Occlusion Robust Multi-Camera Face Tracking

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

This paper presents a novel approach to object tracking by using multiple views to assist with handling occlusion which improves the overall tracking result. The approach is applied to face tracking using a 3D cylinder head model, but any 3D rigid object may be tracked using this approach. All cameras in the system are used to estimate a joint motion model of the face, which is updated at each frame. Self-occlusion is handled by a weighted mask that depends on the pose of the face. Full face occlusion is first detected automatically by measuring and comparing image histograms of the current tracking result and a face template. If an occlusion from a camera is reported, it is not used in the global tracking result of the face from the multi-camera system. Experiments demonstrate that our method succeeds in tracking in both cases of self-occlusion and full face occlusion. Comparisons are made between single camera tracking, multi-camera tracking and occlusion robust multi-camera tracking using results from pose estimation. The performance of the occlusion robust multi-camera face tracking method is shown to produce more accurate estimates of the face pose and is able to estimate the face pose even under severe face occlusion.

1. Introduction

Face tracking, especially the full-motion recovery of the face (3 translations and 3 rotations) is necessary for many computer vision tasks such as human computer interaction and surveillance. Recovering the motion of the face robustly and accurately may assist in face expression and recognition tasks where the alignment of the face is usually required.

An important task within face tracking is pose estimation [18]. There are many approaches in pose estimation and they may be generally classified as either feature-based approaches or model-based approaches. Feature-based approaches, as in [12, 15, 16] , attempt to track features of the face or object to recover the pose. They are regarded as efficient and flexible but their performance depends on the availability of good features. Model-based approaches treat the face as a 3D object and have had much success in tracking faces. The head may be approximated as a 3D rigid object, so recovering the motion of the face is equivalent to recovering the motion of the 3D object. One typical model is a 3D cylinder head model, as seen in [1, 6, 7, 19, 20]. Ellipsoid models may also be used as in [2, 8, 9]. [5] proposes to use a generic 3D surface model with a more detailed 3D shape for tracking. Since the whole face region is used in model-based approaches, self-occlusion and full face occlusion are still very challenging.

Several approaches have been tried in handling occlusion in face tracking. In [21], an occlusion pixel is detected by comparing the motion residual error of each pixel with all pixels used in the tracking. If the motion residual error is larger than a threshold, the pixel is classified as an outlier or occlusion. In some works such as in [14], occlusion is detected simply by calculating the number of skin color pixels. If there is very small portion of skin pixels in the target, occlusion is detected.

In many computer vision applications it is quite common to have more than one camera available for sensing. In this work, we propose a method which utilizes the multiple views of a single face available in a multi-camera setting to make a more robust face motion calculation. This method is different from the use of stereo information, such as in [13], since we may combine the views of any number of cameras for the motion calculation without computing stereo. We present a complete framework for combining the motion from multiple views of the face into a single motion calculation that is then used in all views. Although many 3D models would suffice, We employ the cylinder head model in this work since the relative error between the cylinder model and the real geometry of the face is small [6] and the cylinder model may be easily adapted to any face through initialization of the cylinder. Additionally, the reconstructed

[*]This work is partially supported by Instituto de Telecomunicações and the UT Austin/Portugal Program CoLab grant (FCT) UTA-Est/MAI/0009/2009 (2009).

frontal face from the cylinder model can be used for face recognition [10, 11].

There are two types of occlusion in our problem. The first type is self-occlusion where some parts of the face are hidden by other parts of the face because of the camera view. The second type is when other objects block the camera view and cause a large part of the face, or the entire face, to become invisible. In our algorithm , we tackle the first type of occlusion by creating a weighted mask on which invisible pixels are assigned the lowest weights. Our solution to the second type of occlusion is to detect the occlusion and turn off the motion update from that camera. This still allows the tracking to continue from the other cameras which results in stable tracking of the face.

The proposed method is demonstrated on two typical video sequences from three cameras with ground truth pose of the face obtained from a checker pattern. We show two main results. First, the multi-camera based face tracking method is far superior to the single camera tracking result from all cameras. Second, in the presence of occlusion, the occlusion robust multi-camera face tracking significantly outperforms both the multi-camera based method and the single camera tracking. Both numerical pose estimation results and visual examples of the tracking with each method are shown. Please note that although the focus of this paper is face tracking, the proposed method may be used for tracking any 3D rigid object in a multi-camera setting.

2. Robust Multi-Camera Face Tracking

In this section, we introduce our method of occlusion robust multi-camera face tracking. First, we present our method of combining the motion of the face from multiple views to form a single motion using the cylinder head model. Second, we present our method for handling occlusion.

In our multi-camera system, one of the cameras is specified as "camera 1", as in Figure 1, and the world coordinate system is based on that camera's coordinate system.

2.1. Multi-Camera Face Tracking

In the following description of our face tracking method, we will assume that the 3D model is the cylinder head model [20]. However, any 3D rigid object may be used to capture the six parameters (three translations and three rotations) of motion that are needed to describe the motion performed by the head.

In the world coordinate system (camera 1), the motion of the 3D points of the cylinder is described as

$$\boldsymbol{X}_{t+1} = \boldsymbol{M} * \boldsymbol{X}_t \tag{1}$$

where

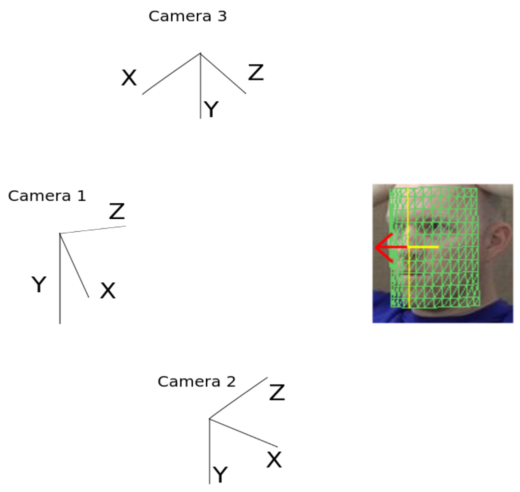

Figure 1. Example 3 camera system viewing the same 3D cylinder

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

$\boldsymbol{X}_{t+1}$ 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 vector representing rigid motion, including 3D rotation $(\omega_x, \omega_y, \omega_z)$ 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. The rigid motion of the head from time t to time $t+1$ is denoted as $\Delta\boldsymbol{\mu}$. If $\boldsymbol{p}_{1t} = (u, v)$ is the projection point in the image plane $\boldsymbol{I}_{1t}$ in camera 1 of point $\boldsymbol{x}$ on the 3D object , then the new location of $\boldsymbol{p}_{1t}$ in the next frame $\boldsymbol{I}_{1t+1}$ is estimated as

$$\boldsymbol{p}_{1t+1} = \boldsymbol{F}_1(\boldsymbol{p}_{1t}, \Delta\boldsymbol{\mu}). \tag{3}$$

The next image frame may then be computed by

$$\boldsymbol{I}_{1t+1}(\boldsymbol{F}_1(\boldsymbol{p}_{1t}, \Delta\boldsymbol{\mu})) = \boldsymbol{I}_{1t}(\boldsymbol{p}_{1t}), \tag{4}$$

where $\boldsymbol{F}_1$ is the 2D parametric motion function of $\boldsymbol{p}_{1t}$. An 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. In most cases, this is a reasonable assumption.

The same motion of the 3D points may be seen from the other camera's coordinate systems and image planes. A three camera system viewing the same 3D cylinder is shown in Figure 1.

In the ith camera's view

$$\boldsymbol{X}_{it+1} = \boldsymbol{M}_i * \boldsymbol{X}_{it} \tag{5}$$

$$\boldsymbol{p}_{it+1} = \boldsymbol{K}_i * \boldsymbol{M}_i * \boldsymbol{X}_{it} \tag{6}$$

$$\boldsymbol{X}_{1t} = \boldsymbol{C}_i * \boldsymbol{X}_{it} \tag{7}$$

where $\boldsymbol{C}_i$ is the 4×4 combined rotation and translation matrix representing the transformation between the ith camera's coordinate system and the world coordinate system, $\boldsymbol{X}_{it}$ are the 3D points w.r.t. the ith camera's coordinate system, and $\boldsymbol{p}_{it}$ are the image coordinates in the ith camera's view after a projection with camera matrix $\boldsymbol{K}_i$.

The motion in each of the cameras is related back to the world coordinate system in the following manner. We first note that

$$\boldsymbol{X}_{it+1} = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{X}_{it} = \boldsymbol{M}_i * \boldsymbol{X}_{it}. \tag{8}$$

Therefore,

$$\boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{X}_{1t} = \boldsymbol{M}_i * \boldsymbol{C}_i^{-1} * \boldsymbol{X}_{1t} \tag{9}$$

$$\boldsymbol{M}_i = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{C}_i. \tag{10}$$

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

$$\boldsymbol{X}_{it+1} = \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{C}_i * \boldsymbol{X}_{it}. \tag{11}$$

We may now explicitly solve for the full-motion recovery and compute $\boldsymbol{\Delta\mu}$ from the information present in all cameras. 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 [17]. The result is

$$\Delta\boldsymbol{\mu} = -(\sum_{\Omega} \hat{\boldsymbol{G}}^T \hat{\boldsymbol{G}})^{-1} \sum_{\Omega} \hat{\boldsymbol{I}}_t \hat{\boldsymbol{G}} \tag{12}$$

where

$$\hat{\boldsymbol{G}} = [\boldsymbol{G}_1 \boldsymbol{G}_2 ... \boldsymbol{G}_m]^T \tag{13}$$

$$\boldsymbol{G}_i = \boldsymbol{I}_{i\boldsymbol{p}} \boldsymbol{F}_{i\boldsymbol{\mu}} \tag{14}$$

and where $\boldsymbol{\Omega}$ is the region of overlapping pixels between the two frames, $\boldsymbol{F}_{i\boldsymbol{\mu}}$ is the partial derivative of $\boldsymbol{F}_i$ w.r.t. the rigid motion vector, and $\boldsymbol{I}_{i\boldsymbol{p}}$ is the spatial image gradient, all w.r.t. camera i. $\hat{\boldsymbol{I}}_t$ is the concatenation of each camera's temporal image gradients.

The solution of $\boldsymbol{F}_{1\boldsymbol{\mu}}$ w.r.t. camera 1 is well known and is [20]

$$\boldsymbol{F}_{1\boldsymbol{\mu}}\Big|_{\Delta\boldsymbol{\mu}=0} = \begin{bmatrix} -xy & x^2+z^2 & -yz & z & 0 & -x \\ -(y^2+z^2) & xy & xz & 0 & z & -y \end{bmatrix} \frac{f}{z^2}, \tag{15}$$

where x, y and z are the 3D coordinates of the object and f is the focal length of the camera.

Recalling equations (6) and (10), we may write the image projection in the coordinate system of camera i as

$$\boldsymbol{p}_{it+1} = \boldsymbol{K}_i * \boldsymbol{C}_i^{-1} * \boldsymbol{M} * \boldsymbol{C}_i * \boldsymbol{X}_{it} = \frac{f_i}{z'} \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{16}$$

where x', y' and z' are the coordinates of the 3D object after the motion described in equation (11) in the ith camera's view, and f_i is the focal length of the ith camera. Notice that the motion $\boldsymbol{M}$ of the cylinder w.r.t. the world coordinate system is captured in equation (16). For camera i,

$$\boldsymbol{F}_{i\boldsymbol{\mu}}\Big|_{\Delta\boldsymbol{\mu}=0} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{bmatrix} \frac{f_i}{(z'_{\mu})^2} \tag{17}$$

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

$$\frac{d}{dw_x}\left(\frac{x'}{z'}\right) = \frac{u_1}{(z'_{\boldsymbol{\mu}})^2} \tag{18}$$

where

$$u_1 = \frac{d}{dw_x}(x') * z' - x' * \frac{d}{dw_x}(z'). \tag{19}$$

The remaining eleven derivatives are similarly computed. Therefore, to compute the entries of $\boldsymbol{F}_{i\boldsymbol{\mu}}$, one needs to compute the derivatives of x', y' and z' w.r.t. each parameter of M and then evaluate each expression at $\boldsymbol{\mu} = 0$. The remaining derivation to obtain the entries of $\boldsymbol{F}_{i\boldsymbol{\mu}}$ is left to the reader in lieu of space.

To compute the global $\Delta\boldsymbol{\mu}$, $\hat{\boldsymbol{G}}$ is formed and equation (12) is used to compute the result. For single camera tracking, the rigid head motion vector $\Delta\boldsymbol{\mu}$ is recovered by only considering camera 1 in the motion calculation.

2.2. Occlusion

In this work, we are concerned with handling two main types of occlusion. The first type, self-occlusion, is very common in face tracking tasks. If the pose of the face is nonfrontal, particularly when the pose is larger than around +/- 30 degrees in yaw and/or tilt from the frontal pose, pixels of the face that have been used for tracking will not be present and therefore may affect the tracking result greatly. The second type is full face occlusion, meaning that the entire face is occluded in one or more cameras.

2.2.1 Self-Occlusion

We handle the case of self-occlusion in our tracking model by weighting a mask that is used in the calculation of the temporal image gradient for each camera's image. For a frontal face pose, the mask is centered on the face and the most weight is given to the center pixels and the least to pixels furthest away from the center using Gaussian weights. If the pose of the face is nonfrontal, the mask is centered on the part of the face that is most visible to the camera's view. Therefore, pixels that are most visible are weighted the most in the face tracking calculation and the camera with the most frontal view of the face will be weighted the most in our multi-camera tracking method. The weighted mask is updated for each frame based on pose estimation.

2.2.2 Full Face Occlusions

While the above method is beneficial for tracking the face, even across face poses that are far from frontal, it is not sufficient for handling partial or full occlusion of the face. Our method for handling face occlusion relies on multi-camera tracking. Occlusions of the face are handled by first detecting the occlusion in a particular camera and then removing that camera from the multi-camera motion calculation.

To determine if a camera is being occluded, we use a comparison of image histograms approach. First, the unwrapped cylinder texture image is taken from the template (usually the first image in the sequence) and the current frame. An example template image is found in Figure 2(a), while a non-occluded current frame and an occluded current frame are found in Figures 2(b) and 2(c), respectively.

The histograms of the images are formed from the H channel of the HSV color map of the image and are shown in Figures 3 and 4.

To measure the similarity of the two histograms (one from the template image and the other from the current frame), we employ the Bhattacharyya coefficient [3] , which is a measure of the amount of overlap between two statistical samples and is commonly used to measure the similarity of histograms. Figures 5, 6 and 7 display the Bhattacharyya coefficients for cameras 1, 2 and 3, respectively. The dashed line in each of the plots is the threshold value which was determined as two standard deviations from the mean when no occlusion are present. This threshold, once found, was used throughout the experiments. From the plots, it is clear that camera 3 has detected an occlusion.

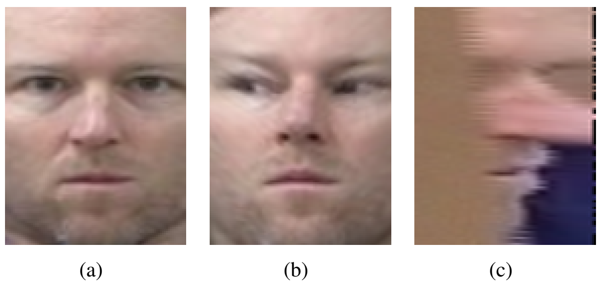

Figure 2. Example template, current and occluded frames

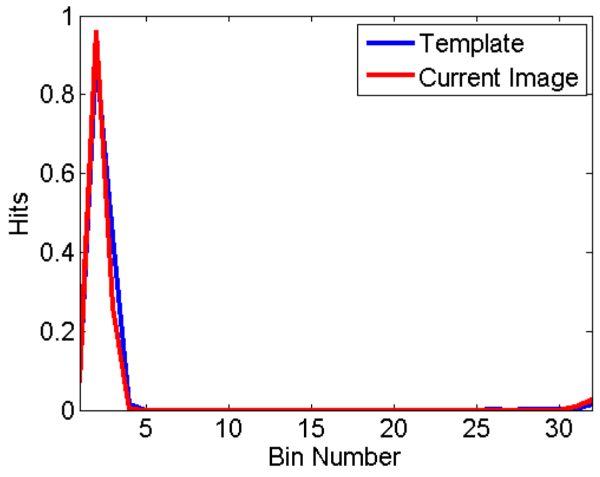

Figure 3. Histogram of template and current frame

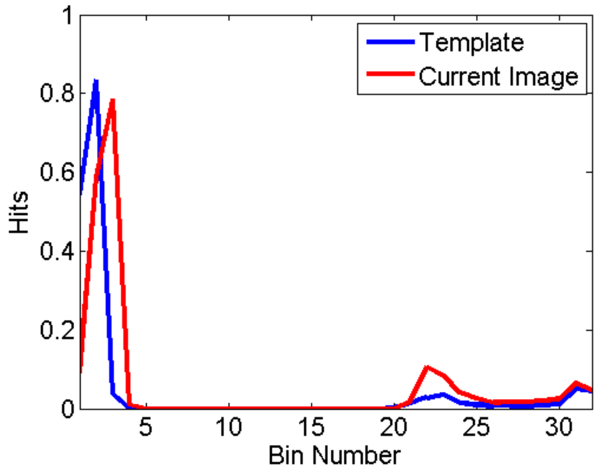

Figure 4. Histogram of template and partially occluded frame

3. Experimental Results

In each of the experiments, 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. The rotation and translation parameters between the cameras are then used to refine the initialization so that the cylinder is initialized in an acceptable position for all cameras.

For these experiments, two video sequences of an individual was obtained from three cameras. The first video sequence was used to estimate the threshold for detecting occlusion and then towards the end of the sequence an occlusion in camera 3 is presented. In the second video sequence, an occlusion is presented in camera 2. In our multi-camera system, camera 1 is the center-most camera, camera 2 is the right-most camera, and camera 3 is the left-most camera. To generate ground truth, a checkerboard pattern was placed on

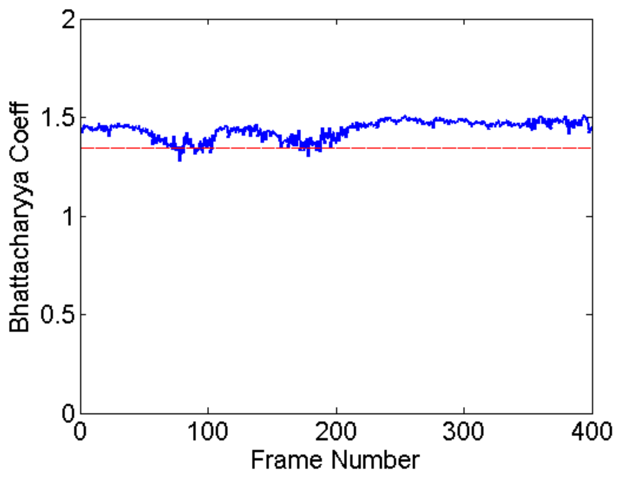

Figure 5. Bhattacharyya coeff for camera 1

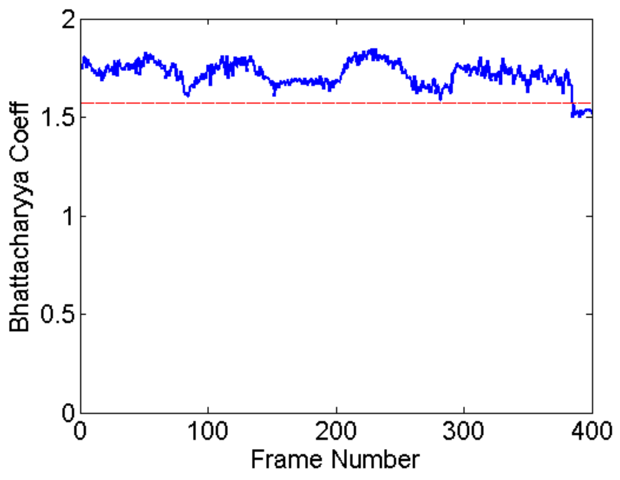

Figure 6. Bhattacharyya coeff for camera 2

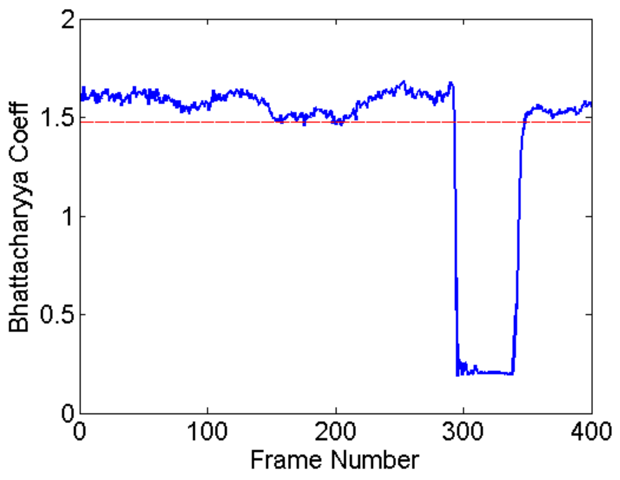

Figure 7. Bhattacharyya coeff for camera 3

Table 1. RMS error between estimated pose and ground truth for yaw (degrees)

Camera	Method	Sequence 1	Sequence 2
1	Single	5.28	3.49
	Multi	6.52	24.93
	MultiOcc	4.27	4.74
2	Single	13.46	6.67
	Multi	6.44	7.78
	MultiOcc	3.89	1.10
3	Single	4.96	6.44
	Multi	4.05	24.98
	MultiOcc	3.45	4.66

Table 2. RMS error between estimated pose and ground truth for tilt (degrees)

Camera	Method	Sequence 1	Sequence 2
1	Single	8.22	3.72
	Multi	8.57	6.77
	MultiOcc	8.36	3.42
2	Single	16.43	3.53
	Multi	7.95	4.81
	MultiOcc	8.20	3.96
3	Single	18.62	3.13
	Multi	7.95	7.36
	MultiOcc	7.99	3.25

the head of the subject and OpenCV [4] was used to obtain the rotation vector of the checkerboard in each frame.

Figures 8, 9 and 10 display the results of the yaw for cameras 1, 2 and 3, while Figures 11, 12 and 13 display the results of the tilt for cameras 1, 2 and 3, respectively. In each of the plots, a camera that has lost track of the face is denoted by a constant value for the pose after the track is lost. An instance of this is apparent in Figure 13 where the single camera has lost track of the face in frame number 300. In addition to these results, we will submit the video sequences of three-camera tracking to visually display the advantages of our multiple camera method over single camera tracking.

Tables 1 and 2 display the root mean squared error (RMS) of the pan and tilt angles comparison between the single camera (Single), multiple camera (Multi) and occlusion robust multiple camera (MultiOcc) tracking of the multi-camera system. The RMS value was computed for each camera and method using the least number of frames that were successfully tracked for all three cameras. For example, in sequence 2, camera 2 is unable to track the face past frame 71. Therefore, only frames 1 through 71 were used for all three cameras in the RMS calculation. Therefore, the RMS values unfairly penalize the occlusion robust multi-camera face tracking method when the single camera tracking has failed.

4. Discussion

It is clear from the results of the pose estimation plots and the RMS error of each method with ground truth that the occlusion robust multi-camera face tracking is far superior to the single camera method. In both sequences there is at least one camera that loses track of the face with single camera tracking. In sequence 2, even the multi-camera face tracking method has trouble with the occlusion. It is important to note that when the multi-camera face tracking approach fails, it loses track of the face in all three cameras since they are all connected by a single motion calculation. However, by explicitly handling occlusion, the cameras work together to maintain a robust tracking of the face in the proposed method. Although our experiments are shown with three cameras, the methodology may easily be extended to any number of cameras.

5. Summary and Future Work

A novel approach to handling occlusion in object tracking using multiple cameras is proposed. The proposed occlusion robust multi-camera face tracking method has been shown to be robust to self-occlusion and full face occlusion and significantly outperforms the single camera tracking from each of the cameras. 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 of the cylinder. We will also investigate other occlusion detection methods, both for occlusion at the pixel level as well as the camera level.

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. 33

[2] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 3, pages 611–616. IEEE, 2002. 33

[3] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc*, 35(99-109):4, 1943. 36

[4] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 37

[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. 33

[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. 33

[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. 33

[8] S. Choi and D. Kim. Robust head tracking using 3D ellipsoidal head model in particle filter. *Pattern Recognition*, 41(9):2901–2915, 2008. 33

[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. 33

[10] J. Harguess, C. Hu, and J. K. Aggarwal. Fusing face recognition from multiple cameras. *Workshop on Applications of Computer Vision (WACV)*, 2009. 34

[11] C. Hu, J. Harguess, and J. K. Aggarwal. Patch-based face recognition from video. In *International Conference on Image Processing (ICIP)*, pages 1–4, November 2009. 34

[12] 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. 33

[13] 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. 33

[14] D. Lin and M. Liu. Face occlusion detection for automated teller machine surveillance. *Lecture notes in computer science*, 4319:641–651, 2006. 33

[15] 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. 33

[16] D. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, 1992. 33

[17] 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. 35

[18] 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. 33

[19] 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. 33

[20] 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. 33, 34, 35

[21] Y. Zhang and C. Kambhamettu. 3D head tracking under partial occlusion. *Pattern Recognition*, 35(7):1545–1558, 2002. 33

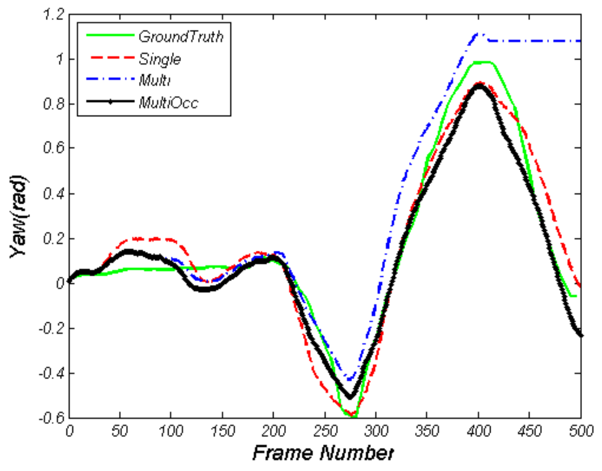

Figure 8. Yaw estimation of tracking sequence 1 from camera 1

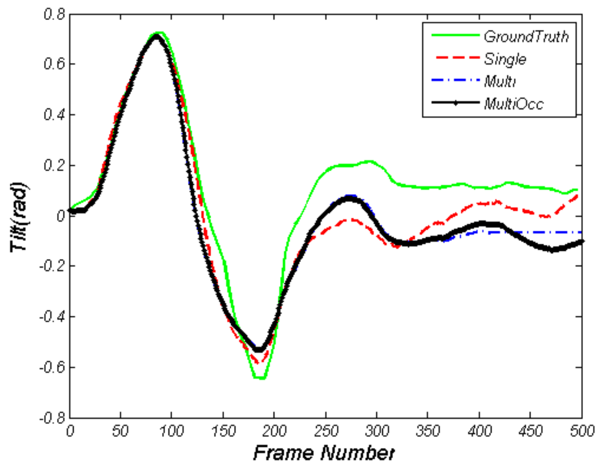

Figure 11. Tilt estimation of tracking sequence 1 from camera 1

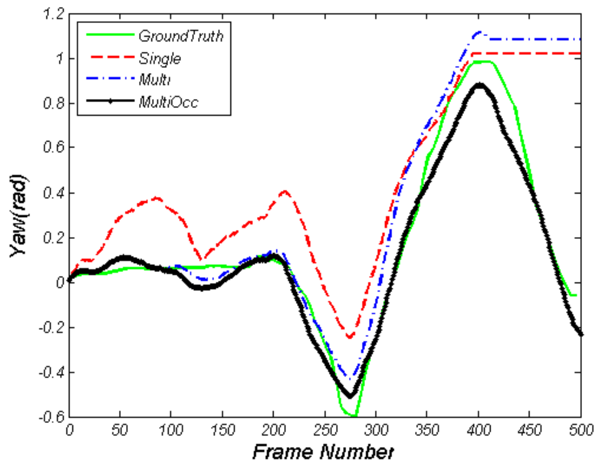

Figure 9. Yaw estimation of tracking sequence 1 from camera 2

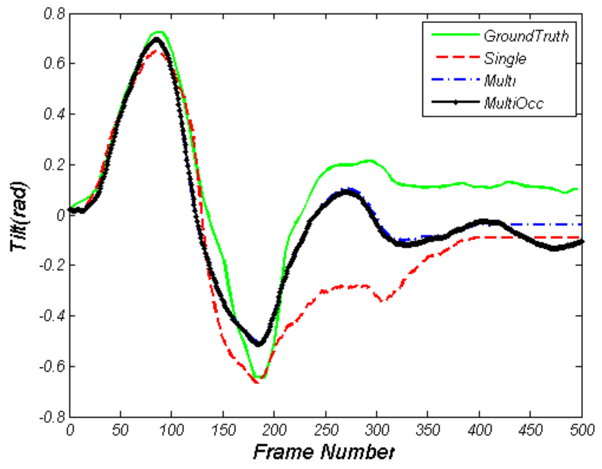

Figure 12. Tilt estimation of tracking sequence 1 from camera 2

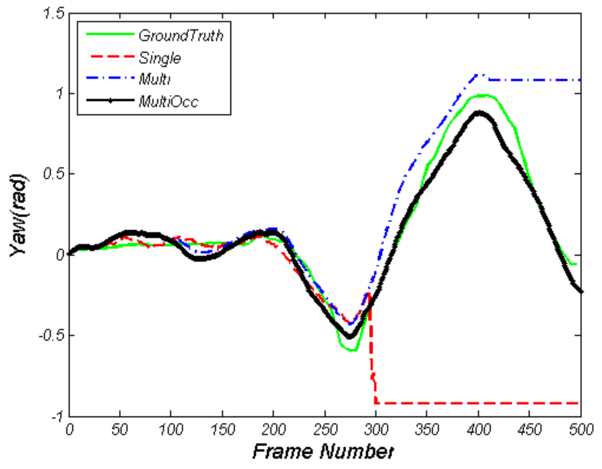

Figure 10. Yaw estimation of tracking sequence 1 from camera 3

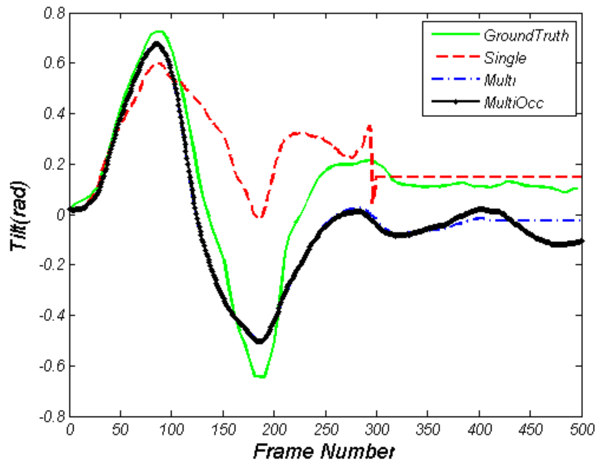

Figure 13. Tilt estimation of tracking sequence 1 from camera 3

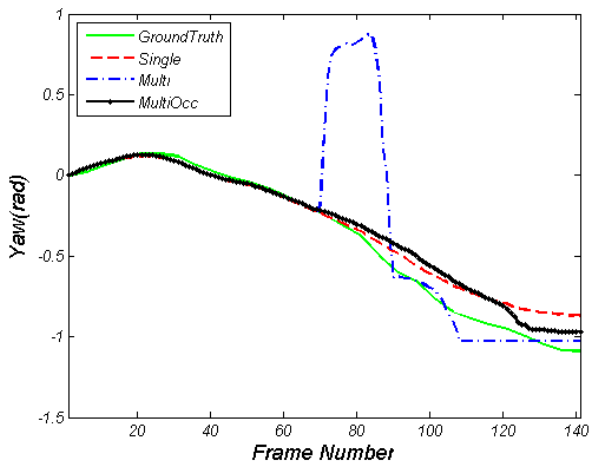

Figure 14. Yaw estimation of tracking sequence 2 from camera 1

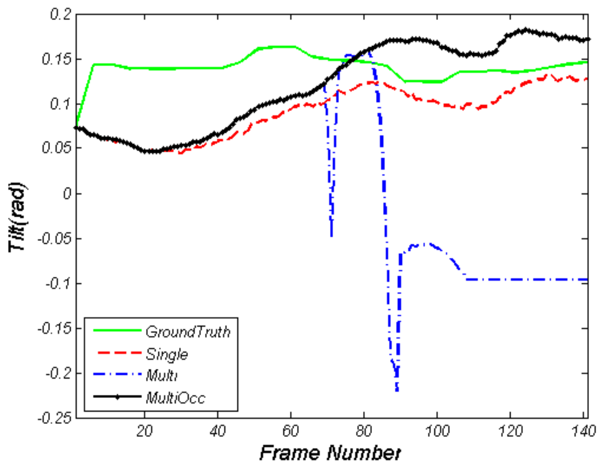

Figure 17. Tilt estimation of tracking sequence 2 from camera 1

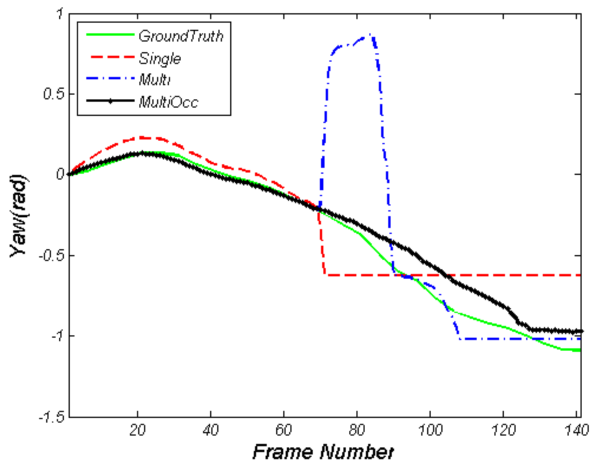

Figure 15. Yaw estimation of tracking sequence 2 from camera 2

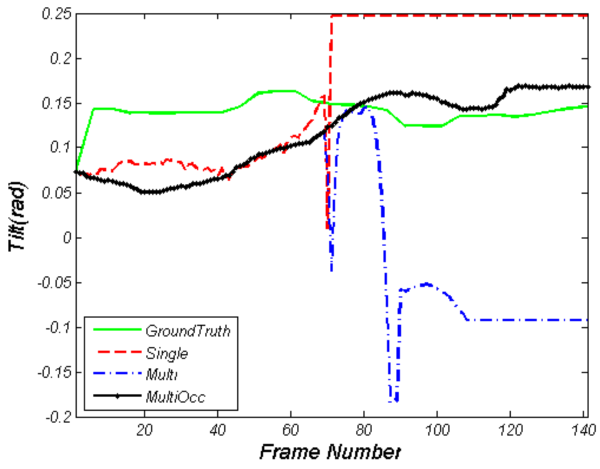

Figure 18. Tilt estimation of tracking sequence 2 from camera 2

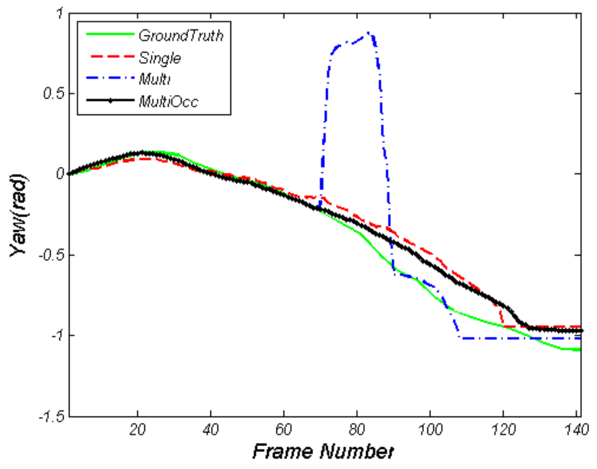

Figure 16. Yaw estimation of tracking sequence 2 from camera 3

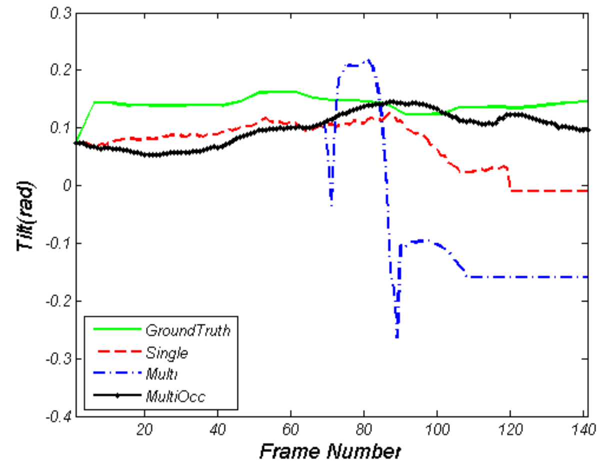

Figure 19. Tilt estimation of tracking sequence 2 from camera 3