}}$ comes from the result of these derivatives. For example, the derivative of the point $\frac{x'}{z'}$ w.r.t. w_x is

$$\frac{d}{dw_x}\left(\frac{x'}{z'}\right) = \frac{u_1}{(z'_{\boldsymbol{\mu}})^2} \quad (21)$$

where

$$u_1 = \frac{d}{dw_x}(x') * z' - x' * \frac{d}{dw_x}(z'). \quad (22)$$

The remaining eleven derivatives are similarly computed. Therefore, to compute the entries of $\boldsymbol{F_{i\mu}}$, one needs to compute the derivatives of x', y' and z' w.r.t. each parameter of M and then evaluate each expression at $\boldsymbol{\mu} = 0$. As an example, the derivatives of x' w.r.t. w_x and t_x are given.

$$\begin{aligned} \frac{dx'}{dw_x} =& x * (r'_{13} * r_{21} - r'_{12} * r_{31}) + \\ & y * (r'_{13} * r_{22} - r'_{12} * r_{32}) + \\ & z * (r'_{13} * r_{23} - r'_{12} * r_{33}) - \\ & r'_{12} * \tau_z + r'_{13} * \tau_y \quad (23) \\ \frac{dx'}{dt_x} =& r'_{12}, \quad (24) \end{aligned}$$

where x, y, and z are the original 3D coordinates of the rigid object. The remaining derivation to obtain the entries of $\boldsymbol{F_{i\mu}}$ is left to the reader in lieu of space.

In the final step of full-motion recovery from multiple cameras, a single $\boldsymbol{\Delta\mu}$ is computed from all camera views. To do this, the spatial image gradients ($\boldsymbol{I}_p$), the temporal image gradients ($\boldsymbol{I}_p$), and the partial derivatives of the 2D parametric motion ($\boldsymbol{F_\mu}$ and $\boldsymbol{F_{i\mu}}$) must be combined. This is done in the following manner. First, we calculate $\boldsymbol{G'}$, a multiple camera version of the $\boldsymbol{G}$ shown in equation (6), as

$$\boldsymbol{G'} = [\boldsymbol{G_1 G_2 ... G_m}]^T \quad (25)$$

where $\boldsymbol{G_i}$ refers to the calculation for $\boldsymbol{G}$ from equation (6) for the ith camera in a multiple camera system with m cameras. For instance, $\boldsymbol{G_2}$ will be calculated for the second camera by $\boldsymbol{G_2} = \boldsymbol{I_{2p}} \boldsymbol{F_{2\mu}}$. Then, $\boldsymbol{G'}$ will represent the spatial image gradients and partial derivatives of the 2D parametric motion for all of the cameras in the system. Similarly, the temporal image gradients from all of the cameras

in the system for the current image are concatenated to form $\boldsymbol{I}'_t$. Now, to compute the global $\boldsymbol{\Delta\mu}$, $\boldsymbol{G}'$ from equation (25) is substituted in for $\boldsymbol{G}$ and $\boldsymbol{I}'_t$ is substituted for $\boldsymbol{I_t}$ in equation (5).

The $\boldsymbol{\Delta\mu}$ computed above represents the motion of the 3D object in the first camera's view, since we have considered that view the same as the world coordinates. For the ith camera, an additional step is needed. First, $\boldsymbol{M}$ is formed from $\boldsymbol{\Delta\mu}$. Then $\boldsymbol{M}$ is used in equation (17) to recover $\boldsymbol{X}_i(t+1)$ after the motion has taken place. This, of course, may be repeated for any number of cameras in a multiple camera system.

2.3. 3D Cylinder Head Model

The above methods for recovering 3D motion in single and multiple cameras may be used with any arbitrary rigid object. In our experiments, a 3D cylinder was chosen as the model. One reason for the choice of this shape is for its ease in initialization and close approximation to the head. Also, by unwrapping the cylinder texture of the face, a suitable image for face recognition is recovered, which is discussed in further detail in section 3.2.

The cylinder is initialized manually by adjusting the size and position of the cylinder on the face in the first frame of the video sequence and then adjusting the pose to match the pose of the face. This is done in our experiments by using an estimated $\boldsymbol{C}_i$ between each camera i and the world coordinate system (first camera). The estimation is done using OpenCV and a checker board pattern similar to that found in Figure 9(a). However, since the cylinder model must be initialized at the beginning of the face tracking task, the initialization from multiple cameras may be used to estimate the rotation and translation matrix $\boldsymbol{C}_i$ that is used in the multiple camera full-motion recovery model.

Cylinder motion in two camera views is displayed in Figures 6 and 7. Figures 6(a) and 7(a) display the cylinder at time t from the two camera views. Figures 6(b) and 7(b) display the same cylinder at time $t+1$ (1 second) after motion $\boldsymbol{M}$ has been applied to it.

3. Experimental Results

We will discuss three experiments that display the efficacy of our multi-camera full-motion recovery model. In the application area of pose estimation, we show the advantage of using the multi-camera model over the single camera model using a two-camera video sequence that contains pose ground truth of the face. The face recognition experiment also shows the advantage of our method in a real application.

3.1. Pose Estimation

The advantage of our multiple camera approach to full recovery of motion of the face is shown when tracking in a realistic setting. For this experiment, a video sequence of an individual was obtained from two cameras. To generate ground truth, a checkerboard pattern was placed on the head of the subject and a camera calibration package was utilized to obtain the rotation vector of the checkerboard in each frame [3]. The results for the yaw and pitch from the face tracking experiment are shown. Figures 2 and 3 display the results of the yaw for the left and right cameras, while Figures 4 and 5 display the results of the tilt for the left and right cameras, respectively. Figures 6 and 7 display images from the tracking results from the single (top) and multiple (bottom) camera models from the right and left cameras, respectively. In addition to these results, video sequences of two-camera and three-camera tracking that visually display the advantages of our multiple camera method over single camera tracking may be found on our web page [10].

Table 1 displays the mean squared error (MSE) and mean absolute error (MAE) of the pan and tilt angles comparison between the single camera and multiple camera models of the left and right cameras.

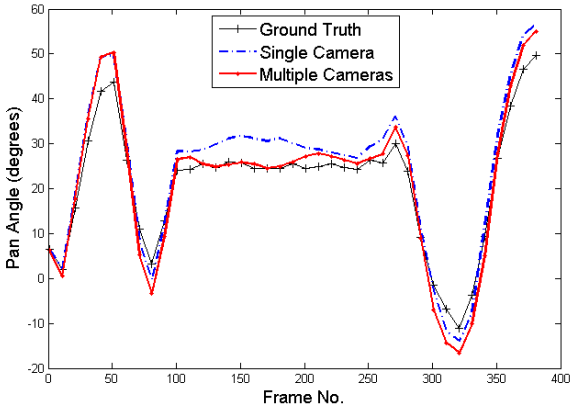


Figure 2. Yaw estimation of face tracking from left camera

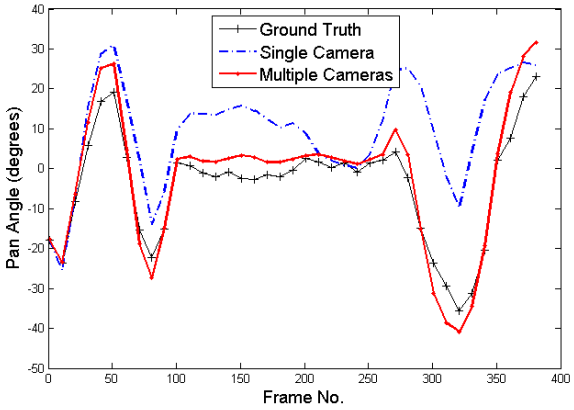


Figure 3. Yaw estimation of face tracking from right camera

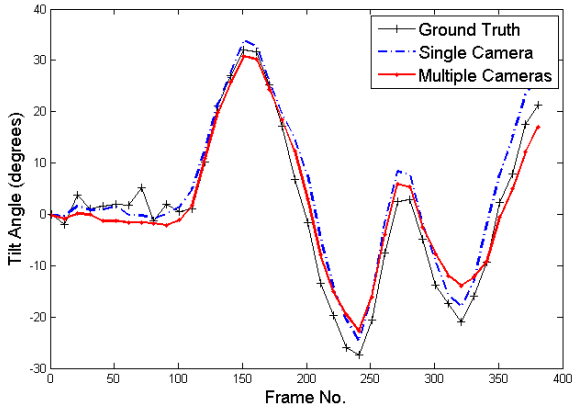


Figure 4. Pitch estimation of face tracking from left camera

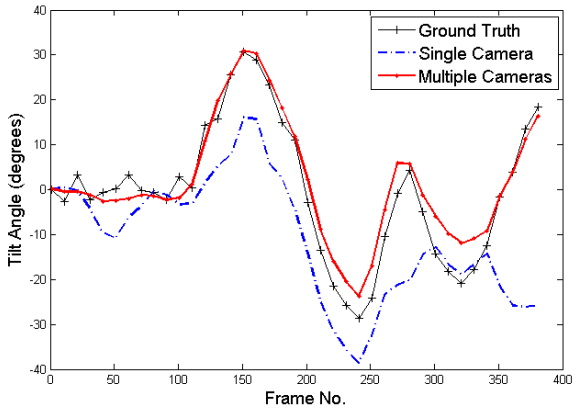


Figure 5. Pitch estimation of face tracking from right camera

Model	Rotation	Camera	MSE	MAE
single	pan	left	21.8	4.2
	pan	right	300.5	13.9
	tilt	left	18.8	3.5
	tilt	right	224.4	11.2
multi	pan	left	14.0	2.9
	pan	right	23.2	3.8
	tilt	left	13.9	3.2
	tilt	right	17.7	3.3

Table 1. Comparison of mean squared error (degrees squared) and mean absolute error (degrees) of pan and tilt between single and multiple camera models

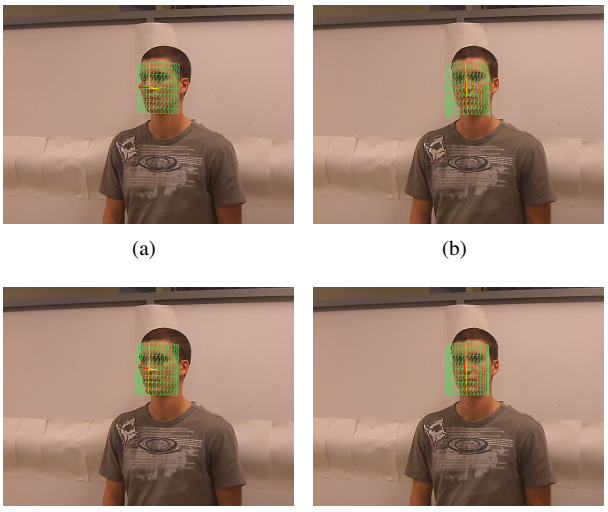
3.2. Face Recognition

A real world application of the full-motion recovery of the face from multiple cameras is in face recognition. Using the methodology similar to that in [11], we show that face recognition may be improved using the multiple camera model when compared to the single camera model. This is intuitive, since a better face tracking result should produce images more suitable for recognition in a traditional frontal face still image face recognition system.

model	none	mindist
single	68.5	82.8
multi	71.2	84.7

Table 2. Face Recognition Results from Single and Multiple Camera Models

Face recognition in our experiments is performed in the following manner for our two camera video sequences. First, the face is tracked using the single camera model and the multiple camera model in both video sequences. Figure 6 and 7 display example images from the tracking result used from one subject of our database. Each frame in the face tracking sequence produces a cylinder texture map of the face which provides as much of a frontal view of the face as possible with the 3D cylinder model. This resulting cylinder texture map is then cropped and used for recognition. An example of such an image is shown in Figure 8. Eigenfaces [21] is used for it's simplicity as a benchmark algorithm in this paper to test our methodology, but any still face recognition method may be used.

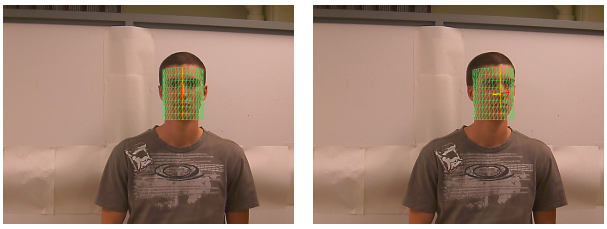


(a) (b) (c) (d)

Figure 6. Cylinder tracking result for single (a & b) & multiple (c & d) camera motion from second (right) camera

Our data consists of 100 frames per subject from a video sequence of 20 seconds over 18 subjects from two cameras. Only one frontal image per subject was used for training and the remaining data was used for testing. The results from our face recognition experiment are in Table 2.

The columns labeled "none" and "mindist" refer to whether or not the results from the two cameras were combined and how, as described in [11]. "None" refers to us-



(a) (b)

(c) (d)

Figure 7. Cylinder tracking result for single (a & b) & multiple (c & d) camera motion from first (left) camera

Figure 8. Example of face from cylinder texture map

ing all of the images for face recognition regardless of their source while "mindist" refers to using a minimum distance nearest neighbors classifier to decide the face recognition result between two cameras at the same time frame. In either case, it is apparent the multiple camera model produces images that are more suitable for still face recognition.

4. Discussion

The motion of a face as viewed from multiple cameras intuitively gives more information than that of a single camera. Using this information explicitly, we have formed a model for full-motion recovery of the face from multiple cameras. This multiple camera model outperforms the single camera model in pose estimation and face recognition.

In regards to pose estimation, the multiple camera model provided results closer to ground truth than the single camera model. The results reported in Table 1 clearly display the overall increase in robustness of the tracking result when using the multiple camera model over the single camera model. It is worth mentioning that the right camera produces more error in pose estimation than the left camera (as in Figures 3 and 5) because the initialization of the cylinder is performed on an image of a nonfrontal face, as seen in Figure 6(a). Although our experiments are shown with two cameras, the methodology may easily be extended to any number of cameras.

It is clear from the face recognition experiment that the multiple camera model produces images that are more suitable for still face recognition and thus improve the accuracy of the recognition result. It is worth noting that in this experiment, the video sequence was chosen so that the single camera model did not lose track of the face to provide a fair comparison of the quality of image that is produced for face recognition between the two methods. Obviously, if the single camera model loses track of the face, the recognition results would suffer greatly. This is seen most easily in Figure 9, where the masked texture produced from the single camera tracking (Figure 9(e)) is not suitable for face recognition (and is incorrectly labeled by our face recognition system) while the texture from the multiple camera tracking of the same frame (Figure 9(f)) is recognized correctly by our face recognition system.

5. Summary and Future Work

A novel approach to robust object tracking by full-motion recovery from multiple cameras is presented. This approach builds on the single camera model by explicitly including the motion from multiple cameras into a joint motion calculation. The multiple camera full-motion recovery model improves face tracking over a single camera as is shown in our experiments on pose estimation and face recognition. Future work on this topic includes improving the face tracking by including more cameras in our system, applying the motion model to other 3D shapes and implementing automatic initialization. We also plan to adapt and test our approach on a distributed surveillance camera system.

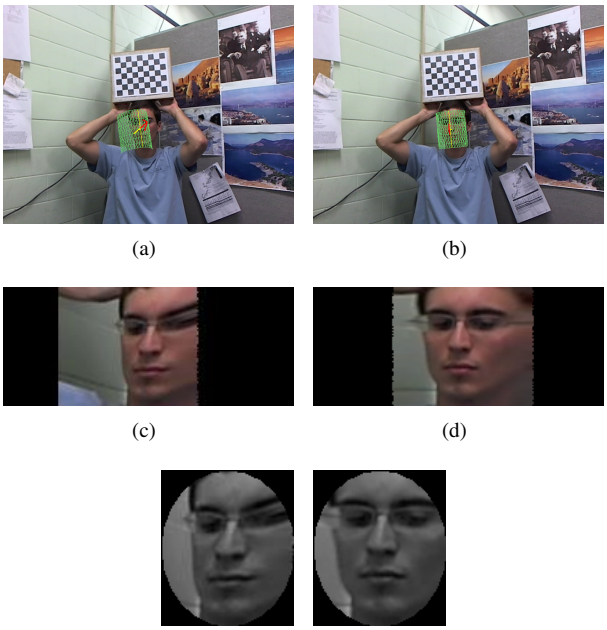
Acknowledgments

The authors would like to thank the reviewers for their valuable comments which have helped to improve the quality of the paper. This work is partially supported by Instituto de Telecomunicações and the UT Austin/Portugal Program CoLab grant (FCT) UTA-Est/MAI/0009/2009 (2009), supported by the Portuguese government.

References

[1] G. Aggarwal, A. Veeraraghavan, and R. Chellappa. 3d facial pose tracking in uncalibrated videos. *Pattern Recognition and Machine Intelligence*, pages 515–520, 2005. 1

[2] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *Pattern Recognition, 1996.,*



(a) (b) (c) (d) (e) (f)

Figure 9. Cylinder tracking result for single (a) & multiple (b) & camera motion, the extracted textures (c) & (d), and the masked images used for face recognition (e) & (f), respectively

Proceedings of the 13th International Conference on, volume 3, pages 611–616. IEEE, 2002. 1

[3] J.-Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/. 5

[4] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 8–15. IEEE, 2002. 4

[5] Q. Cai, A. Sankaranarayanan, Q. Zhang, Z. Zhang, and Z. Liu. Real time head pose tracking from multiple cameras with a generic model. In *Analysis and Modeling of Faces and Gestures Workshop, CVPR, 2010 IEEE Computer Society Conference on*, pages 25–32. IEEE, 2010. 2

[6] M. L. Cascia and S. Sclaroff. Vassilis athitsos, fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000. 1, 2

[7] Z. Chen, C. Chiang, and Z. Hsieh. Extending 3D Lucas–Kanade tracking with adaptive templates for head pose estimation. *Machine Vision and Applications*, pages 1–15, 2010. 1

[8] S. Choi and D. Kim. Robust head tracking using 3D ellipsoidal head model in particle filter. *Pattern Recognition*, 41(9):2901–2915, 2008. 1

[9] I. Essa and A. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):757–763, 2002. 1

[10] J. Harguess. Full-motion recovery from multiple video cameras: http://cvrc.ece.utexas.edu/research/vs2011, August 2011. 5

[11] J. Harguess, C. Hu, and J. K. Aggarwal. Fusing face recognition from multiple cameras. *Workshop on Applications of Computer Vision (WACV)*, 2009. 2, 6

[12] J. Harguess, C. Hu, and J. K. Aggarwal. Occlusion robust multi-camera face tracking. *The 3rd International Workshop on Machine Learning for Vision-based Motion Analysis (MLvMA-2011) in conjunction with IEEE CVPR 2011*, June 2011. 2

[13] C. Hu, J. Xiao, I. Matthews, S. Baker, J. Cohn, and T. Kanade. Fitting a single active appearance model simultaneously to multiple images. *Proceedings of the British Machine Vision Conference (BMVC2004)*, September 2004. 2

[14] T. Jebara and A. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. In *cvpr*, page 144. Published by the IEEE Computer Society, 1997. 1

[15] P. Jimenez, J. Nuevo, and L. Bergasa. Face pose estimation and tracking using automatic 3D model construction. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–7. IEEE, 2008. 1

[16] Z. Liu and Z. Zhang. Robust head motion computation by taking advantage of physical properties. In *Human Motion, 2000. Proceedings. Workshop on*, pages 73–77. IEEE, 2002. 1

[17] D. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992. 1

[18] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IEEE Proceedings of the 7th International Joint Conference on Artificial Intelligence*, April, 1981, pp. 674-679. 2

[19] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:607–626, 2009. 1

[20] J. Sung, T. Kanade, and D. Kim. Pose robust face tracking by combining active appearance models and cylinder head models. *Int. J. Comput. Vision*, 80(2):260–274, 2008. 1

[21] M. A. Turk and A. P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991. 6

[22] J. Xiao, T. Moriyama, T. Kanade, and J. Cohn. Robust full-motion recovery of head by dynamic templates and re-registration techniques. *International Journal of Imaging Systems and Technology*, 13:85 – 94, September 2003. 1, 3