Fusing Face Recognition from Multiple Cameras

Josh Harguess, Changbo Hu, J. K. Aggarwal *
Computer & Vision Research Center / Department of ECE
Department of Electrical & Computer Engineering
The University of Texas at Austin
{*harguess,chu1*}*@ece.utexas.edu, aggarwaljk@mail.utexas.edu*

Abstract

Face recognition from video has recently received much interest. However, several challenges for such a system exist, such as resolution, occlusion (from objects or self-occlusion), motion blur, and illumination. The aim of this paper is to overcome the problem of self-occlusion by observing a person from multiple cameras with uniquely different views of the person's face and fusing the recognition results in a meaningful way. Each camera may only capture a part of the face, such as the right or left half of the face. We propose a methodology to use cylinder head models (CHMs) to track the face of a subject in multiple cameras. The problem of face recognition from video is then transformed to a still face recognition problem which has been well studied. The recognition results are fused based on the extracted pose of the face. For instance, the recognition result from a frontal face should be weighted higher than the recognition result from a face with a yaw of 30°*. Eigenfaces is used for still face recognition along with the average-half-face to reduce the effect of transformation errors. Results of tracking are further aggregated to produce* 100% *accuracy using video taken from two cameras in our lab.*

1. Introduction

Face recognition from video has received much interest in recent times. This is likely due to heightened security and the availability of inexpensive surveillance cameras. Also, face recognition from video may produce better overall accuracy since a video will have many frames of a subject's face instead of just a few examples. Many researchers have focused on face recognition from a single video camera [3, 8, 7, 2], but a more realistic surveillance scene would include video from multiple cameras, such as cameras monitoring activity in a computer store.

Li *et al*. [3] use a multi-view face model to automatically fit to a detected face from video. The detected faces are then warped to the mean shape with the frontal view. Kernel discriminant analysis (KDA) is then used for recognition. Zhou *et al*. [8] use a probabilistic model to recognize faces from video using both still images and video as the gallery. Face recognition from video with occlusions is considered by Hu *et al*. [2]. They use a patch-based framework to reconstruct full frontal images for still face recognition. Zhang *et al*. [7] build 3D face textures from faces that are present in video.

The focus of this paper is on a scenario where multiple cameras are cooperating to build a more accurate face recognition system. The cameras are conducting a broad area surveillance and they focus on a person of interest. In this scenario, stereo reconstruction of the face would require calibration of the cameras and would not be profitable since overlap of the views of the two cameras may be minimal. Also, a generic model of the head is desired so that tracking of different individuals is possible without training a specific face model. We propose a method to track the subject using a cylinder head model (CHM) [6] in multiple cameras. One of the advantages of the CHM is that a cylinder is a close approximation of a 3D head, allowing for accurate tracking of the face. Knowing the pose from the CHM allows us to produce a frontal view of face even though some information from the face might be missing due to self-occlusion. This tracking result helps to transform the problem of face recognition from video to a still face recognition task.

Using the pose information and the recognition result, the classification results of both cameras are fused using several methods; independent, minimum distance, best pose, multiplier weights, and Gaussian weights. The minimum distance does not use pose information explicitly. The best pose, multiplier weights and Gaussian weights use a weighting scheme that gives preference to the face recognition result from the frontal pose (0° yaw) and pe-

*The research was supported in part by Texas Higher Education Coordinating Board award # 003658-0140-2007.

nalizes the result from the maximum pose present in the video ($\pm 70° yaw$ in our experiments). The face recognition method used in this paper is based on eigenfaces [5]. The pose weights are applied to the Euclidean distance between the test image and closest training image in the subspace created by principal components analysis (PCA). This weighting scheme gives more preference for recognition to frames where the pose of the face is mostly frontal. By fusing results from multiple cameras, the recognition of faces is improved from 67.4% to 94.4% in our videos. Face recognition results from a single subject's track are further aggregated to produce 100% accuracy for all subjects.

2. Face Tracking with Cylinder Head Models

In order to translate the problem of face recognition from video to a still face recognition problem, we desire a method to robustly track the face of an individual from multiple cameras so that we may combine the tracking results in a meaningful way. The cylinder head model (CHM) [6] has several advantages. First, CHMs are able to recover the full-motion parameters (3 rotations and 3 translations) of the head. Since this paper deals with multiple surveillance cameras, the recovery of these parameters is crucial in order to fuse information about the head and face in all camera views. In order to keep this paper self-contained, a summary of the cylinder head model and tracking algorithm found in [6] follows.

The cylinder head model makes a basic assumption that we can treat the head (and thus face) as a cylinder. Therefore, rotation or translation performed by the head can be estimated by a cylinder with 6 parameters (3 rotations and 3 translations). Let the vector $\boldsymbol{\mu}$ represent the rigid motion, including 3D rotation (θ_x, θ_y, θ_z) and the translation (t_x, t_y, t_z). If $\boldsymbol{x} = (x, y, z)^T$ is a 3D coordinate of a point on the cylinder surface, then the new location of $\boldsymbol{x}$ after rigid motion transformation by $\boldsymbol{\mu}$ is

$$\boldsymbol{M}(\boldsymbol{x}, \boldsymbol{\mu}) = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{T}, \tag{1}$$

where $\boldsymbol{M}$ 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 described by the change in the rigid motion vector, $\Delta\boldsymbol{\mu}$. Therefore, if $\boldsymbol{p}_t = (u, v)$ is the projection point in the image plane $\boldsymbol{I_t}$ of point $\boldsymbol{x}$ on the cylinder in 3D (which are depicted in Figure 1), then the new location of point $\boldsymbol{p}_{t+1}$ in the next frame $\boldsymbol{I}_{t+1}$ is estimated as

$$\boldsymbol{p}_{t+1} = \boldsymbol{G}(\boldsymbol{p}_t, \Delta\boldsymbol{\mu}) \tag{2}$$

and the next frame can be computed by

$$\boldsymbol{I}_{t+1}(\boldsymbol{G}(\boldsymbol{p_t}, \Delta\boldsymbol{\mu})) = \boldsymbol{I}_t(\boldsymbol{p}_t), \tag{3}$$

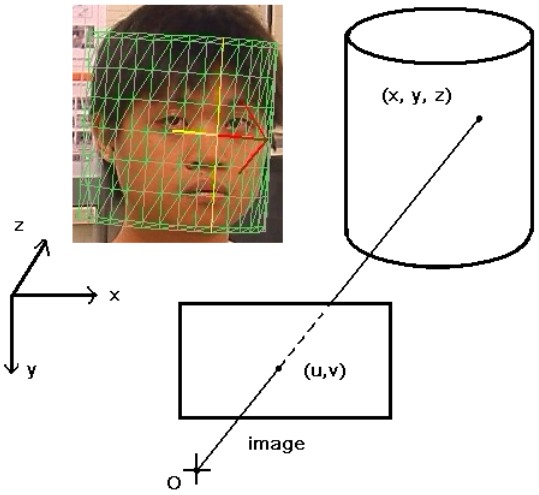


Figure 1. Relationship between points on the 3D cylinder model and the image plane.

where $\boldsymbol{G}$ is the 2D parametric motion function of $\boldsymbol{p_t}$. In this estimation we assume that the illumination does not change between frames, so the pixel intensities between the two frames are consistent.

The change in rigid motion vector $\Delta\boldsymbol{\mu}$ can be obtained through a minimization of the error between two successive image frames and can be solved by using the Lucas-Kanade image alignment algorithm [4]. The solution is

$$\Delta\boldsymbol{\mu} = -(\sum_{\Omega}(\boldsymbol{I_p}\boldsymbol{G_\mu})^T(\boldsymbol{I_p}\boldsymbol{G_\mu}))^{-1}\sum_{\Omega}(\boldsymbol{I_t}(\boldsymbol{I_p}\boldsymbol{G_\mu})) \tag{4}$$

where $\boldsymbol{\Omega}$ is the region of overlapping pixels between the two frames, $\boldsymbol{G_\mu}$ is the partial derivative of $\boldsymbol{G}$ with respect to the rigid motion vector, and $\boldsymbol{I_p}$ and $\boldsymbol{I_t}$ are the spatial and temporal image gradients, respectively.

Assuming that the camera projection matrix depends only on the focal length, the derivative of $\boldsymbol{G}$ with respect to the rigid motion vector at $\boldsymbol{\mu} = 0$ is [6]

$$\begin{aligned}&\boldsymbol{G_\mu}\Big|_{\Delta\boldsymbol{\mu}=0}\\ &= \begin{bmatrix} -xy & x^2+z^2 & -yz & z & 0 & -x \\ -(y^2+z^2) & xy & xz & 0 & z & -y \end{bmatrix}\frac{f}{z^2},\end{aligned} \tag{5}$$

where f is the focal length of the camera and x, y and z are the 3D coordinates. By plugging the result of (5) into equation (4), the rigid head motion vector $\Delta\boldsymbol{\mu}$ is recovered.

Using the CHM tracking result, we scan the image to the cylinder, then unwrap the cylinder to a standard texture map as portrayed in Figure 2. The pixel value for each point in the texture image ((d) in Figure 2) is found by locating the point on the cylinder model ((c) in Figure 2), finding the

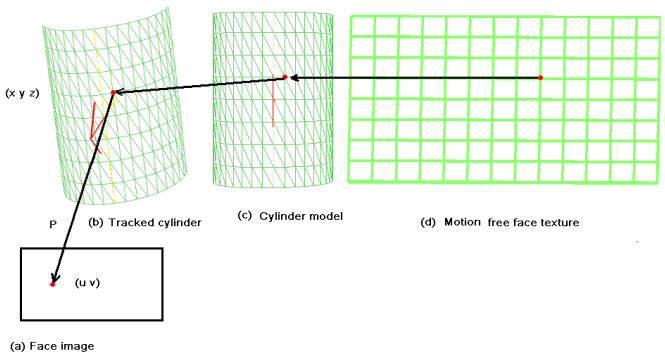


Figure 2. Cylinder Tracking Result.

corresponding location on the tracked cylinder ((b) in Figure 2) and finally estimating the value of the pixel from the original face image ((a) in Figure 2). In the ideal case of perfect tracking, the texture map is stabilized from global motion and produces a frontal face that is centered horizontally in the unwrapped image.

An example tracking result using the CHM is shown in Figure 3 in which a single person is tracked from two cameras. Each row of the figure represents a single from number from both cameras. The images in Figures 3(a), 3(e), 3(i) and 3(m) are from camera A while the images in Figures 3(d), 3(h), 3(l) and 3(p) are from camera B. The image pairs in the center of Figure 3 are the unwrapped cylinder images from their corresponding CHM in the original images.

3. Fusing Face Recognition from Multiple Cameras

Incorporating the results of multiple cameras viewing a common subject may increase the accuracy and robustness of the face recognition task. In this paper, the face recognition results (on a set of full faces or average-half-faces) of multiple cameras are fused. Since eigenfaces is used along with NN for the face recognition task, a distance between the projected testing weights and the projected training weights is calculated. Let us define camera A as the camera that views mostly the right half of the face and camera B as the camera that views mostly the left half of the face. Considering the case of a two-camera system, at every time t, there will be a frame from camera A that corresponds to a frame from camera B. Therefore, each frame will have a minimum distance calculation that will be used to assign the classification result of the face in each frame. Also, by using the CHMs to track the face in each frame, an estimate of the pose of the face (only yaw in our case) is calculated. These two pieces of information (distance to classified training sample and pose estimation) are used in the combination of results from multiple cameras. We present results using 5 different methods for fusing the results between the two cameras:

1. Independent (Ind). Each face is recognized independently as if it was from a single camera, so no combination of the two cameras is used.

2. Minimum distance (MinDist). Between the two cameras, the camera with the minimum NN distance is chosen as the classification result.

3. Best pose (BestPose). The camera with the most frontal pose is used for the classification result.

4. Multiplier weights (MultWts). The NN distance and pose information of the two cameras are multiplied along with a constant and the minimum of this new distance is used for the classification result.

5. Gaussian weights (GaussWts). The pose of each of the cameras is used to produce a Gaussian weight which is then applied to the NN result. The minimum of this result is used for recognition.

The distance from the testing image weights to the nearest training image weights gives some measure of how close the two samples are in "face space". The estimated pose of the face gives a measure of the ability to recognize the subject's face correctly. For instance, a full frontal face should be easier to classify than a non-frontal face with a yaw of 30° or more. Each method of fusing the results requires a straightforward calculation that may have dramatic accuracy gains to the multi-camera face recognition system as discussed in the experiments section.

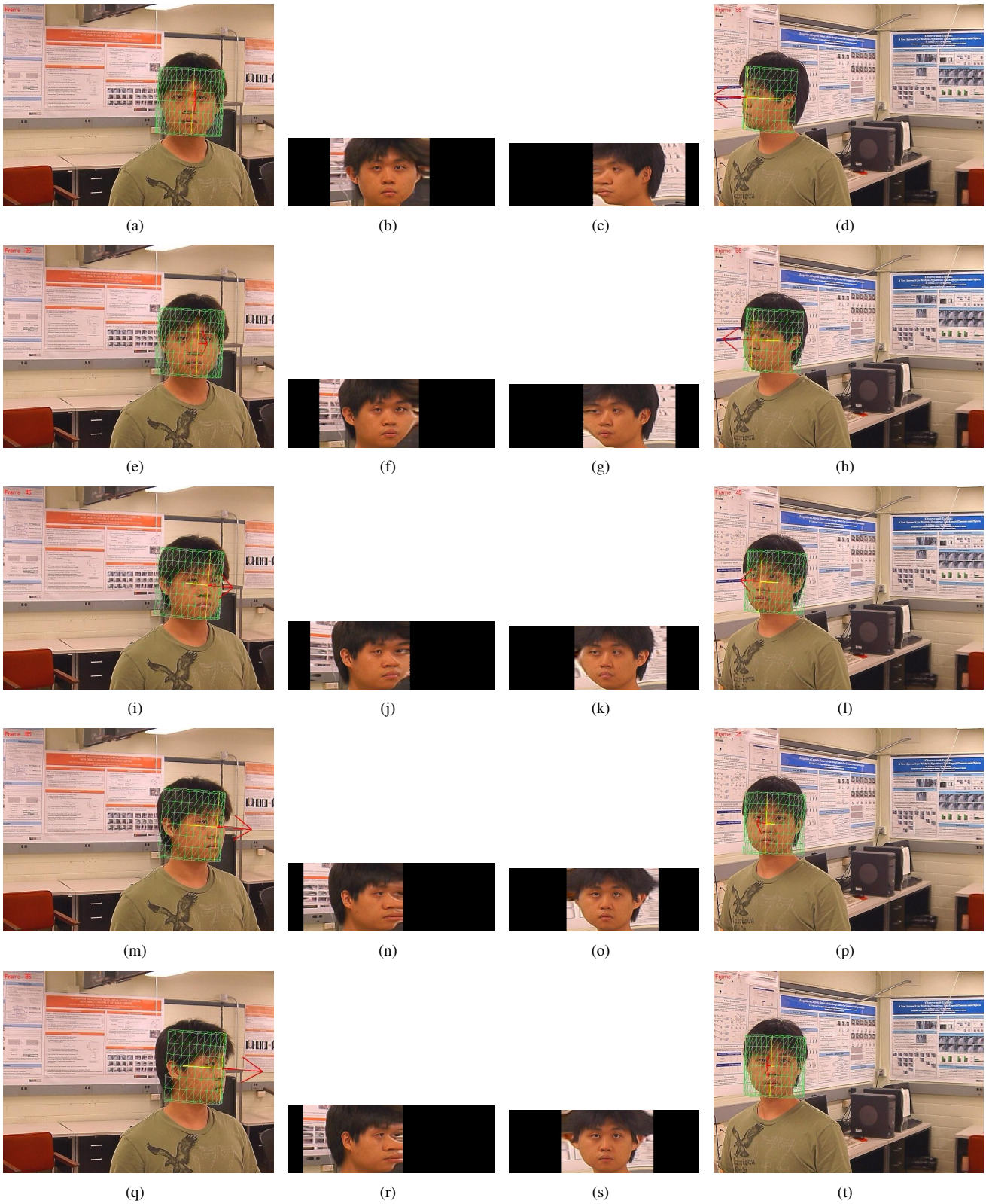
A brief description of each of these fusion methods is presented.

3.1. Independent

This result assumes that there is no information to share between the two cameras. Each face is recognized on its own independent of pose or camera location. This is used as a baseline for the other results.

3.2. Minimum Distance

Since we will get a NN distance calculation for each of the cameras, the simplest way to fuse the results from the cameras using only the NN distances and choose the minimum of the two for classification. Therefore, the label of the training image of the camera with the minimum NN distance is chosen for the recognition result of the two cameras combined.



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

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

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 3. CHM Tracking and Scanning Result on 5 Pairs of Images from the Two Cameras.

3.3. Best Pose

Using only the calculated pose information of each of the cameras from the CHM model, we can fuse the recognition results by simply choosing the camera with the most frontal pose. Therefore, the fused recognition result is the label applied to the most frontal face image of the two cameras.

3.4. Multiplier Weights

The first attempt to combine both the pose information and the NN distance from each of the cameras is the simplest. We form a normalized pose from the pose calculation of cameras A and B (P_A and P_B, respectively) by dividing it by the maximum pose ($Pnorm_A = \frac{|P_A|}{max(P)}$ and $Pnorm_B = \frac{|P_B|}{max(P)}$). Then we multiply the NN distances from each of the cameras by the normalized poses and use the minimum result for classification.

3.5. Gaussian Weights

In the final fusing of recognition results between the two cameras, a Gaussian weighted approach is used. Let d_A be the Euclidean distance between the projected weights from the face tracked in camera A to the nearest training sample's projected weights and let d_B be similarly defined. Let P_A and P_B be the estimated pose of the face (where a frontal face image has a yaw of 0°) from the CHM calculated from the frames in camera A and B respectively. Weights w_A and w_B are calculated for each pair of frames using the pose estimations and a Gaussian function as

$$\begin{aligned} w_A &= 1 - \frac{1}{\sigma\sqrt{2\pi}} e^{-(Pnorm_A)^2/2\sigma^2} \\ w_B &= 1 - \frac{1}{\sigma\sqrt{2\pi}} e^{-(Pnorm_B)^2/2\sigma^2} \end{aligned} \tag{6}$$

where $Pnorm_A = \frac{|P_A|}{max(P)}$ and $Pnorm_B = \frac{|P_B|}{max(P)}$) are the ratios of estimated poses to the maximum pose, the mean of the Gaussian was chosen to be zero (frontal pose) and $\sigma = 0.01$, which favors frontal poses and penalizes poses that are non-frontal. The maximum pose ($max(P)$) calculated from our video was 70°. The calculated weights are then multiplied to produce new distance measures D_A and D_B by

$$\begin{aligned} D_A &= d_A * w_A \\ D_B &= d_B * w_B. \end{aligned} \tag{7}$$

The resulting modified distances are then used for classification. If $D_A \leq D_B$, the training label applied to the face from camera A is applied to the face in the camera A and camera B pair. Otherwise the label from camera B is chosen.

4. Building Confidence by Aggregating Results

One contribution we have made in this paper is to use face recognition results from several frames of a successfully tracked subject to give a better recognition result with a certain level of confidence. In the videos used in this paper, each video sequence has only one subject, so the successful tracking of the face is made simpler. However, one could imagine a scenario in which several subjects are tracked in the same video sequence and the tracking results of each subjects' faces are successful. The results of the face recognition on each of the pair of frames of a single subject's tracking result using the fused results method explained in Section 3 are aggregated together. This aggregation of results is a simple scoring process. For instance, if 100 pairs of frames are labeled as subject 2 and 50 pairs of frames are labeled as subject 1 in a 150 frame tracking sequence, we would label the subject of the track as subject 2 with a confidence of $100/150$, or 66.7%. This method gives intuitively positive results based on the assumption that the more frames you have of an individual, the more confidently the correct identification will be applied.

5. Still Face Recognition Method

An oval mask is first applied to a face image that has been successfully tracked by the CHM so that background noise is removed for the recognition process. Two examples of the application of this mask to the face images are found in Figures 4(a) and 5(a). Once the mask has been applied to every training and testing image in the database, recognition of the faces can proceed.

5.1. The Average-Half-Face

The authors in [1] introduce the concept of the average-half-face (AHF) applied to face recognition. In summary, the average-half-face is a preprocessing step applied to a still full face image. The full face must first be centered by locating the bilateral symmetry axis of the face. Then, computing the average-half-face is equivalent to splitting the face image into two, flipping one of the images horizontally and then averaging the two resulting half-faces pixel-wise.

The average-half-face is attractive in this application for several reasons. It has been shown to be successful along with eigenfaces for recognition [1], which is the recognition algorithm this paper is utilizing. The average-half-face reduces noise and errors from the transformation to the unwrapped cylinder image. In some cases the face recognition accuracy of the system is increased when using the average-half-face instead of the full face for recognition.

The result of the average-half-face is shown on two examples from the video sequence of the same subject. The image in Figures 4 and 5 are from cameras A and B, respectively and correspond to the same frame number in the

(a) Full Face (b) AHF

Figure 4. Average-Half-Face (b) Result of Centered Full Face (a).

(a) Full Face (b) AHF

Figure 5. Average-Half-Face (b) Result of Unaligned Full Face (a).

sequence. The resulting images in Figure 4 appear to be fit for the recognition task. However, the full face in Figure 5(a) is clearly not desirable. The average-half-face present in Figure 5(b) is likely a better representation for the recognition task.

5.2. Eigenfaces

Eigenfaces [5] is used for recognition for mainly two reasons. We are interested in transforming the problem of face recognition from video to a still face recognition problem so that any still face recognition algorithm can be used. Also, the implementation of the eigenfaces algorithm is well studied and needs little discussion.

Nearest-neighbors (NN) is used to classify the test weights to the nearest training sample using the Euclidean distance between the weights.

6. Experimental Results

Video sequences taken from two cameras with unique views of the face are used for the experiments. Each subject varied the pose of their face in each of the video sequences by changing the yaw (or pan) from 0° to $\pm 70^\circ$. Example images from our video sequences are shown in Figure 3. This variation in pose produced around 100 frames per subject from each of the cameras which were used as test data. Two frontal images per subject are used for the training data.

Two face recognition experiments are performed on the two-camera video sequences. In both experiments, the face in the video is tracked using CHMs and the estimated pose is returned for each frame. Eigenfaces is applied to the cropped faces tracked in each frame directly in the first experiment. This is used in comparison to our methodology in the second experiment of using an unwrapped cylinder face image. In each experiment, the full face and the average-half-face were used along with the first 12 eigenfaces of the training data. The Euclidean distance was used as the distance measure for nearest-neighbor (NN) classification. These two experiments are used along with the 5 methods for fusing the recognition results described in Section 3 for a total of 20 results (5 methods times 2 experiments times 2 sets of images; full face and average-half-face).

Table 1. Face Recognition Results.

	Full Orig	AHF Orig	Full CHM	AHF CHM
Ind	72.6%	67.7%	67.4%	68.5%
MinDist	82.2%	78.0%	94.4%	93.5%
BestPose	76.2%	75.2%	91.5%	92.5%
MultWts	75.9%	75.5%	93.5%	94.0%
GaussWts	81.9%	78.3%	94.4%	93.5%

The results of the experiments are displayed in Table 1. The results from the original 2D cropped face images from the tracking results are displayed first under "Full Orig" and "AHF Orig", for the experiments using the full face and the average-half-face, respectively. The results gathered by using the faces generated by the CHM tracking and cylinder unwrapping for the full face and the average-half-face are denoted "Full CHM" and "AHF CHM". The "Ind", "MinDist", "BestPose", "MultWts", and "GaussWts" results are those generated by fusing the recognition results from each of the cameras by using the NN distance and/or pose information as described in Section 3. The highest accuracy reported is 94.4% by using the full face along with the faces generated by the CHMs and the fused results from both cameras with methods "MinDist" and "GaussWts".

As mentioned in Section 4, one can use multiple frames from a successfully tracked face to build confidence in the recognition result. In the experiment using the CHM tracked faces and the fused results (using the GaussWts method), we achieve 100% accuracy for all subjects with an average confidence of 93.8% and 95% for the full face and the average-half-face respectively.

7. Discussion

Clearly, the idea of fusing the recognition results from both of the cameras is more successful than independently recognizing the results. This fusing method alone is responsible for an increase in accuracy of 4% to 25% in our experiments. Using CHMs to track the face and produce an unwrapped cylinder face image for recognition has be shown to outperform the original 2D face captured from the video frames by almost 20%, but only in the case of fusing the recognition results. However, to achieve the full face

recognition result of 94.4%, both methods were necessary. The use of the average-half-face achieved an accuracy of 94.0%, which could be further improved by calculating the bilateral symmetry axis of the face (a usual step in the computation of the average-half-face). The two most successful methods for fusing the recognition results between the two cameras appear to be the minimum distance and Gaussian weights methods. We suspect that with a larger database, the minimum distance method alone would tend to produce a lower accuracy than using the Gaussian weighted method which uses pose information, but this needs to be tested. The further step of aggregating recognition results of a subject that has been successfully tracked in video produces 100% recognition on the all of the subjects tracked in our video sequences with most confidence levels above 90%. It is possible that with a faster frame rate and/or longer video sequences, the confidence of the recognition results could be further improved. A worthy contribution of the average-half-face in these results is that it increases the confidence of the aggregated recognition results.

8. Conclusion

Face recognition from video presents many challenges such as self-occlusion, occlusion from objects, and illumination changes. We present a method to overcome the problems of self-occlusion and lack of frontal face images in video by using CHMs to produce an unwrapped cylinder face image and the estimated pose of the face. Using these outputs we fuse the face recognition results of both cameras, which results in a dramatic increase in accuracy. The average-half-face is used to reduce the errors in scanning and unwrapping the CHM to a 2D face image. Eigenfaces is used for the face recognition task. The proposed method achieves an accuracy of 94.4%. By further aggregating recognition results from a successfully tracked individual, a recognition rate of 100% is achieved with high confidence.

Future work includes improving the CHM tracking so that initialization using a frontal face is avoided as well as improving the quality and pose of the unwrapped cylinder image. Also, improving the tracking method to better find the middle of the face might lead to more accuracy when using the average-half-face for the face representation. Perhaps the use of a 3D mesh model would improve the transformation from a non-frontal pose to a frontal face image, however, this could require more computation. Generalizing the fusing method to other face recognition algorithms is desired. One could also combine the unwrapped images from the CHMs in each camera in a more direct manner to reconstruct a full face image for recognition. Further, we will test the proposed methodology on a large database of subjects in a multi-camera setting.

References

[1] J. Harguess and J. K. Aggarwal. A case for the average-half-face in 2D and 3D for face recognition. In *IEEE Computer Society Workshop on Biometrics (in conjunction with CVPR)*, June 2009.

[2] 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.

[3] Y. Li, S. Gong, and H. Liddell. Constructing facial identity surfaces for recognition. *International Journal of Computer Vision*, 53(1):71–92, 2003.

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

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

[6] 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.

[7] Z. Zhang, Z. Liu, D. Adler, M. F. Cohen, E. Hanson, and Y. Shan. Robust and rapid generation of animated faces from video images. *International Journal of Computer Vision*, 58(2):93–119, 2004.

[8] S. Zhou, V. Krueger, and R. Chellappa. Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding*, 91:214–245, 2003.