

Unsupervised Learning from Local Features for Video-based Face Recognition

Ajmal Mian

School of Computer Science and Software Engineering
The University of Western Australia

ajmal@csse.uwa.edu.au

Abstract

This paper presents an unsupervised learning approach to video-based face recognition that does not make any assumptions about the pose, expressions or prior localization of landmarks on the faces. The proposed algorithm exploits spatiotemporal information obtained from local features that are extracted from arbitrary keypoints on faces as opposed to pre-defined landmarks. The algorithm is inherently robust to large scale occlusions as it relies on local features. During unsupervised learning, faces from a video sequence are automatically clustered based on the similarity of their local features and a voting-based algorithm is employed to pick the representative features of each cluster. During recognition, video frames of a probe are sequentially matched to the clusters of all individuals in the gallery and its identity is decided on the basis of best temporally cohesive cluster matches. The proposed algorithms can also detect sudden identity changes in video by utilizing the temporal dimension. The algorithm was tested on the Honda/UCSD video database and a maximum of 99.5% recognition rate was achieved.

1. Introduction

Numerous physiological (e.g. iris, fingerprints) or behavioural (e.g. voice, gait) biometrics can be used for human identification. However, biometrics which can be acquired non-intrusively and without the knowledge of the subject are of special interest due to their potential use in security applications. The human face is one of the most attractive biometrics for this purpose. However, machine recognition of faces is extremely challenging not only because the distinctiveness of facial biometrics is comparatively low [5] but because there are a number of factors over which there is little or no control. These factors include changing illumination, pose, facial expressions, facial ornamentation and occlusions.

Initial face recognition research was based on matching single pairs of images [15] [1]. However, such recogni-

tion techniques do not cope well with the above challenges. More recently, recognition from 3D facial scans has been explored by many research groups [3] [4] [11] [12] [13]. The main limitation of 3D face recognition lies in the 3D scanning part. Compared to cameras, 3D scanners are more expensive, have lower resolution and slower acquisition time. Even though 3D scanners are continuously improving on these three factors, they will always lag behind cameras. For comprehensive surveys of face recognition from images and 3D scans, the reader is referred to [16] [2].

During the past few years, many research groups have developed interest in video-based face recognition because video cameras are commonly available and provide more information compared to still cameras. Moreover, motion helps in the recognition of faces [16]. Early video-based face recognition algorithms were frame-based. They matched individual frames from the training and test videos, and made the decision using a voting or averaging criterion. These techniques do not fully exploit the spatiotemporal information. More promising techniques match video sequences and use temporal coherence between the query and database videos in addition to the spatial information contained in individual frames. Video-based face recognition algorithms can have three possible learning modes, namely offline batch learning, online learning and hybrid learning. In batch learning the classifier is trained offline once only [6]. The system is not updated unless a new identity is to be added in the database. In online learning, the system is completely trained online [9], however manual labeling of identities is still required. The hybrid learning approach learns generic (or specific) face models offline in a batch mode and continuously updates them during online recognition [7].

Many video-based face recognition algorithms assume a prior knowledge of the pose and identification of pre-defined facial landmarks [8]. Others rely on supervised learning whereby each frame is manually assigned to its corresponding pose cluster [7]. Even though batch learning is an offline process, manual labeling could be very laborious and time consuming due to the bulk (up to 25

frames/second) of video data.

Most video-based face recognition algorithms use the complete detected face to extract global features. Global features are sensitive to registration errors, occlusions and pose variations. Local features have proved their superiority in image and 3D face recognition algorithms. However, local features have not achieved much attention in video-based face recognition because of the excessive amount of global data already available due to the temporal dimension. Local features add complexity by introducing yet another dimension, however they are important as they are robust to occlusions and provide an additional cue for accurate pattern recognition. Sivic et al. [14] used local features for retrieving different shots of a person from a movie. However, they extracted local features from pre-defined landmarks on the face. This simplifies the dimension problem as the features from the same landmarks can be compressed by projecting then to the PCA subspace [14].

This paper proposes a fully automatic video-based face recognition algorithm which performs unsupervised learning in batch mode. The frames of an input video sequence are automatically clustered during the learning phase. The proposed algorithm uses local features extracted from unordered keypoints as opposed to pre-defined landmarks [14]. A voting scheme is used for picking the representative features and frame from each cluster. During recognition, local features from a probe face are matched with the cluster representative features and a compound frame similarity measure is used for making the final decision.

2. Local Features

The SIFT (Scale Invariant Feature Transform) [10] is used in this paper for extracting local features. However, the proposed algorithm is generic and is not tied up to specific features. An advantage of the proposed algorithm is that it does not impose any restrictions on the location of features. They can be extracted from any point on the face and need not be ordered.

SIFTs [10] are 128 dimensional unit vectors extracted at keypoints in an image. The keypoints do not conform to any specific landmarks (e.g. eye corners) on the face but are detected at the scale space extrema in the Difference-of-Gaussian function convolved with the image. To qualify as a keypoint, the points must also satisfy other conditions including high contrast, good localization along an edge and principal curvature ratio of above a threshold.

At each keypoint, a histogram is formed from the local gradient orientations weighted by their magnitudes and by a circular Gaussian window. Dominant gradient directions are used to extract SIFT making it rotation invariant. Sample regions of 4×4 are used to create orientation histograms with eight bins forming a 128 dimensional vector. For illumination robustness, the vector is normalized to unity,

thresholded to a ceiling of 0.2 and finally renormalized to unit length.

3. Unsupervised Learning

The Honda/UCSD first dataset [6] was used for experiments in this paper. During the learning phase, faces are detected in the training video sequence. Note that face detection is outside the scope of this paper and prior detection is assumed. The detected faces are cropped and normalized with respect to scale and illumination. The proposed approach is robust to the scale and location of the cropping window because it extracts *scale invariant* features at automatically detected keypoints which are independent of the position of the cropping window. A scale of 150×150 was used in this paper and simple histogram equalization was used for illumination normalization. For each normalized face, SIFTs were calculated and matched as discussed in Section 3.1. The matching process resulted in a similarity matrix which was used to cluster the faces (Section 3.2) in a training video sequence.

3.1. Face Matching

For a given training video sequence of an identity, every face is matched to every other face in order to construct a $N \times N$ (where N is the number of frames) similarity matrix. Since the matrix is symmetric, only $\frac{N(N-1)}{2}$ entries need to be calculated. The similarity between two faces is determined by matching their respective SIFT features using the equation

$$e = \cos^{-1}(\mathbf{f}_a \mathbf{f}_b^T), \quad (1)$$

where \mathbf{f}_a and \mathbf{f}_b correspond to the SIFT features from face a and b respectively. The pairs of SIFTs which had the minimum error e were considered matches and only one-to-one matches were allowed. For example, if a feature in face b turned out to be the best match to more than one feature in face a , only the one with the minimum value of e was considered as its match. Moreover, a distance constraint was used to avoid matching SIFTs from far off points in the two face images. Due to these constraints, different faces ended up with a different number of SIFT matches. The overall similarity of the two faces was determined by normalizing the average error \bar{e} between their matching pairs of SIFTs and the total number of matches m . Both measures were normalized on the scale of 0 to 1 and combined using a weighted sum rule.

$$\bar{e}'_i = \frac{\bar{e}_i - \min(\bar{e}_i)}{\max(\bar{e}_i - \min(\bar{e}_i)) - \min(\bar{e}_i - \min(\bar{e}_i))}, \quad (2)$$

$$m'_i = \frac{m_i - \min(m_i)}{\max(m_i - \min(m_i)) - \min(m_i - \min(m_i))}, \quad (3)$$



Figure 1. Sample clustered faces. Each row contains a different cluster. Notice that faces with similar facial expression are also clustered together (fifth row).

$$s_i = \frac{1}{2}(w_e \bar{e}'_i + w_m(1 - m'_i)), \quad (4)$$

where $i = 1 \dots N$ and w_e, w_m are the corresponding weights given to the normalized average error \bar{e}' and the normalized number of matches m' . Note that the \bar{e}'_i has a negative and m'_i has a positive polarity which is why m'_i is subtracted from 1. The polarity of the final similarity measure s_i is also negative i.e. lower values mean higher similarity.

3.2. Frame Clustering

The above matching process results in an $N \times N$ symmetric matrix of similarity measures which is used to automatically cluster the N frames. Hierarchical clustering with mean similarity distance was used in our experiments. The total number of clusters per video sequence (and hence identity) was empirically chosen to be 20. This number was chosen keeping in view that there are three degrees of freedom in pose, five common facial expression types and that the variation in pose and expression can occur in many possible combinations e.g. pitch + yaw + smile. Another possibility was to use a threshold for determining the output number of clusters instead of keeping it fixed. However, a fixed number of clusters was chosen to avoid biasing in the database identities. Fig. 1 shows sample faces from 10 different clusters of the same identity. Notice that the clustering is quite accurate even though the faces are not perfectly registered. Another interesting outcome of automatic clustering (unsupervised learning) is that faces with simi-

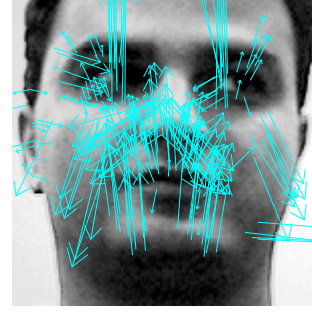


Figure 2. Representative SIFT features of a cluster drawn on the cluster representative face. The direction of the arrow points towards the orientation and its length represents the magnitude of the gradient.

lar expressions have been assigned to the same and unique clusters even though they share the same pose with other clusters.

3.3. Selection of Cluster Representatives

One of the motivations behind clustering is data compression whereby each cluster is represented by its subset. Global features-based algorithms or the ones that extract local features from pre-defined landmarks, generally use the mean features as representatives of clusters. However, this is not possible in the proposed algorithm as the local features are extracted from arbitrary keypoints as opposed to pre-defined landmarks. Therefore, a voting scheme was used to select the representative local features from each cluster as follows. Within each cluster, the matching pairs of features found as a result of the face matching process described in Section 3.1 for all combinations of two faces were given a vote each. A stable feature from a frame is more likely to match a feature in another frame and hence receive more votes compared to an unstable feature that might appear in a few frames due to noise and never get repeated. For each cluster, the top n features that received the maximum votes were selected as the representatives of the cluster. It is interesting to note that the representative features of a cluster came from different locations of different faces (frames) within the cluster. This ensures that the keypoints sufficiently cover the face and represent the whole cluster.

In addition to the representative features, the face whose local features get the maximum accumulative votes is selected as the cluster representative. This representative face is currently not used, however it can be useful for a global features-based classifier that runs in parallel to the existing one in order to increase the accuracy of recognition. Fig. 2 shows cluster representative features marked on the cluster representative face. Notice that the features come from different locations and mostly from important landmarks like the eyes, nose, lips and chin.

Note that the keypoint detection process of SIFT gener-

ally finds different numbers of keypoints in different frames, hence biasing the matching process in favor of frames with more features. This is not critical in frame clustering as all frames belong to the same video sequence and are therefore acquired in similar illumination conditions. However, this biasing could be critical during recognition when different video sequences are matched. Selecting a fixed n number of features for each cluster removes the biasing due to the number of features as well as the number of frames per cluster. In this paper, n was fixed at 200.

4. Online Recognition

For recognition, a separate set of test videos was used. Face was detected in each input frame from the test video and normalized as described in Section 3. SIFT features were then calculated for the face and matched with the cluster representative features of all identities in the database. The same matching algorithm was used as described in Section 3.1. The mean SIFT error \bar{e} and number of matches m were normalized on the scale of 0 to 1 (Eqn. 2 and Eqn. 3) for the clusters of all identities and then combined using Eqn. 4. Each test video results in a three dimensional similarity matrix of size $N \times G \times C$ where N is the number of input frames of the test video, G is the total number of identities in the database (also referred to as gallery) and C is the number of clusters. It is important to understand this matrix as it contains the complete recognition information. The first dimension i.e. N is basically the temporal dimension which increases at the frame rate (15 frames/second with the database used). Suppose the system is initialized at t_0 . Then at any instant t in time, there is a total of $15(t - t_0)$ frames available for making the recognition decision. However, a recognition system must limit the number of frames that it uses because identities could change in videos. More details on this are given in Section 5.2.

The last two dimensions were fixed because there were 20 identities in the database and each identity had 20 clusters. Many options are available to get an overall similarity score of each identity with a given frame, for example mean cluster similarity or minimum cluster similarity. The latter gave better results and was used in the experiments.

5. Results

Two types of experiments were conducted. In the first one, a comparison was carried out between the performance of two different schemes for combining the frames along the temporal dimension (Section 5.1). In the second experiment, the effects of identity changes in a video sequence were studied (Section 5.2).

5.1. Temporal Face Recognition from Video

Video-based face recognition is a sequential process where every incoming frame adds to the information provided by the previous frames. In other words, each new frame, when matched to the database, gives a $G \times C$ matrix which can be concatenated to the end of the existing $N \times G \times C$ matrix from the previous N frame matches. However, system memory is finite and old data must be discarded after a certain time. This was done by considering only the last f number of frames. Two different schemes were used for the purpose of combining the results of the last f frames. The first one, referred to as batch temporal recognition, combines the similarity of the last f frames in a batch mode i.e. the original similarity scores are averaged over the last f frames. Thus,

$$s'_i = \frac{1}{f} \sum_{i-f+1}^i s_i, \quad (5)$$

where s'_i is the similarity score averaged over the last f frames and s_i is the original similarity score of frame i . As discussed in the previous section, s_i refers to the minimum cluster similarity of each identity in the database and they are averaged separately for each identity. The second technique, referred to as compound temporal recognition, on the other hand, combines the similarity scores in a compound fashion i.e. the average similarity scores of the last f frames are averaged again. Thus,

$$s'_i = \frac{1}{f} \sum_{i-f+1}^i s'_i. \quad (6)$$

Fig. 3 shows the recognition rate versus the number of frames used for the above two methods. The recognition rate corresponds to the number of frames correctly recognized divided by the total number of frames in the test videos. The compound temporal recognition performs much better than the batch recognition by achieving a maximum of 99.55% recognition rate. There were no changes in identity in a single video sequence for this experiment.

5.2. Effects of Identity Changes

Notice that the compound temporal recognition gets biased towards an identity with the passage of time. This is likely to result in a lag in the recognition process when there is a change in identity in the test video. The batch temporal recognition will also have a lag but it will be limited to about $0.5f$ frames. However, in the compound case, the lag will be much greater. The minor drop in compound temporal recognition rate after frame 15 in Fig. 3 is also due to this phenomenon where the effect of incorrectly recognized frames gets distributed over many frames. When there is a

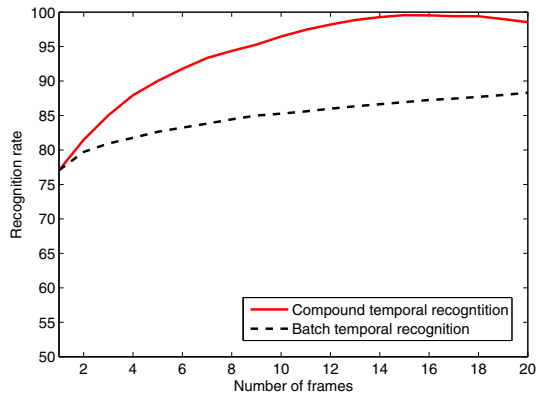


Figure 3. Plot of the recognition rate versus the number of frames used. The recognition rate of compound temporal recognition rapidly increases and reaches its peak at 15 frames and then drops because the effect of misclassified frames gets distributed over more frames.

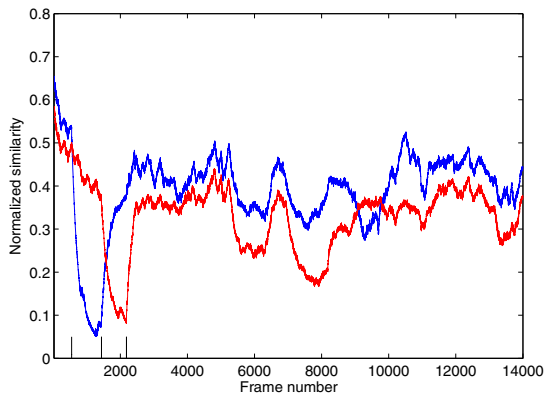


Figure 4. Similarity plots of the second and third database identities with a video sequence containing multiple unknown identities. The two slumps show where the second and third identities were found in the video sequence. The small vertical lines mark the first three ground truth locations where the identity changes in the video sequence. Notice that the slope of the similarity plots changes abruptly after the identity change.

change in identity, this effect is more prominent and is directly proportional to how frequently the identity changes in the video sequence.

To study the effects of identity change, the test videos of the Honda/UCSD first dataset were concatenated and compound temporal recognition was performed on the resulting video sequence. Fig. 4 shows the plots of similarity scores of the second and third database identities with the test video. Recall that a smaller value means more similarity. The two slumps in the plots indicate where the second and third identities were correctly recognized in the test video. Also notice that the slope of the similarity plot changes abruptly when the identity changes. Fig. 5 shows

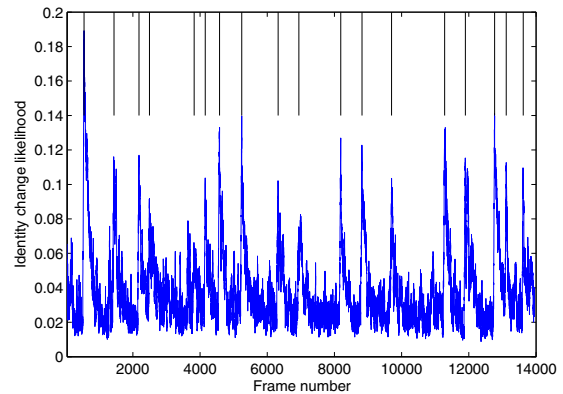


Figure 5. Plot of the difference between the maximum and minimum slopes (vertical axis) of database identities for each frame match (horizontal axis). The vertical lines at the top mark the ground truth locations where the identity changes in the video sequence. Notice that the plot accurately detects identity changes in the video sequence.

a plot of the maximum minus the minimum slope of the similarity values of the database identities for each frame in the concatenated video sequence. The peaks in the plot correspond to the detected identity changes. Comparing these peaks to the ground truth locations (marked by vertical lines at the top) of identity changes, it is possible to see that the identity changes are correctly identified using the maximum minus minimum similarity slope measure. Once an identity change is detected, the system can be reset to remove the lag in recognition.

6. Discussion and Future Work

Facial biometrics are not as powerful as fingerprints, iris and retina. While the answer to which features and classifiers give the best face recognition performance is debatable, many researchers agree that multimodal techniques will always outperform unimodal ones. Multimodal face recognition may use multiple sensors (e.g. visible, IR, range images), multiple features (e.g. global, local, semi-local) or multiple classifiers. It may also use a combination of all three multiple modalities. While the algorithm proposed in this paper may not outperform all existing techniques, it certainly makes a valuable contribution to the video-based face recognition literature because it achieves good results without making any assumptions, without the need for supervised learning and without imposing restrictions of pre-defined landmarks for the extraction of local features. Moreover, the proposed algorithm provides additional information that can help other classifiers e.g. it automatically clusters the training video frames and selects cluster representative faces. Moreover, it also provides one-to-one correspondences between the faces in each cluster and be-

tween the test and training frames which could be used for registration of the faces by global feature-based face recognition algorithms which must perform a prior registration of the faces in order to achieve good results.

An interesting phenomenon to be noted in Fig. 4 is that apart from the two slump locations, the two curves follow more or less a similar pattern. Deviation of a curve, corresponding to a database identity, from the pattern gives important information about the identity appearing or disappearing from the test video. Currently, this information is not used by the proposed algorithm and will be considered for future work.

7. Conclusion

This paper presented a fully automatic video-based face recognition algorithm. The offline learning phase of the algorithm is also automatic as it performs unsupervised learning. The proposed algorithm does not assume prior knowledge of the pose or the identification of pre-specified landmarks. In fact, it extracts features from automatically detected keypoints which could be in any order and need not correspond to specific landmarks on the face. The algorithm uses a compound temporal averaging of the similarity scores to accurately establish the unknown identity in a test video. Moreover, it can easily detect changes in identity by observing the behavior of the similarity curves of all identities in the database. Experiments were performed using the Honda/UCSD first dataset and a maximum of 99.5% recognition rate was achieved.

Acknowledgments

Thanks to UCSD for providing the data and D. Lowe for providing the SIFT code. This research is sponsored by ARC Discovery grant DP0881813 and partly by UWA Research Grant 2007.

References

- [1] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Trans. on PAMI*, 19:711–720, 1997.
- [2] K. Bowyer, K. Chang, and P. Flynn. A Survey Of Approaches and Challenges in 3D and Multi-modal 3D + 2D Face Recognition. *CVIU*, 101(1):1–15, 2006.
- [3] K. Chang, K. Bowyer, and P. Flynn. Multiple Nose Region Matching for 3D Face Recognition under Varying Facial Expressions. *IEEE Trans. on PAMI*, 28(10):1695–1700, 2006.
- [4] T. Faltemier, K. Bowyer, and P. Flynn. A Region Ensemble for 3-D Face Recognition. *IEEE Trans. on Information Forensics and Security*, 3(1):62–73, 2008.
- [5] A. K. Jain, A. Ross, and S. Prabhakar. An Introduction to Biometric Recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.
- [6] K. Lee, J. Ho, M. Yang, and D. Kriegman. Visual tracking and recognition using probabilistic appearance manifolds. *CVIU*, 99(3):303–331, 2005.
- [7] K. Lee and D. Kriegman. Online Probabilistic Appearance Manifolds for Video-based Recognition and Tracking. In *CVPR*, volume 1, pages 852–859, 2005.
- [8] Y. Li, S. Gong, and H. Liddell. Constructing Facial Identity Surfaces in a Nonlinear Discriminating Space. In *CVPR*, volume 2, pages 258–263, 2001.
- [9] L. Liu, Y. Wang, and T. Tan. Online Appearance Model Learning for Video-Based Face Recognition. In *CVPR*, pages 1–7, 2007.
- [10] D. Lowe. Distinctive Image Features from Scale-invariant Key Points. *IJCV*, 60(2):91–110, 2004.
- [11] X. Lu, A. K. Jain, and D. Colbry. Matching 2.5D Scans to 3D Models. *IEEE Trans. on PAMI*, 28(1):31–43, 2006.
- [12] A. Mian, M. Bennamoun, and R. Owens. An Efficient Multimodal 2D-3D Hybrid Approach of Automatic Face Recognition. *IEEE Trans. on PAMI*, 29(11):1927–1943, 2007.
- [13] G. Passalis, I. Kakadiaris, and T. Theoharis. Intra-class Retrieval of Nonrigid 3D Objects: Application to Face Recognition. *ACM Computing Survey*, 29(2):218–229, 2007.
- [14] J. Sivic, M. Everingham, and A. Zisserman. Person Spotting: Video Shot Retrieval for Face Sets. In *CIVR*, 2005.
- [15] M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.
- [16] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face Recognition: A Literature Survey. *ACM Computing Survey*, 35(4):399–458, 2003.