Unsupervised learning technique for audience privacy protection in video lectures

Juthika Dabholkar
juthika@stanford.edu

Xunjia Lu
rluxj@stanford.edu

Harsh Nayyar
hnayyar@stanford.edu

Sijia Zheng
sijiazh@stanford.edu

***Abstract*—This work presents a novel technique to perform audience privacy protection in video lectures. The main contribution of this work is a heuristic based iterative clustering procedure that isolates the lecturer from audience members. This iterative process provides the labelling required to identify and blur audience members.**

I. Introduction

In this work we present a solution to the problem of protecting audience privacy in video lectures. This technique consists of first performing robust face detection and tracking, and using this as input to an iterative clustering process that is optimized to accurately isolate the lecturer from audience members.

Section II provides a detailed description of the problem. Section III provides a summary of related work. Section IV consists of an overview of the face detection, tracking, and clustering algorithms we employ in this work. Section V outlines our proposed design, while Section VI presents our initial results. We evaluate these results in Section VII and conclude in Section VIII with a discussion on how to improve our proposed design.

II. Problem Description

This work is motivated by the Class-X system at Stanford University. Class-X is an online archive of video lectures of Stanford Electrical Engineering courses. In order to make this valuable video archive available to the public without restriction, it is necessary to protect the identity of any students who may appear in the videos.

Formally, this requires that all students appearing in a given video are identified and blurred. We assume no prior information on the identity of the lecturer. Hence, the problem requires that the lecturer be identified, isolated from student appearances, and not be erroneously blurred.

It is also important to note that the video lecture may be captured with either a still or moving camera. As a result, the *ideal* solution should be invariant to how the input video is captured.

III. Relevant Literature

A survey of the literature reveals some relevant work in the area of privacy protection in video surveillance. Much of this work is motivated by the proliferation of video surveillance systems, and the resulting need to protect an individual's privacy.

While offering solutions for scenarios under which all identified faces must be obscured, the literature does not offer techniques that can *discriminate* between faces and refrain from obscuring a particular target (e.g., the lecturer).

In [1] Wang, Suwandy, and Yau describe present a technique that uses a modified Adaboost face detector and kernel-based mean shift combined with active contour to track faces. This approach is able adapt to changes in the scale of faces. Subsequently, each detected face is blurred using a 5x5 median filter.

In [2], Senior offers a set of five design principles for the design of privacy protection systems. The most important and practical principle for our scenario is the author's suggestion that such systems bias towards false-positives for optimal privacy protection. The logic behind this principle is that a single detection failure can compromise the identity of an individual and thereby render the privacy protection scheme useless.

IV. Background

A. Face Detection

Many algorithms cast face detection as a binary classify problem. One technique is to is detecting faces by color. A common ML algorithm used in this method is Principal Components Analysis (PCA) [3]. The disadvantage of this technique is that it is not very robust under varying lighting conditions and that it may not work for all skin colors.

Detecting faces by motion is commonly used in real-time videos. Since faces are usually moving, calculating the moving area by background subtracting will get the face segment. With the interference of other moving objects, a face can be detected by detecting a blinking pattern in the moving segments [4].

Viola & Jones' weak classifier cascade is a breakthrough in face detection [5]. Instead of using pixel values as features, they use a new image representation called integral image that allows for faster and more robust feature evaluation. To improve performance, it selects a small number of important features by using the AdaBoost procedure. Finally, it uses a cascade of successively more complex classifiers to study on promising regions of the images, which yields significant improvement in the speed of face detection. This technique is now the most commonly used algorithm for face detection; it is also implemented in OpenCV.

B. Tracking

Background subtraction and color-based filtering are two simple approaches that may be used in face tracking. Another approach is model-based face tracking, which uses a model describing the appearance, shape and motion of faces to aid in estimation. Upon face detection, a model is laid over the face so that the system can perform tracking.

Mean shift, which shifts each data point to the average of neighboring data points, is also a commonly used technique in face tracking [6]. Ensemble tracking is a face tracking algorithm based on mean shift. It uses an ensemble of weak classifiers to create a confidence map in the new frame according to the faces in the previous frame, and uses mean shift to find the peak of a confidence map near the faces' old positions [7].

C. Clustering

Clustering is a popular unsupervised machine learning technique. In this technique, the input is an unlabeled training set and the objective is to produce a given number of cohesive clusters.

One simple and popular clustering algorithm is the k-means algorithm. This algorithm is initialized (using some heuristic) to k means (or centroids). The algorithm then assigns all input vectors to the closest centroid and proceeds to recalculate the means. After sufficient iterations, the centroids converge.

V. Proposed Design

A. Design Overview

As described above, our proposed system processes an unprotected video stream in order to identify and obscure all audience members. The high level system overview is depicted in Figure 1.

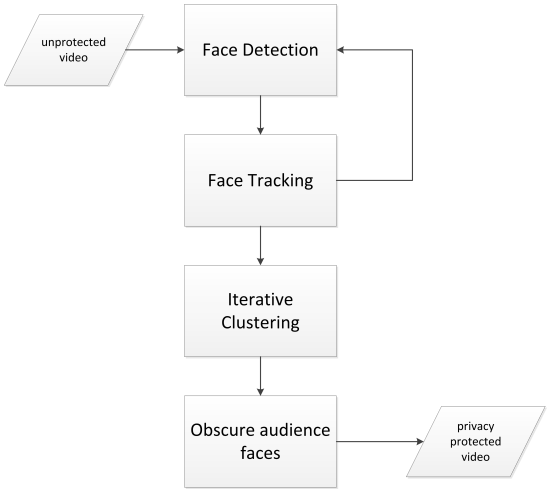

Fig. 1. High level system overview.

Given an unprotected input video, we iteratively perform face detection followed by tracking, in a single pass through the video. This set of detected and tracked faces is the input to our iterative clustering procedure. After this procedure, we obscure all identified audience faces to produce the privacy protected output video.

B. Implementation Details

We now present the detailed implementation with respect to each stage of our technique as depicted in Figure 2.

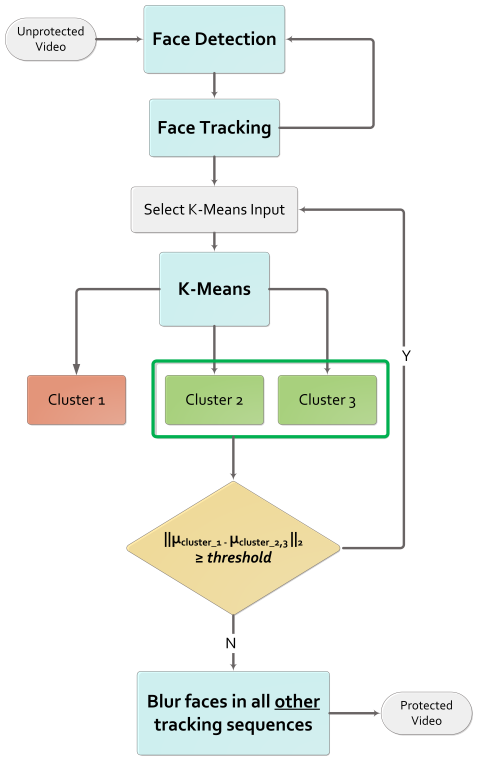

Fig. 2. Detailed system design.

1) Face Detection and Tracking: We perform face detection using the OpenCV implementation of Viola and Jones' technique. Meanwhile, tracking is performed using the mean shift approach based on skin tone.

Due to the fact that most of the false positives come from background, it makes sense to identify some area of interest for every frame and do a face detection only on that area. Suppose we have a frame set as background, our approach was to locate a bounding box on current frame that identifies the region with the greatest change from the background. The difference between two frames is given by the absolute value of differences in the pixel values of the frames, parameterized by threshold τ.

While these are two seperate modules, they are are intricately connected. We perform face detection at a parameterized interval (subsequent to background subtraction as a preprocessing step). In order to ensure that all faces are detected, we then track all detected faces, both forward and backwards. We track backwards only in case the number of faces detected on a particular instance of the face detector increases. The faces detected from the video are grouped by the tracks that they belong to.

2) Iterative Clustering: In [8], Huang, Wang, and Shao present a promising iterative clustering scheme to seperate different individuals into clusters. We adapt this scheme to our scenario by postulating that the largest clusters will be the lecturer. Based on this heuristic, we are able to discriminate between audience faces and the lecturer's face. In this stage, we consider each detected face individually for the purposes of determining our clusters.

We assume that the professor's face appears most often in the video. So most faces detected (i.e. images fed to iterative clustering) are the professor's face. Using this heuristic, we assume that when running k-means, the professor's faces will always be in the larger clusters while students' faces and other noises will be in the smaller clusters. Given the students' faces are in the smaller clusters, we eliminate the students' faces by iteratively running k-means and excluding the smallest cluster.

The stopping criterion is a threshold parameter. In our algorithm, this is called *diff*. We stop when the difference in the mean between the two large clusters and the small cluster is below a threshold parameter. Figure 3 outlines this algorithm in detail.

Input faces: M = $\{M_i^j\}$ i = track ID, j = frame ID
$\{M_1^1, M_1^2, ..., M_1^{n_1}\}$, $\{M_2^1, M_2^2, ..., M_2^{n_2}\}$, ..., $\{M_t^1, M_t^2, ..., M_t^{n_t}\}$
$S_{init} \leftarrow size(M)$
DIFF$\leftarrow$ inf
while DIFF $>$ threshold **do**
  Initialize cluster centroids μ_1, μ_2, μ_3 randomly

  Repeat until convergence { //standard k-means
  for $i = 1 : t$ **do**
    for $j = 1 : n_i$ **do**
      set $C_{ij} = argmin_k ||M_i^{(j)} - \mu_k||^2$
    end for
  end for
  for $k = 1 : 3$ **do**
    set $\mu_k = \frac{\sum_{i=1}^{\tau}\sum_{j=1}^{n_i} 1\{C_{ij}=k\}M_i^{(j)}}{\sum_{i=1}^{\tau}\sum_{j=1}^{n_i} 1\{C_{ij}=k\}}$
  end for
  }

  //find sizes of 3 clusters
  for $k = 1 : 3$ **do**
    $S_k := \sum_{i=1}^{\tau}\sum_{j=1}^{n_i} 1\{C_{ij} = k\}$
  end for

  min_cluster $:= argmin_k S_k$ //find smallest cluster

  for $i = 1 : t$ **do**
    for $j = 1 : n_i$ **do**
      if $C_{ij} =$ min_cluster **then**
        remove $M_i^{(j)}$ from $\{M\}$
      end if
    end for
  end for

  $\mu_{small} = \mu_{min\_cluster}$ //mean of smallest cluster
  $\mu_{big} = \frac{\sum_{k \neq min\_cluster} \mu_k S_k}{\sum_{k \neq min\_cluster} S_k}$ //mean of other clusters
  DIFF $= \text{sqrt}(\text{norm}(\mu_{small} - \mu_{big}))/S_{init}$

Fig. 3. Iterative clustering algorithm pseudocode.

3) Obscuring Audience Faces: The output of the previous stage is a binary tag corresponding to whether a detected face corresponds to audience or lecturer. Based on this binary classification, we can apply any desired technique to obscure the audience faces.

For illustrative purposes we will apply a simple block color replacement. In practice, this stage may be adapted based on the level of privacy protection that is desired.

VI. Results

In the process of developing and evaluating our technique, we used two sample video sequences from the Class-X system. We refer to them as them fb2 and fb3.

As a first step towards implementing our proposed design, we simplified the problem to only operate on a still camera video source. This allows us to to localize the region with people by performing background subtraction.

As a preliminary step, we tried basic face detection algorithms with different thresholds against the sample video. We used the OpenCV library to read the input video frame by frame. We then perform frontal face detection on each individual frame using the OpenCV implemention of the Viola-Jones' technique. Each frame is then an output video with rectangles indicaiting faces recognized. Figure 4 below is a representative frame output:

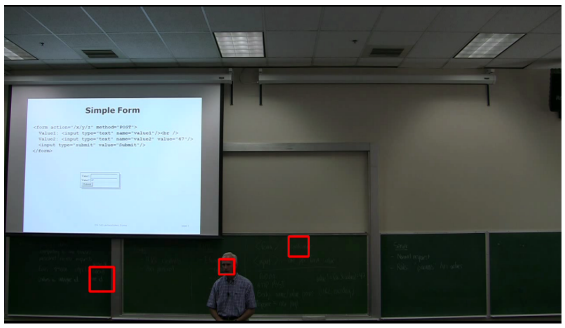

Fig. 4. Initial face detection output is noisy.

The face detector is able to recognize faces of different orientations up to a cetain angle. As we can conclude from the figure above, the output contains considerable noise. This is unacceptable. Figure 5 demonstrates the limited ability of the face detector to detect side faces.

Fig. 5. Frontal face detection is accurate while side face detection at extreme poses fails.

We tried two approaches for background substraction. The first approach is to use the very first frame as the background reference. The second approach is to update background frame every n frames. The second approach yields much better results. We also tried different value of n. If n is too large, we skip a large number of frames. If n is too small, we are actually comparing frames that are really similar to one another. Figure 6 is an ideal output.

Fig. 6. Output after background subtraction to determine bounding box.

After performing face detection as described above using background subtraction, we perform mean shift tracking both forward and backwards over each interval. Figure 7 is a representative tracking sequence. The side faces present in this output (in contrast to Figure 5) are a result of tracking. Without tracking, such faces cannot be captured due to the face orientation. This becomes the input to our iterative clustering process (as described above).

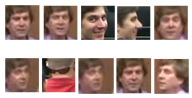

Fig. 7. Output of tracker.

Figure 8 is representative of our clustering output.

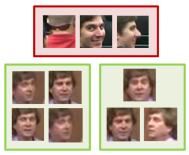

Fig. 8. Output of iterative clustering, red = small cluster, green = big cluster.

Finally, we show the protected output for a random frame in both the fb2 and fb3 sequences:

Fig. 9. Protected output for frame in fb2 sequence.

VII. Evaluation

A. Methodology

In order to evaluate our results, we focus on optimizing the two key parameters in our technique. The first of these is the frame interval between successive background subtractions and face detections. The second parameter we optimize is the stopping threshold, *DIFF*, that we use to terminate the iterative clustering procedure.

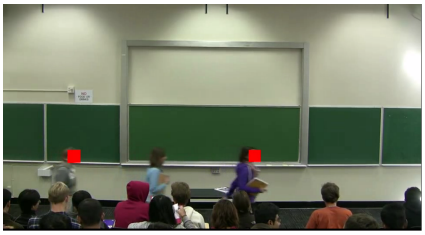

Fig. 10. Protected output for frame in fb3 sequence.

We evaluate the performance with respect to precision (P) and recall (R) statistics. We measure accuracy using the F-score. We define these measures with respect to true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN):

$$P = \frac{TP}{TP + FP}$$
$$R = \frac{TP}{TP + FN}$$
$$Fscore = \frac{2PR}{P + R}$$

It is worth noting that in this context, the recall (R) measure is more appropriate. This is because the system performance is ultimately dependant upon the degree of privacy protection the technique achieves. This measure directly calculates the fraction of audience faces blurred.

B. Analysis

Due to time constraints, we perform this detailed analysis on the fb3 sequence. In order to perform this analysis, we first determined the ground truth for this input sequence to the clustering process, and compare it with the output set of the clustering stage.

We perform this analysis for frame intervals 10, 20, 30, 40. For each interval, we analyze the results for the following clustering stopping thresholds: 31, 37, 43, 48. We summarize the results in Figure 11, 12 and 13.

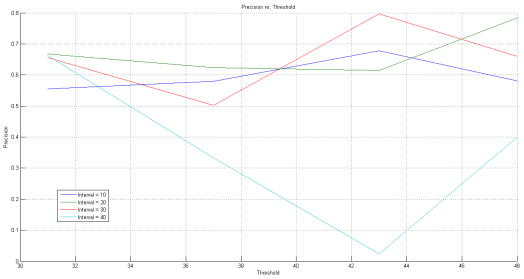

Fig. 11. Precision vs. threshold for varying intervals

To summarize, our best precision is with an interval of 30 and a threshold of 43. Our best recall result is with a interval

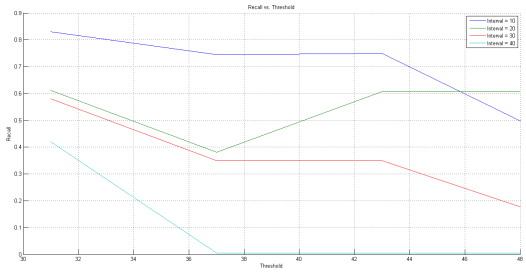

Fig. 12. Recall vs. threshold for varying intervals

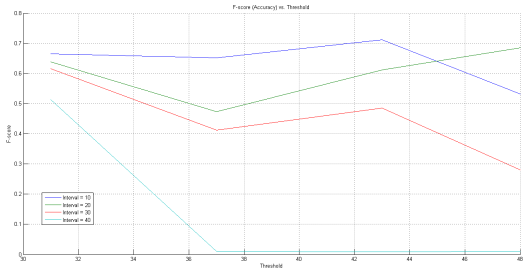

Fig. 13. F-score (Accuracy) vs. threshold for varying intervals

of 10 and threshold of 31. Finally our best F-score is with an interval of 10 and a threshold of 43.

Moreover, the general trend in all three measures is that as the interval size increases, the performance metric decreases. This suggests that the increased window for tracking is resulting in significant noise.

VIII. Conclusion

Based on the analysis above, we can conclude that the performance of our technique is limited by the quality of the detected set of faces. When there is minimal noise (i.e., non-faces like blackboard), we can get good recall results.

In the future, we would revisit the design of the face detection and tracking stages. We might also investigate using features such as the skin tone for our clustering process.

References

[1] Jian-Gang Wang; Suwandy, A.; Wei-Yun Yau; , "Face obscuration in a video sequence by integrating kernel-based mean-shift and active contour," Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on , vol., no., pp.2314-2318, 17-20 Dec. 2008

[2] Senior, A.; , "Privacy enablement in a surveillance system," Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on , vol., no., pp.1680-1683, 12-15 Oct. 2008

[3] Menser, B.; Muller, F.; , "Face detection in color images using principal components analysis ," Image Processing and Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465) , vol.2, no., pp.620-624 vol.2, 1999

[4] L. Sun, G. Pan, and Z. Wu, "Blinking-based live face detection using conditional random fields," International Conference on Biometrics, Aug. 2007, Lecture Notes in Computer Science, vol. 4261, 2007, pp.252-260.

[5] Viola, P.; Jones, M.; , "Rapid object detection using a boosted cascade of simple features," Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on , vol.1, no., pp. I-511- I-518 vol.1, 2001

[6] Yizong Cheng; , "Mean shift, mode seeking, and clustering," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.17, no.8, pp.790-799, Aug 1995

[7] Avidan, S.; , "Ensemble Tracking," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.29, no.2, pp.261-271, Feb. 2007

[8] Panpan Huang; Yunhong Wang; Ming Shao; , "A New Method for Multi-view Face Clustering in Video Sequence," Data Mining Workshops, 2008. ICDMW '08. IEEE International Conference on , vol., no., pp.869-873, 15-19 Dec. 2008