Detecting Questionable Observers Using Face Track Clustering

Jeremiah R. Barr, Kevin W. Bowyer and Patrick J. Flynn
Department of Computer Science and Engineering
University of Notre Dame
Email: {jbarr1,kwb,flynn}@nd.edu

Abstract

We introduce the questionable observer detection problem: Given a collection of videos of crowds, determine which individuals appear unusually often across the set of videos. The algorithm proposed here detects these individuals by clustering sequences of face images. To provide robustness to sensor noise, facial expression and resolution variations, blur, and intermittent occlusions, we merge similar face image sequences from the same video and discard outlying face patterns prior to clustering. We present experiments on a challenging video dataset. The results show that the proposed method can surpass the performance of a clustering algorithm based on the VeriLook face recognition software by Neurotechnology both in terms of the detection rate and the false detection frequency.

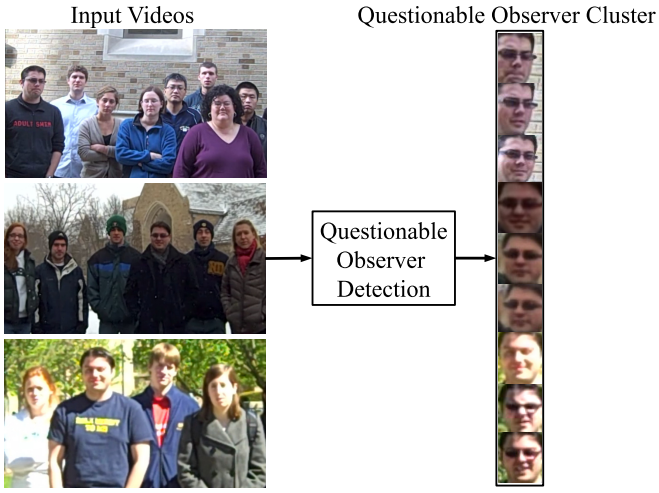

Figure 1. A solution to the questionable observer detection problem is a collection of face track clusters, each of which represents a distinct individual and contains face tracks from multiple videos.

1. Introduction

A potentially useful application of automatic face recognition technology is that of analyzing videos of crowds observing the aftermath of criminal activities such as arson. Informants, accomplices or culprits may observe a collection of related crime scenes; this tendency could indicate their involvement in a series of offenses. We thus call these individuals *questionable observers*. In contrast, we call individuals that observe relatively few scenes of this nature *casual observers*. The distinguishing feature that separates the questionable observers from the casual observers is the percentage of videos in which they appear. We propose an algorithm for the *questionable observer detection problem*, which is to differentiate questionable observers from casual observers.

Whereas much of the work on face recognition from video assumes that an identification algorithm has prior knowledge about the persons to recognize and typically focuses on the recognition of one subject at a time, the problem posed here requires the use of unsupervised classification techniques to aggregate images of the same face observed in different crowd videos. Identifying information is lost when crowd members occlude one another, which makes both tracking and classification more difficult. Variations in pose, illumination, and facial expression throughout a single video and between different videos can affect face appearance and, hence, complicate questionable observer detection as well. Finally, the video evidence may be recorded by camera phones or surveillance cameras and so the quality of the face image sequences can be very low.

We propose an unsupervised classification algorithm for detecting questionable observers that begins by merging face image sequences, *i.e. face tracks*, that correspond to the same individual and come from a particular video. We then remove outlying face images from the merged face tracks based on the observations that certain head poses and facial expressions are more likely than others [2, 3]. This operation reduces the influence of unrepresentative data and increases the homogeneity of the sampling encompassed by an individual face track. The detection algorithm subsequently clusters the face tracks, ideally placing all of the face tracks that represent a particular individual in the same

cluster and creating a distinct cluster for every individual. If a cluster contains face tracks from more than a specified number of videos, then the detection algorithm outputs all of its images for review, as shown in Figure 1.

This paper is organized as follows. In Section 2, we review related work on unsupervised face recognition from video and video indexing. Section 3 presents the problem formulation and our proposed detection method. Experimental results obtained on a challenging crowd video dataset are provided in Section 4. Finally, a discussion of our conclusions is given in Section 5.

2. Related Work

During the last decade, an increased amount of attention has been placed on using unsupervised learning methods to aid in face recognition from video. Raytchev and Murase propose graph theoretic clustering methods [12] for recognizing faces and learning previously unobserved facial appearances simultaneously. A hierarchical clustering algorithm is used by Mian [8] to form face clusters from SIFT (Scale Invariant Feature Transform) descriptors that represent local facial features. Hadid and Pietikäinen [7] employ K-means clustering to group feature patterns obtained by locally linear embedding for the purpose of computing face exemplars. Similarly, hierarchical agglomerative clustering is applied on the face manifold, as characterized by the geodesic distances between face patterns, by Fan and Yeung [6] to achieve the same end. These exemplar based recognition methods provide efficiency gains as a nearest neighbor classifier does not need to perform as many comparisons after a dataset is reduced to a collection of representatives from a large set of face images.

These works focus on face recognition from videos recorded in one or two homogeneous environments with a single individual in view at a time. In contrast, we focus on the appearance frequency of individuals within a set of multi-person videos. We present experimental results on videos recorded in highly varied environments containing crowds of people. Moreover, the questionable observer detection problem necessitates clusterings that contain as few clusters for each person as possible. The formation of redundant clusters that contain images of the same person can impact the number of videos that each cluster spans and, hence, result in false positives, that is, false classifications of questionable observers as casual observers. The unsupervised classification techniques described in [6, 7, 8] yield clusters of face data with related appearances, *e.g*. similar expressions and poses, so that multiple clusters correspond to the same individual. The algorithm described here mitigates this problem by 1) exploiting the multitude of face observations afforded by videos through the use of face tracks, 2) merging disjoint face image sequences from the same video that contain images of the same person, and 3) performing outlier detection to reduce the impact of intermittent effects that alter facial appearance.

The questionable observer detection problem shares much in common with those encountered in video and photo album indexing. Some video indexing algorithms rely on the repeated appearance of an individual to decompose a video into semantically coherent subsequences. K-means clustering is employed by Chan *et al*. [5] to group feature patterns that correspond to an individual who appears frequently in a single news video. Similarly, Ozkan and Duygulu [9] describe a greedy algorithm that exploits facial appearance features as well as information contained within a video transcript with the objective of finding a specified person in a news video. A meeting understanding system that clusters face sequences and then counts meeting attendees is proposed by Vallespi *et al*. in [14]. This system is comprised by an adaptive subspace face tracker and a temporal subspace clustering method that extends the normalized cuts clustering algorithm. Prince and Elder [10] combine clustering with a Bayesian approach to count the number of different people that appear in a collection of face images. The parameters of a generative probabilistic model describing the face manifold are learned during training. This model enables the computation of the posterior probability over possible clusterings, so that Bayesian model selection can be applied to compare partitionings of varying sizes.

These methods use appearance frequency in a single video or over a collection of images as a way to understand the composition of a dataset, while we focus on detecting individuals that appear in multiple videos with varied backgrounds and crowds of people in every frame. The questionable observer videos do not afford a text transcript comprised by additional semantic information that can aid clustering, but news videos often do. Likewise, in contrast to the meeting videos discussed in [14], the face tracks of a particular subject from our dataset are not guaranteed to contain similar illumination conditions as we recorded both indoor and outdoor videos. Finally, the database described in [9] notwithstanding, our dataset of nearly 100 individuals contains significantly more people than the test sets mentioned in the video and photo album indexing works [5, 10, 14].

3. Proposed Method

Given a collection of m videos, $\{V_1, V_2, ..., V_m\}$, each of which shows a crowd of people observing a distinct event, we intend to identify those individuals that appear in multiple sequences. Each V_i has a corresponding set of m_i face tracks, $T_i = \{t_{i,1}, t_{i,2}, \ldots, t_{i,m_i}\}$, where every *face track* consists of a unique sequence of face images that represent the same person. The individual elements in a face track need not come from a contiguous sequence of video frames, but no two elements can come from the same video frame.

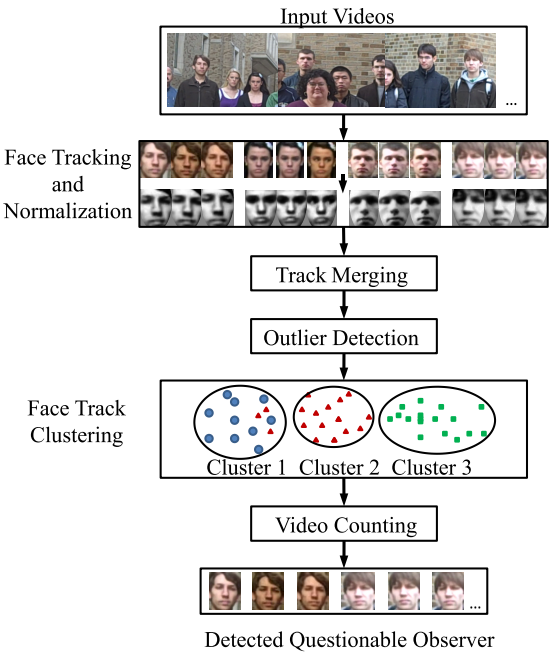

Figure 2. The proposed questionable observer detection scheme begins with face tracking and proceeds through image normalization, track merging, outlier detection, cross-video face track clustering and video counting phases.

Multiple face tracks from a single video can contain images of the same individual if she leaves the view of the camera and then reenters it later.

We frame the questionable observer detection problem as one of unsupervised classification in which we assign a class label, $l(t_{i,j})$, to each face track. The set of face tracks of the same person should be assigned the same label. A collection of face tracks that share a common label form a *face track cluster* C_L:

$$C_L = \{t_{i,j} \in \cup_{k=1}^{m} T_k : l(t_{i,j}) = L\}. \quad (1)$$

The *questionable observer detection problem* requires that we mark any cluster C_L as questionable if the number of videos from which its constituent tracks were extracted surpasses a video count threshold v. That is, an individual whose face tracks make up the majority of a cluster for which

$$|\{i \in 1, 2, \ldots, m : \exists j \; such \; that \; t_{i,j} \in C_L\}| > v \quad (2)$$

is considered to be a questionable observer.

3.1. Algorithm Overview

For each video in a collection, we extract its face tracks, normalize the constituent images, and then merge the face tracks that correspond to the same individual. An outlier detection algorithm determines a representative collection of face images within each face track to provide robustness to intermittent occlusions or changes in camera focus, facial expression, head pose, etc. Hierarchical agglomerative clustering is subsequently performed on the face tracks from all of the videos to group together the data that represents the same individual. We count the number of videos from which the face tracks in each cluster originate and output the clusters that span a number of videos beyond a specified threshold.

3.2. Face Detection and Tracking

Face tracks are formed by detecting faces in each video frame and connecting them with the faces from prior video frames that share a set of common features. We employ the face detector provided with the VeriLook 4.0 Standard SDK from Neurotechnology [1], which detects faces observed from near frontal viewpoints. Features are tracked across frames with the Kanade-Lucas-Tomasi (KLT) optical flow tracker [13]. If no features lie on a detected face, a new set of features is found within the face region and associated with a new track that only contains this face initially. Conversely, if a detected face contains features that were tracked from the prior video frame, its image is inserted at the end of the face track with which it shares the greatest number of features. A face track is marked complete when the optical flow tracker cannot find any of its corresponding features in the current video frame.

We also perform automatic eye localization as the eye positions are used to align face images to a canonical position after tracking. The VeriLook 4.0 Standard SDK includes an eye detector that is designed to locate eyes in still images. We found that the relative locations of the same pair of detected eyes can vary between consecutive video frames. These variations can lead to poor face image alignments, so that comparisons between two misaligned face tracks that represent that same person suggest that they represent different people.

We handle this issue by using singular value decomposition to form a two dimensional linear subspace that spans the positions of the features associated with a newly created face track. We compute the positions of the eye detections within the subspace and normalize their coordinates to attain partial invariance to scaling. The eye locations for a face that belongs to a preexisting face track are estimated with two operations. First, the basis vectors of the linear subspace spanning the associated feature points are computed. Second, the normalized eye coordinates from the first face track image are transformed into video frame coordinates. The normalized eye positions for a face track are recomputed when the KLT tracker cannot locate one of its corresponding features or the estimated eye locations lie

Figure 3. A collection of images from a single face track split into inlier and outlier subsets. The inliers tend to capture appearance features that remain stable throughout the face track, whereas the outliers either have poor alignment, exhibit significant in-plane rotation or contain a face making a transient facial expression.

more than a percentage of the face width and height away from the eye position estimates from the VeriLook detector. The second condition is enforced to prevent drift while the first maintains the accuracy of the normalized eye positions when the set of feature points changes.

3.3. Face Image Normalization

In general, the faces observed by the tracker rotate within the image plane and vary in scale, the extracted face images contain elements of the background, and the dynamic range of intensity values varies between video frames. We use the csuPreprocessNormalize tool from the Colorado State University Face Identification Evaluation System Version 5.0 [4] to perform geometric normalization, image masking and histogram equalization on all face images to abate these respective issues. Finally, we convert each image into a one dimensional pattern vector by unraveling the non-masked image content.

3.4. Track Merging and Outlier Detection

After extracting face tracks and normalizing face images, we iterate over the set of videos and apply hierarchical agglomerative clustering (Sec. 3.5) to merge the face tracks from a specific video that contain images of the same person. We can make clustering decisions in the context of a single video with a high degree of confidence as some of the factors that affect facial appearance, such as illumination and hair growth, typically do not vary as drastically within a video clip as across different video clips.

The merged face tracks can nevertheless span a greater range of appearance variations than the input tracks. Some observations may capture transient facial features caused by changes in facial expression, momentary head rotation, image noise and other influences, as shown in Figure 3. Such observations yield unstable feature vectors and increase the within-class appearance variability insofar as they cause a particular face to appear different when viewed at different times.

We employ an outlier detection algorithm to alleviate the impact of such problems on later clustering decisions. For a face track t with m_t face patterns, the algorithm determines the average distance from each face pattern in t to its nearest neighbors. If the average distance for a pattern lies beyond a specified outlier boundary, that pattern is discarded. Specifically, outlying face patterns are detected with the following steps:

1. For each pattern x_i in face track t, compute the average pattern distance, $d_a^{(i)}$, to its k nearest neighbors with respect to the L2 norm.

2. Compute the mean, μ_a, and the standard deviation, σ_a, of the average pattern distances.

3. Discard any pattern x_i for which $d_a^{(i)} > \mu_a + s * \sigma_a$, where s is scaling parameter that defines the boundary between outliers and inliers.

We chose $k = 0.25 * m_t$, but found that the performance of the outlier detection algorithm is not particularly sensitive to the choice of k. The scaling parameter, s, was set to -0.7 based on experimental results. We also constrained the third step to always leave at least one pattern in a track to avoid the degenerate case where a large number of face tracks are left empty.

3.5. Face Track Clustering

We perform hierarchical agglomerative clustering (HAC) to group together similar face tracks over a variety of scales. The HAC algorithm takes a similarity matrix as input and outputs a tree, called a *dendrogram*, in which every level contains a cluster formed by merging a pair of clusters from levels located closer to the leaves. The leaves of the tree represent singleton clusters consisting of individual face tracks, whereas the root represents a single cluster that contains all face tracks. The dendrogram levels correspond to similarity values, i.e. the clusters that are merged together to form clusters near the leaves are more similar than the clusters that are merged together to form clusters near the root. A particular set of clusters is selected by cutting the tree at some level.

The distance d between two face tracks, t' and t'', is given by the variance of their face patterns. That is, we aggregate the l closest face pattern pairs that span both tracks into a face pattern set $X = \{x_1, x_2, \ldots, x_{2l}\}$ and take

$$d(t', t'') = \frac{1}{2l} \sum_{i=1}^{2l} ||x_i - \mu_X||^2, \tag{3}$$

where μ_X is the mean pattern of X. We set $l = \min(|t'|, |t''|)$ to guarantee that we weight the overall contributions of the two face tracks equally when computing the mean. Moreover, we construct X from the nearest pairs of patterns to ensure that appearance variations related to identity, as opposed to those resulting from differences in

pose or facial expression, are the primary factors in computing the variance.

We apply the HAC algorithm to cluster face tracks based on the pairwise track distances. This clustering method reflects a bottom-up approach, as shown below:

1. Generate a collection of initial clusters, $\{C_1, C_2, \ldots, C_T\}$, where T is the number of face tracks and each cluster contains a single track.

2. While we have more than one cluster,

   (a) find the most similar pair of clusters, C_i and C_j, according to a metric S, and

   (b) merge C_i and C_j.

The selection of the cluster similarity metric, S, can determine the shape of clusters represented by the dendrogram. We define S in terms of the least similar pair of tracks from two clusters as indicated by the pairwise track variance, d.

$$S = \min_{t' \in C_i, t'' \in C_j} -d(t', t''). \tag{4}$$

The use of the $\min$ operator results in compact clusters with small diameters as opposed to chain-like clusters with large diameters, which tend to result from the application of the $\max$ operator. The variance of two tracks located at the opposite ends of a chain-like cluster can be significantly higher than the variance of a pair of tracks encompassed by a compact cluster. Allowing for such a large variance increases the chance that a cluster will contain face tracks that represent different individuals.

We select the dendrogram level that minimizes the ratio of the total intra-cluster variance to the inter-cluster variance, as each cluster should be homogeneous with respect to identity and different clusters should represent different individuals. Moreover, if we choose a dendrogram level with a large number of clusters, multiple clusters may contain face tracks from the same person. This can decrease questionable observer detection accuracy substantially if none of these clusters contains face tracks from more than one video in which their corresponding individual appears. We thus take into account the number of groups within a clustering when cutting the dendrogram.

Instead of measuring cluster variance in terms of face tracks, we decompose each track into its face patterns and compute the inter- and intra-cluster variances using these vectors. For a cluster $C_i = \{x_1, x_2, \ldots, x_{c_i}\}$ with c_i face patterns and cluster center μ_i, we express the intra-cluster variance as

$$Var(C_i) = \frac{1}{c_i} \sum_{i=1}^{c_i} ||x_i - \mu_i||^2 \tag{5}$$

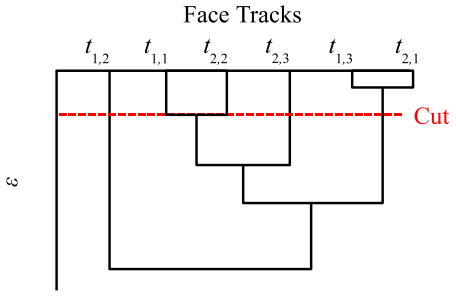

Figure 4. An example of a dendrogram constructed from the face tracks of two videos. A particular partitioning of the data is obtained by cutting the dendrogram at the level that minimizes the cost function, ϵ. If the level indicated by the red dashed line yields the lowest value of ϵ, we would cut the dendrogram at that level to attain the following clustering: $C_1 = \{t_{1,2}\}, C_2 = \{t_{1,1}, t_{2,2}\}, C_3 = \{t_{2,3}\}, C_4 = \{t_{1,3}, t_{2,1}\}$, where C_i denotes a cluster. Furthermore, clusters C_2 and C_4 would be labelled as questionable observer clusters assuming that the video count threshold was set to one video.

under the assumption that all face images are equally likely to be observed. We compute the overall intra-cluster variance, Var_i, by taking the sum of all the individual intra-cluster variances. The inter-cluster variance, Var_e, is expressed in terms of the cluster centers:

$$Var_e = \frac{1}{c} \sum_{i=1}^{c} ||\mu_i - \mu||^2 \tag{6}$$

where c denotes the number of clusters and μ represents the mean of the cluster centers. The clustering level that we choose is the one that minimizes the following cost function:

$$\epsilon = \alpha * \frac{Var_i}{Var_e} + (1 - \alpha) * c. \tag{7}$$

The α weighting parameter determines the tradeoff made between clustering accuracy and clustering redundancy. A high value of α will result in an accurate clustering with a large number of redundant clusters, whereas a low value will result in a lot of clusters that contain data from multiple people but fewer redundant clusters. For the first clustering by which the face tracks that were extracted from the same video and represent a common individual are merged, α is set to 1.0 to avoid propagating errors to the second clustering step. For the second clustering by which the face tracks from multiple videos are grouped together, we determined that $\alpha = 0.97$ provides the best results experimentally.

4. Experimental Results

To evaluate the ability of the proposed technique to detect questionable observers, we acquired a dataset comprised by fourteen 25-59 second crowd video clips recorded

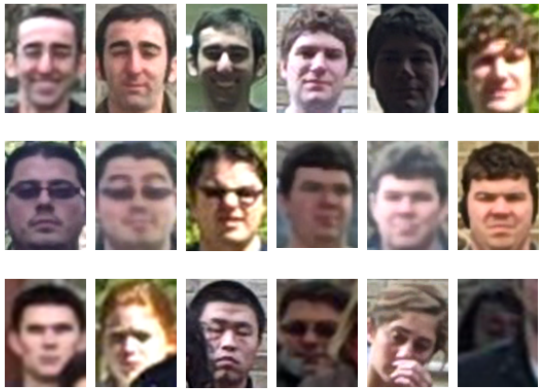

Figure 5. Complicating factors in the experimental dataset. Top row: images of two questionable observers taken under varying illumination conditions. Middle row: images of another two questionable observers making distinct facial expressions in different videos. Bottom row: instances where subjects were occluded by other crowd members or their own body parts.

in different locations around the University of Notre Dame campus over a period of seven months. This collection of videos, called the ND-QO-Flip dataset, contains 90 subjects overall, five of whom appear in multiple videos and 85 of whom appear in one video. The appearances of the questionable observers differ between videos as they wear different clothes and their facial hair lengths change, due to the seven-month time lapse. Details about how to obtain this dataset are available at http://cse.nd.edu/ cvrl/CVRL/Data_Sets.html.

In each clip, the camera pans and zooms over a crowd of four to 12 people. Most people are seen from a nearly frontal viewpoint because the observers tend to face toward the camera or focus on an object behind it. The crowd members were allowed to exhibit any facial expression they chose. The video set contains 12 outdoor videos, including six that were aquired under overcast conditions, six that were recorded when the sun was visible, three with snow cover and one with falling snow. The videos thus contain extensive variations in illumination and facial expression along with partial face occlusions caused by the way the crowds formed.

A Cisco Flip handheld camcorder was used to acquire the videos. All of the videos were compressed with the H.264 codec, have a 640x480 resolution and a frame rate of 30 frames per second. As shown in Figure 6, the Flip camcorder performs automatic white balancing and exposure control, which can cause the color range of a video to vary between successive frames. Moreover, the zoom feature is implemented in software and so some video frames can become highly blurred. Finally, the videos tend to have short subsequences in which the colors of adjacent image regions bleed together.

Figure 6. Face images from the Flip videos. Top row: blurry images recorded under full zoom. Bottom row: instances where the automatic exposure and white balancing adjustments changed facial appearance within the same face track.

4.1. Performance Metrics

In the ideal clustering, all face tracks belonging to the same individual would be assigned to the same cluster and all individuals would have a distinct cluster. The classification accuracy would be perfect in this case, as would the questionable observer detection rate. The clusterings produced in practice differ from this ideal because of two types of errors: 1) some of the face tracks belonging to one subject may be assigned to a cluster that better represents a different subject, or 2) multiple clusters may represent the same individual.

The first type of error impacts clustering accuracy by reducing the homogeneity of clusters with respect to identity. The *self-organization rate* (SOR), which was used by Raytchev and Murase [11], accounts for this error type. It is given by

$$SOR = (1 - \frac{\Sigma n_{ab} + n_e}{n}), \tag{8}$$

where n_{ab} denotes the number of patterns representing individual a that were assigned to a cluster dominated by the patterns of individual b, n_e represents the number of patterns that are assigned to a cluster in which no single individual corresponds to more than half of the patterns, and n denotes the number of patterns in the clustering.

When the images of an individual represent the majority in numerous clusters, all but one of those clusters are redundant. For a particular person a whose data points represent the majority in c_a clusters, $c_a - 1$ of those clusters are redundant. Hence, the number of redundant clusters is given by

$$c_r = \sum_a c_a - 1. \tag{9}$$

In turn, we define the *cluster conciseness*, CON, as

$$CON = (1 - \frac{c_r}{c}), \tag{10}$$

where c is the number of clusters. CON is inversely proportional to the number of redundant clusters so that higher CON values indicate a higher quality clustering, as is the case for the SOR.

The *false positive rate* (FPR) and *false negative rate* (FNR) are used to convey how well the detection algorithm distinguished questionable observers from casual observers. A *false positive* occurs when any cluster that represents a casual observer has face tracks from more than v videos. A *false negative* occurs when none of the clusters that correspond to a questionable observer contain face tracks from more than v videos. Both the false positive and false negative rates vary from zero to one. A lower value indicates better detection performance in each case.

4.2. Results

We implemented four additional detection methods for the purposes of comparison. Three of these provide the means to determine the impacts of track merging and outlier detection. The first performs HAC without track merging or outlier detection, the second combines HAC with track merging, while the third employs HAC and outlier detection. We refer to these techniques as HAC, HAC/track merging and HAC/outlier detection, respectively.

The fourth technique, called HAC/VeriLook, incorporates components from the state-of-the-art VeriLook face recognition software by Neurotechnology [1]. The VeriLook 4.0 Standard SDK provides a face template generation algorithm that can create a generalized template characterizing a wide range of appearance variations for a particular person. We performed k-means clustering on every face track after the track merging step to form clusters representing different appearance modes. We subsequently selected the highest quality face image from each cluster, as indicated by the VeriLook face detector, and inserted them into the generalized template for the associated face track. The generalized templates from all of the videos were compared using the VeriLook matcher to create a similarity matrix. The hierarchical agglomerative clustering algorithm discussed in Section 3.5 was then used to group the face tracks based on the VeriLook similarity scores. The clustering is selected from the dendrogram in terms of the questionable observer detection FPR and FNR, not equation 7, so that the HAC/VeriLook algorithm has the advantage of using the best grouping possible with respect to the detection performance metrics.

As the results in Table 1 demonstrate, the track merging and outlier detection steps improved questionable observer detection performance, which allowed the proposed algorithm to outperform the HAC/VeriLook algorithm across all metrics. Track merging increased clustering accuracy and decreased the percentage of redundant tracks, as shown by the superior performance of the HAC/track merging method relative to the HAC technique and that of the proposed algorithm relative to the HAC/outlier detection method.

Table 1. Questionable observer detection results for a video count threshold of one video with the best result for each performance metric highlighted in green.

Method	SOR	CON	FPR	FNR
HAC	0.966	0.603	0.018	0.60
HAC/track merging	0.972	0.629	0.018	0.60
HAC/outlier detection	0.912	0.642	0.064	0.00
HAC/VeriLook	0.895	0.317	0.061	0.40
Proposed algorithm	0.960	0.664	0.056	0.00

Both the proposed algorithm and the HAC/outlier detection technique surpass the HAC method in terms of the CON and FNR yet produce less accurate clusterings with more false detections. One reason for this discrepancy is that the outlier detection step removes images from all face tracks. The misclassified face tracks that contain a relatively large number of images incur a greater penalty after a significant number of face images are discarded overall. The HAC/outlier detection method and the proposed algorithm achieved a lower FNR because they yielded fewer redundant clusters, but they obtained a higher FPR since they formed more clusters that span multiple identities. These performance trends exemplify the tradeoff between clustering accuracy and redundancy as well as the tradeoff between the FPR and the FNR.

In the context of the questionable observer detection problem, the penalty for a false negative is generally higher than that for a false positive. The detection algorithm outputs a relatively small number of questionable observer clusters for review. In the case of a false positive, a user can analyze such a cluster to verify that it actually contains a set of face tracks that were extracted from a specified number of videos and represent the same person. In the case of a false negative, all of the clusters that represent a questionable observer are assigned to a large collection of casual observer clusters. The time cost associated with reviewing this collection is much higher than that associated with validating the questionable observer clusters. These relative costs indicate that the achievement of a low FNR is of the utmost importance, followed by the attainments of a low FPR and a high SOR. Hence, the proposed algorithm achieved the best performance balance for the questionable observer detection problem.

5. Conclusions

The face classification problem presented in this paper is difficult for a number of reasons. Unlike the face exemplar selection task to which face clustering methods are often applied [6, 7, 8], the questionable observer detection

problem does not allow for the aggregation of a large number of redundant clusters as this could result in an excessive number of false negatives. Here, we focus on analyzing the appearance frequency of individuals across collections of crowd videos acquired under a wide variety of illumination conditions, which, to the best of our knowledge, is an unresearched application of face clustering.

The instance of the questionable observer detection problem considered here is challenging due to the large number of nuisance variables within the experimental dataset, including facial expression, illumination, occlusion, and video quality, amongst others. We found that a combination of track merging, outlier detection and hierarchical agglomerative clustering can successfully detect the questionable observers while yielding a low frequency of false detections despite these difficulties. Moreover, these techniques outperformed a system constructed from the VeriLook 4.0 Standard SDK [1] in terms of the clustering conciseness as well as the self-organization, false positive and false negative rates.

6. Acknowledgements

This research is supported by the Central Intelligence Agency, the Biometrics Task Force and the Technical Support Working Group through US Army contract W91CRB-08-C-0093. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of our sponsors.

References

[1] VeriLook Standard SDK and Extended SDK. http://www.neurotechnology.com/vl_sdk.html, 2010.

[2] O. Arandjelović and R. Cipolla. An information-theoretic approach to face recognition from face motion manifolds. *Image and Vision Computing*, 24(6):639–647, 2006.

[3] J. Bavelas and N. Chovil. Faces in dialogue. *The psychology of facial expression.*, pages 334–346, 1997.

[4] D. S. Bolme, J. R. Beveridge, M. Teixeira, and B. A. Draper. The CSU face identification evaluation system: Its purpose, features and structure. *Proc. International Conference on Vision Systems*, pages 304–311, 2003.

[5] Y. Chan, S. Lin, Y. Tan, and S. Kung. Video shot classification using human faces. *Proc. IEEE International Conference on Image Processing*, 3:843–6, 1996.

[6] W. Fan and D. Yeung. Face recognition with image sets using hierarchically extracted exemplars from appearance manifolds. *Proc. 7th International Conference on Automatic Face and Gesture Recognition*, pages 177–82, 2006.

[7] A. Hadid and M. Pietikäinen. Selecting models from videos for appearance-based face recognition. *Proc. 17th International Conference on Pattern Recognition*, 1:304–8, 2004.

[8] A. Mian. Unsupervised learning from local features for video-based face recognition. *Proc. 8th IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[9] D. Ozkan and P. Duygulu. Finding people frequently appearing in news. *Proc. 5th International Conference on Image and Video Retrieval*, pages 173–82, 2006.

[10] S. Prince and J. Elder. Bayesian identity clustering. *Proc. 2010 Canadian Conference on Computer and Robot Vision*, pages 32 –39, 2010.

[11] B. Raytchev and H. Murase. Unsupervised face recognition from image sequences based on clustering with attraction and repulsion. *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:II–25–30, 2001.

[12] B. Raytchev and H. Murase. Unsupervised face recognition by associative chaining. *Pattern Recognition*, 36:245–57, 2003.

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

[14] C. Vallespi, F. De la Torre, M. Veloso, and T. Kanade. Automatic clustering of faces in meetings. *Proc. IEEE International Conference on Image Processing*, pages 1841–1844, Oct. 2006.