Clustering and Super-Resolution of Facial Images in a Forensic Scenario

João Satiro¹, Paulo L. Correia¹, Kamal Nasrollahi² and Thomas B. Moeslund²

¹ Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Portugal
e-mail: joao.satiro@tecnico.ulisboa.pt, plc@lx.it.pt
² Visual Analysis of People (VAP) laboratory, Aalborg University, Denmark
e-mail: {kn,tbm}@create.aau.dk

Abstract—One of the main objectives when analysing surveillance videos is to recognize the suspects. However, nowadays, most of the biometric data gathering from surveillance videos is done manually, this can be a very time-consuming process which is propitious to errors. Therefore, in this thesis a tool to automatically enhance the visualization of facial images from surveillance videos is presented. This work is a complement to the BioFV’s project (surveillance video analysis software), and it consists on using BioFV’s face detector to extract face images from videos of crime scenes, eliminating the outlier images, clustering faces images of different people into different groups, and applying a multiple-image super-resolution technique on faces of the same cluster. This is done to improve the quality of the facial images, with the goal of easing facial recognition of the suspect. Every step of the system is explained in detail, and supported on bibliographic research. The results indicate that the proposed work improves facial recognition scores and suggest that it could be used by forensic investigators.

Keywords—Forensics, Biometrics, Clustering, Facial Images, Super-Resolution.

I. INTRODUCTION

The face, as one of the most common biometrics, is of great importance in many real-world applications, like human-computer interaction, human identification, access control, border control, to name a few. Facial images are of critical importance in forensic scenarios as well. The main difference when considering forensic scenarios is that facial images are taken by surveillance cameras, if there is any in the scene, and subjects of interest are not cooperative with the system nor are the imaging conditions as controlled. This makes it very challenging to work with facial images in forensic scenarios.

The problems with facial images in forensic scenarios are: 1) they are usually very small and 2) they are of poor quality. The former one is a result of the distance between surveillance cameras and subjects of interest while the latter can be a result of, among others, not facing the camera, facial expression, bad illumination, and blur [14]. These problems make it very hard to use such facial images in automatic systems, like face recognition. One solution for dealing with these problems (mainly the small sizes of the images) is to use upsampling methods, like interpolation methods. The problem, however, with simple interpolation approaches is that they can’t produce artifact-free higher resolution images from lower resolution input images. One way to deal with this problem is by using information from several LR face images, in order to produce higher resolution images, using super-resolution (SR) algorithms.

A. Face Clustering

Given a set of face images, eventually including false positives, detected in a video sequence using BioFov’s face detector, the first goal is to find a process that groups faces from the same person into the same cluster. Faces of the same person that appear in different poses should be grouped into different clusters. This is done because when the goal is to perform accurate Super-Resolution, the faces that belong to the same cluster can’t have very different poses.

First, the outliers must be removed. This can be done by using face detectors on the candidate face images. Some non-face filtering techniques include skin tone classification [7]. It is based on the assumption that human skin contains a characteristic colour type which may be distinguished from other objects. Explicit colour thresholding [8], [3], [1] is an easy to use technique, which consists on defining boundaries on the different colour space components. Other way to classify pixels as skin-toned, is by using machine learning techniques, such as Gaussian models, Neural Networks, Self Organizing Maps, Maximum entropy or Bayesian networks [6]. One widely employed face detection technique, which may also be used as a pre-processing method, is based on Principal Component Analysis (PCA) [7], [20]. This method is independent from colour, which is important in our context, since not all surveillance videos have colour information available.

After the outliers removal, the goal is to create a dissimilarity matrix which compares each face image to all the others. To compare two images, one option is to extract a set of keypoint features to describe relevant areas of the image before comparing them. Some feature extraction methods include Gabor Features [7], SIFT [10] (and its variants), or Local Binary Patterns.

Having the dissimilarity between every image, it is finally possible to perform the clustering. According to [2], clustering algorithms may be classified as 1) Partitioning relocation methods, which partition the data in several subsets using heuristic algorithms (like k-means and k-medoids); 2) Hierarchical methods, which gradually merge clusters increasing their size (agglomerative), or separate clusters into various smaller clusters (divisive), creating a "hierarchy" of clusters; 3)
to have a good registration algorithm to be able to find the improvement factors [13]. Furthermore, most of them need based SR algorithms is that they cannot provide very large produce a HR image. The problem with the reconstruction-step, where all the inputs are registered to a common frame. The registered images are then fused together to produce a HR image. The problem with the reconstruction-based SR algorithms is that they cannot provide very large improvement factors [13]. Furthermore, most of them need to have a good registration algorithm to be able to find the

In Hierarchical Agglomerative Clustering (HAC), clusters are combined based on some similarity measure. Three commonly used similarities are single-link (SL), complete-link (CL) and average-link (AL) similarities, described in the following equations [7]:

\[
SL(i, j) = \max \{ \text{sim}(a, b), \forall a \in C_i, \forall b \in C_j \} 
\]

\[
CL(i, j) = \min \{ \text{sim}(a, b), \forall a \in C_i, \forall b \in C_j \} 
\]

\[
AL(i, j) = \frac{1}{|C_i||C_j|} \sum_{a \in C_i} \sum_{b \in C_j} \text{sim}(a, b)
\]

where, \( \text{sim}(a, b) \) is the similarity between objects \( a \) and \( b \). \( C_x \) denotes cluster \( x \) and \( |C_x| \) refers to the size of cluster \( C_x \).

**B. Super-Resolution**

SR algorithms are generally divided into two groups: 1) reconstruction-based, applied when multiple input images of the same person are available; and 2) hallucination-based, usually applied when there is a single input image [13]. In the hallucination-based approaches there is usually a training step in which the relationship between the low-resolution (LR) images (or their patches) and their high-resolution (HR) counterparts are learned. In the testing step, this learned relationship is then used to predict or hallucinate missing HR details of an input LR image. However, they are not suitable for forensics applications for a critical reason: the learning algorithm behind these systems actually teach them to hallucinate missing HR details. This might not be that problematic with general computer vision applications, but for forensics scenarios where legal issues are of top priority for law enforcement, such hallucination techniques will not be acceptable to the court of law. The proposed solution is therefore based on using reconstruction-based SR algorithms.

The reconstruction-based SR algorithms, also known as multi-frame algorithms [13], usually take a set of LR images of the same scene (here human face) and try to utilize the differences between them to reconstruct missing HR details by reversing the steps involved in the imaging model. The differences between LR images can be of different forms, e.g., like sub-pixel misalignment, depth, etc.

These differences are usually compensated for in a registration step, where all the inputs are registered to a common frame. The registered images are then fused together to produce a HR image. The problem with the reconstruction-based SR algorithms is that they cannot provide very large improvement factors [13]. Furthermore, most of them need to have a good registration algorithm to be able to find the motion (or generally the differences) between different LR images. This is very challenging in forensics scenarios as the LR images are really of poor quality and explicit estimation of the motion is very error prone. Therefore, in this paper, we have used a modified version of [15] which does not need an explicit motion estimation. We have shown that the proposed system can produce good quality results for typical images taken from surveillance cameras.

The rest of this paper is organized as follows: The proposed system is detailed in section II. Then, the experimental results are given in III. Finally, the paper is concluded in section VI.

**II. THE PROPOSED SYSTEM**

The block diagram of the proposed system is shown in Fig. 1. Following this diagram, faces are first detected from the input video sequence and then clustered based on similarity measures. The clustered faces are then fed into a reconstruction-based SR algorithm. This algorithm combines information from different LR facial images into a HR one, which is of better quality than the interpolated LR input. Details of these steps are provided in the following subsections.

**A. Face detection and clustering**

For the present implementation the frontal face detection module of BioFoV\(^1\) has been used which employs Haar like features of [9]. This detector usually contains some outlier and background in the detected faces. To remove these, before clustering faces into groups a pre-processing step using skin-tone filtering is considered.

Outlier filtering is two-fold: skin tone based and PCA based filtering. The latter holds to the assumption that human skin has a characteristic colour which may be distinguished from many other objects, although some will share the same colours, like some types of wood. In addition, body parts other than faces will also pass the skin-tone filtering test, meaning that this filter cannot ascertain that a given image contains a face, but it is still able to exclude many of the outliers initially detected as faces. Explicit colour thresholding has been employed in the YCbCr colour space, following the analysis in [7], which has shown that when working with the HSV or YCbCr spaces, skin colour is concentrated in a small region of the space defined by the colour components. The selection of thresholds to be used was decided based on a set of tests performed on a database containing face and non-face images. The face images used for testing are from the Labeled Faces in the Wild database [5], which contains faces from people of different ethnicity acquired in unconstrained scenarios. Non-face images were collected from various Internet sites and from an MPEG-7 test set containing indoor and outdoor images of varying scenes. From these tests, a pixel is classified as skin-coloured if its YCbCr values fall under the following criteria: \(-48 \leq Cb \leq -8, 5 \leq Cr \leq 45\) [1]. From the performed tests we also concluded

\[\text{https://github.com/BioFoV/BioFoV}\]

---

\(^1\)https://github.com/BioFoV/BioFoV
that 80% of outlier images have less than 40% of its pixels classified as skin-coloured, while 95% of face images have more than 40% of skin-coloured pixels.

The PCA filter consists on using a representative database of faces in order to compute a subspace that contains most of the faces information. The concept behind this method is that the distance between a face image and its projection onto this “face-space” would be smaller than the image of a non-face [20]. If the distance ε between an image and its projection onto the face-space is bigger than a threshold τ, then it is considered to not belong to a face. Because input images are acquired in unconstrained environments, if the training set isn’t representative enough, the distance to their projection in the face space may be too large. For this reason, the training set must be carefully chosen (we used the Yale faces database), as well as τ. According to the conducted experiments, we chose τ = 4500.

For discriminating faces of different users, or of the same user in different poses, SIFT features are used, because of their robustness to different scales, illuminations, orientations and affine transformations [10].

When computing the similarity between face images, we use the number of matching SIFT features. This global similarity measure is described in equation 4, where \( M_{ij} \) is the maximum number of keypoint matches found between the pairs of images \( i,j \), and \( K_i \) and \( K_j \) are the number of keypoints found in images \( i \) and \( j \), respectively.

\[
S_{ij}^G = \frac{M_{ij}}{min(\ K_i, \ K_j)} \tag{4}
\]

However, when matching face images, we want to avoid that keypoints located in different areas of the face are matched. For instance, a feature close to an eye should not be matched to another close to the mouth. Therefore, the spatial distribution of keypoint matches is taken into account using the local similarity measure proposed in [12], and presented in Eq. 5.

\[
S_{ij}^L = \frac{1}{k} \sum_{n=1}^{k} \left( \max(s(f_{1n}, f_{2n})) \right) \times w_n, \quad \forall x,y \tag{5}
\]

Where \( k \) is the number of face subregions, \( s(f_1, f_2) \) denotes the cosine similarity between the feature descriptors \( f_1 \) and \( f_2 \), \( f_{1n}^x \) is the feature \( x \) of image \( i \) in subregion \( n \), \( w_n \) is the importance associated with the region \( n \) (the sum of all weights must be unitary), and \( x \) and \( y \) represent all the features in image \( i \) and \( j \), respectively. The total similarity, is simply the multiplication of \( S_{ij}^G \) with \( S_{ij}^L \).

For clustering, HAC [7] is used in the present implementation, with a complete linkage metric and a stopping threshold \( \gamma = 0.7 \), as it achieved the best results in the conducted clustering experiments. Also, in order to help the clustering process, contextual (spatial) information was used, namely if two face images appeared on the same video frame, they couldn’t be joined in the same cluster.

B. Super-resolution

As mentioned before, for forensics scenarios multi-frame SR techniques should be used. When trying to register images belonging to a given face cluster one may find complex motions, as the face is not a rigid body, and self-occlusions, with some face elements, such as the nose, occluding others. Therefore, the choice was to use a SR algorithm that does not need an explicit motion estimation [15]. This method involves: (i) choosing the reference image from the available LR images; (ii) register each image onto the reference; and (iii) fuse them. As mentioned before, for forensics scenarios multi-frame SR techniques should be used. When trying to register images belonging to a given face cluster one may find complex motions, as the face is not a rigid body, and self-occlusions, with some face elements, such as the nose, occluding others. Therefore, the choice was to use a SR algorithm that does not need an explicit motion estimation [15]. This method involves: (i) choosing the reference image from the available LR images; (ii) register each image onto the reference; and (iii) fuse them into the HR space, using sub-pixel resolution. A deblurring filter is then applied.

1) Reference frame selection: In a forensics scenario, all LR images are typically of low quality. However, if one of them is of better resolution, illumination and/or sharpness, it should be selected as the reference. In the present implementation the image with the best illumination is chosen as reference.

2) Registration: The registration step finds the correspondences between pixels in the reference and all other LR images. Registration accuracy is crucial for the reconstruction success, determining pixel values and positions in the HR grid. Before registration all LR images are interpolated to the desired HR size using a bicubic interpolation.

Due to the complex nature of face images, a parametric registration using translation, rotation, scale or affine transformations (i.e., an explicit motion estimation) may not work well. Instead, an optical flow technique is used, finding the motion of every pixel between two images. In our implementation the optical flow was estimated according to Deqing Sun
and Stefan Roth’s implementation\(^3\) of the Horn and Schunck’s method [4], computing a motion vector for each pixel, which is then reversed for registration purposes. Motion vectors often correspond to sub-pixel displacements, not allowing a “direct” registration. Two registration processes were considered: (i) rounding the motion vector to pixel resolution (OF); (ii) performing a weighted average according (OF Prob.):

\[
R(i,j) = \sum_{rows} \sum_{cols} I(rows, cols) w_{row} w_{col} \tag{6}
\]

Where \( R \) is the registered image, \( I \) the input image, \( i \) and \( j \) the pixel coordinates, \( u \) and \( v \) the motion vector components, \( rows = \{i + \text{floor}(u), j + \text{ceil}(v)\} \), \( cols = \{i + \text{floor}(u), j + \text{ceil}(v)\} \), and \( w_{row} \) and \( w_{col} \) weights associated with the elements of \( rows \) and \( cols \), respectively. For the first element of \( rows \), \( w_{row} = 1 - \text{abs}(\text{round}(u) - u) \), for the second element, \( w_{row} = \text{abs}(\text{round}(u) - u) \). \( w_{col} \) is computed similarly, replacing \( u \) with \( v \).

By the conducted experiments we concluded that the optical flow registration methods (specially OF Prob.) perform much better than the geometric transformation techniques, which is due to the complex motions of the human face, like local movements, which can’t be handled by simple geometric transformations.

3) Fusion: Conventional fusion processes include a mean or median operation, assuming that registration is perfect. Considering this is not generally true, it is advantageous to adopt a fusion technique resilient to image registration imperfections, such as the one proposed by Elad and Protter [15]. This technique computes the SR image using weighted values from the neighborhood of each pixel.

The present implementation uses a different registration procedure, requiring the fusion technique to also be adapted. The first main difference is that in [15] fusion is performed directly from the LR to the HR space, with HR pixel weights depending on a neighborhood of the corresponding pixel in the LR images. But here images are interpolated to the desired HR size, with the computation of each HR pixel being done according to:

\[
\hat{z}(i,j) = \sum_{|x| \leq N(i,j)} \sum_{|y| \leq N(i,j)} W_t(k,l) |y| |x| |N\{((k,l)-(i,j))\}|_{z,0,\sigma} \sum_{|k| \leq N(i,j)} \sum_{|l| \leq N(i,j)} W_t[k,l] \tag{7}
\]

Where \( N(i,j) \) is a square neighborhood of pixel \( i,j \); \( W_t(k,l) \) the weight associated with pixel \( k,l \) from image \( t \), \( y \) the \( t \)\textsuperscript{th} interpolated LR image, and \( N\{x, \mu, \sigma\} \) is the Gaussian distribution operator with mean \( \mu \) and standard deviation \( \sigma \), evaluated at \( x \). Using Eq. 7 each pixel is computed as a combination of the weighted values of its neighborhood, times a penalizing factor, which reduces the importance of pixels farther away from the neighborhood center. This penalizing factor was not considered in [15], being a contribution of this paper. The weight of a pixel \( k,l \) in the interpolated image \( t \) is computed using:

\[
W_t[k,l] = \exp\left\{ -\frac{||P_{k,l} I_{OF}^{-1}(x-w)||}{2\sigma_P^2} \right\} N\{dk_t + |dl_t|, 0, \sigma_D\} \tag{8}
\]

Where \( z \) is the HR targeted image, \( dk_t \) and \( dl_t \) are the \( k,l \) values of the optical flow in image \( t \), and \( OF^{-1} \) is the reversed optical flow registration information, which yields the simulated interpolated image when applied to \( z \). Since \( z \) is not known, the interpolated reference image is used, which is a relatively good guess. Then, its difference to the real interpolated image \( y_t \) is found, providing an error image. A patch \( P \) is extracted around the \( k,l \) pixel from this error image, and the \( L_1 \)-norm is applied to it. In [15], the equivalent error (not the same because they simulate the LR image, instead of the interpolated one) was computed with the \( L_2 \)-norm. However, we used the \( L_1 \)-norm in order not to penalize in excess relatively small errors. Finally, a displacement penalty is also included, which reduces the importance of pixels that have large motions in the registration. Here we also use the \( L_1 \) instead of the \( L_2 \)-norm to compute the displacement error.

4) Deblurring: The two most relevant kinds of blur are motion blur and the camera’s blurring from its natural Point Spread Function (PSF). The former is implicitly dealt with by using several pictures and registering them to a reference. Therefore, the latter is the one which we need to address. However, we don’t know the PSF from the camera. Also, considering a neighborhood of the pixel in the fusion process causes a kind of blur similar to the one from a low-pass filter. Therefore, we use a blind deconvolution algorithm which deblurs the image and tries to find the PSF, simultaneously. The PSF and deblurred image are found using the Lucy-Richardson method [11] of `deconvblind`.

5) Alpha-blending: For some parts of the image, where the gradient is small and the image values alter slowly, the interpolation of the image may be enough (or even perform better) than applying SR. On the contrary, SR is preferred when sub-pixel displacements from several LR images are needed to reconstruct the HR image. Therefore, blending the interpolated reference image with the super-resolved image may achieve better results. The blending is performed by:

\[
I_{\text{ob}} = \alpha I_{\text{SR}} + (1 - \alpha) I_{\text{int}}, \alpha \in [0,1] \tag{9}
\]

where \( I_{\text{SR}} \) is the super-resolved image and \( I_{\text{int}} \) is the interpolated image. In the conducted experiments, we chose \( \alpha = 0.8 \).

By analysing the results of the proposed technique, we showed that in the region of interest (the face) it outperforms the interpolated image by about 1dB in terms of Peak Signal to noise Ratio (PSNR). Comparing with the state of the art works from [16] and [21], the PSNR values, although slightly smaller, are comparable to theirs, but in terms of Structure Similarity Index (SSIM) our method outperforms both.

\(^3\)https://github.com/ahmadh84/occlusiontracking
III. Results

In this chapter, the whole system will be evaluated in an integrated manner, where the input are simulated surveillance videos which will be subjected to the implemented processes, and the main objective is to recognize the suspects using the super-resolved face images with a better confidence than when using the low-resolution image.

Summarizing, in this chapter the following is addressed using as input simulated surveillance videos:

- Detect the suspects’ face images (with outliers) using BioFoV’s face detector;
- Filter non-face images (outliers) from the detector’s output using the described skin-tone and PCA based filters;
- Cluster the detected faces into groups, where each group contains only face images of one person in a single pose. In the case that some outliers are still present, no outlier should be aggregated in a cluster containing face images;
- Manually choose clusters with more than 3 images in which the suspects’ are in a pose suitable for recognition;
- Create a super-resolved face image for each suspect, using the face images in the chosen clusters;
- For recognition purposes, resize all the images to the same size ($500 \times 500_p$) using bicubic interpolation;
- Try to recognize the super-resolved face against a database of faces;
- Try to recognize several LR faces against the same database of faces;
- Compare the recognition scores for each case.

A. Input videos and face detection

As an input for the system, three videos were used. These videos were recorded in the laboratory of Multimedia Signal Processing Group of Instituto de Telecomunicações, but in different days and with different angles for having different illumination conditions. Two videos have two subjects (considered suspects from now on) and the third has only one. The first sequence contains suspects 1 and 2, the second one has suspects 3 and 4, and the third suspect 5. It is important to notice that suspect 2 and suspect 4 are the same person, but for simplicity the results will be presented as if they were different people.

After passing the videos by BioFoV’s face detector, we obtain:

- Video 1:
  325 Face images;
  263 Outliers. Mean size of images of $83 \times 83_p$
- Video 2:
  222 Face images;
  522 Outliers. Mean size of images of $75 \times 75_p$
- Video 3:
  101 Face images;
  31 Outliers. Mean size of images of $60 \times 60_p$

In the first video, the majority of non-face images are from a window blind, only 4 outliers are from different objects. In the third video 29 of the 31 outliers are also very similar. In the second video, there is a wider variety of outliers.

B. Pre-Processing

Both skin-tone based and PCA based filters were used to remove the outliers. In video 1, the skin-tone based filter removed every outliers except for one (a yellowish image), which is very acceptable since its color resembles the skin-color. The PCA based filter didn’t filter any additional outlier. For video 2, the skin filter removed 478 outliers. The PCA filter identified 52 outliers, but only one of them wasn’t also removed by the skin-filter. Therefore, 43 non-face images passed through the filters and were present in the clustering process. In both these videos no face image was mistakenly filtered out. In video 3 the skin filter removed all the outliers, however, it also removed 24 face images that were too bright. The PCA filter correctly removed 29 outliers, none of them corresponding to face images. The pre-processing results are summarized in Table 1.

By the obtained results, we can verify that the pre-processing module is very successful. The skin tone filter has identified 99.6%, 91.6% and 100% of the outliers in videos 1, 2 and 3, respectively. The drawback is that in video 3, it also wrongly removed 23.8% of face images due to excessive illumination, but these wrong removals didn’t have a negative impact on the final results.

IV. Clustering and Super-Resolution

For the clustering process, the described HAC algorithm using spatial information was used. The chosen metric was Complete-Linkage with $\gamma = 0.7$, to have the minimal amount of cluster mixtures (faces of different subjects in the same cluster), and the maximum similarity between poses intra-cluster, while maintaining a reasonable amount of faces in each cluster.

For video 1, 66 clusters were obtained. Two of them had mixtures, namely a face of subject 2 in clusters where the majority of images were from subject 1. This happened because the faces in the mixtures had very similar poses. The chosen clusters for super-resolution had 11 and 9 face images for subject 1 and 2, respectively. They were chosen because the face images are frontal and, therefore, suitable for face recognition. For video 2, 77 clusters were obtained, and none of them had mixtures. Here the chosen clusters had 9 images each. The non-face images that were present in the process were in different clusters from the face images, so they didn’t interfere with the process. Clustering for video 3 resulted in 20 clusters. The chosen cluster had 10 images.

Finally, the proposed SR algorithm was used to obtain a higher resolution face image from the images present in the chosen clusters. The final SR image, is obtained with the alpha-blending technique, having $\alpha = 0.8$. 
V. FACE RECOGNITION

In order to evaluate the usefulness of the proposed SR technique, a face recognition experiment is performed. This experiment was done using Kernel Feature Analysis face recognition, according to the implementation presented in [19], [18]. The database used to train the recognizer was the AT&T’s “Database of Faces” [17] along with some facial images of each suspect, captured in a controlled environment with appropriate lighting conditions.

The matching scores are obtained using the euclidean distance between the feature vectors from the training images and the ones from the test images. The matching results for the 5 suspects are summarized in Table 2. The SR x rows are relative to the Super-resolved image of suspect x, while the LR x rows refer to several Low-Resolution images from suspect x. Some representative example images used in when performing face recognition can be seen in Fig. 2. In this figure, the SR technique doesn’t seem to visually improve the LR images. This happens because, in order to fit side by side with the LR images, the SR images must be down-sized, losing some information.

Starting by the results from Suspect 5, one can verify that the SR image has significantly better matching scores than all the other test images, clearly proving the value of the proposed technique. With the first suspect, we are able to see that the Super-Resolved image doesn’t have a better matching score with Suspect 1 than the other images. However, we can also verify that none of the test images from this set is clearly matched to Suspect 1 (there are better matching scores to Suspect 3 and Suspect 4). This happens because Suspect 1 has a beard in the training images and not in the video sequence. The test images of Suspect 2 didn’t have favourable lighting conditions, however the SR image clearly improved the face recognition performance, specially because some LR images were matched to Suspect 4. The images in the cluster of Suspect 3 had some motion blur, for this reason the SR image didn’t improve the LR images significantly. Therefore, the face recognition scores are similar for the SR and the LR images. For suspect 4, the SR image also presents an improvement in the matching scores when compared to the LR ones (only the last LR image yielded a similar score).

Notice that the chosen training images have similar poses as the test images. This was done on purpose to understand the importance of pose on face recognition. Recall that Suspect 2 and Suspect 4 are the same person but have training images with different poses. Ideally, because they are the same person, the scores between images of Suspects 2 and 4 should be similar. However the euclidean distances between the test images from Suspect 4 and the training images from Suspect 2 are very high. On the contrary, the training images from Suspect 2 and Suspect 4 have similar distances to the test images from Suspect 2. This indicates that the pose of the subjects (and general image conditions beyond resolution) can be very important to face recognition. However, if in a similar pose as the training images, Super-Resolution may indeed help recognizing the suspects, which is crucial in a forensic scenario where every improvement is important.

VI. CONCLUSION AND FUTURE WORKS

In this work we presented a way to detect and cluster face images from a surveillance video, to use them as input to a proposed multi-frame SR algorithm, in order to obtain a higher resolution image from an individual for forensic purposes. We used colour information and PCA to remove outlier images from the face detector, then SIFT features were extracted and the similarity between images was computed based on them. HAC was used for clustering, and different parameters and metrics were analysed. Then, we addressed some registration techniques and chose the one which yielded the best results for the SR algorithm. This algorithm is a direct one, which contains an image fusion step. This step was based on the state of the art work from [15], but adapted to our registration. The results from the proposed algorithm when in favourable conditions, improved face recognition scores when compared with LR images, suggesting that they are suitable to be used in real life scenarios. As for future works, we can impose geometry conditions to the skin colour filter or use machine learning techniques to improve it; use a better suited database of faces in order to improve the generalization of the PCA-based filtering; if the video has an acceptable frame rate; include temporal information when clustering; find the optimal value for $\gamma$ for each video sequence; find the optimal relation between the fusion parameters and the input images characteristics; and add the proposed system to the BioFoV platform.

REFERENCES

Fig. 2. Representative example images of what is being matched in Table 2. The images in the LR x columns are a single example of the matched LR images set.


Table 1: Pre-Processing Results.

<table>
<thead>
<tr>
<th>Pre-Processing</th>
<th>Video 1</th>
<th>Video 2</th>
<th>Video 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face Images</td>
<td>325</td>
<td>222</td>
<td>101</td>
</tr>
<tr>
<td>Outliers</td>
<td>263</td>
<td>522</td>
<td>31</td>
</tr>
<tr>
<td>Total</td>
<td>588</td>
<td>744</td>
<td>132</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pre-Processing</th>
<th>Video 1</th>
<th>Video 2</th>
<th>Video 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skin Filtering</td>
<td>0</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>PCA Filtering</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Remaining</td>
<td>325</td>
<td>222</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>43</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>326</td>
<td>265</td>
<td>77</td>
</tr>
</tbody>
</table>

Table 2: Matching results (Euclidean distance) between test images and the five suspects

<table>
<thead>
<tr>
<th>Test Imgs</th>
<th>Susp 1</th>
<th>Susp 2</th>
<th>Susp 3</th>
<th>Susp 4</th>
<th>Susp 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR1</td>
<td>316</td>
<td>467</td>
<td>302</td>
<td>314</td>
<td>454</td>
</tr>
<tr>
<td>LR1</td>
<td>411</td>
<td>464</td>
<td>345</td>
<td>268</td>
<td>549</td>
</tr>
<tr>
<td>LR2</td>
<td>329</td>
<td>462</td>
<td>301</td>
<td>281</td>
<td>458</td>
</tr>
<tr>
<td>LR3</td>
<td>311</td>
<td>468</td>
<td>297</td>
<td>307</td>
<td>447</td>
</tr>
<tr>
<td>LR4</td>
<td>297</td>
<td>471</td>
<td>271</td>
<td>312</td>
<td>436</td>
</tr>
<tr>
<td>LR5</td>
<td>305</td>
<td>464</td>
<td>279</td>
<td>313</td>
<td>441</td>
</tr>
<tr>
<td>Remaining</td>
<td>302</td>
<td>462</td>
<td>287</td>
<td>325</td>
<td>437</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Imgs</th>
<th>Susp 1</th>
<th>Susp 2</th>
<th>Susp 3</th>
<th>Susp 4</th>
<th>Susp 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR2</td>
<td>603</td>
<td>291</td>
<td>431</td>
<td>343</td>
<td>442</td>
</tr>
<tr>
<td>LR1</td>
<td>521</td>
<td>351</td>
<td>404</td>
<td>282</td>
<td>482</td>
</tr>
<tr>
<td>LR2</td>
<td>541</td>
<td>322</td>
<td>423</td>
<td>300</td>
<td>474</td>
</tr>
<tr>
<td>LR3</td>
<td>560</td>
<td>304</td>
<td>434</td>
<td>313</td>
<td>473</td>
</tr>
<tr>
<td>LR4</td>
<td>559</td>
<td>300</td>
<td>434</td>
<td>314</td>
<td>467</td>
</tr>
<tr>
<td>LR5</td>
<td>570</td>
<td>283</td>
<td>437</td>
<td>317</td>
<td>462</td>
</tr>
<tr>
<td>Remaining</td>
<td>502</td>
<td>405</td>
<td>386</td>
<td>268</td>
<td>498</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Imgs</th>
<th>Susp 1</th>
<th>Susp 2</th>
<th>Susp 3</th>
<th>Susp 4</th>
<th>Susp 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR3</td>
<td>473</td>
<td>522</td>
<td>221</td>
<td>373</td>
<td>337</td>
</tr>
<tr>
<td>LR1</td>
<td>463</td>
<td>505</td>
<td>182</td>
<td>331</td>
<td>366</td>
</tr>
<tr>
<td>LR2</td>
<td>497</td>
<td>496</td>
<td>186</td>
<td>350</td>
<td>378</td>
</tr>
<tr>
<td>LR3</td>
<td>508</td>
<td>502</td>
<td>178</td>
<td>357</td>
<td>406</td>
</tr>
<tr>
<td>LR4</td>
<td>492</td>
<td>507</td>
<td>203</td>
<td>379</td>
<td>389</td>
</tr>
<tr>
<td>LR5</td>
<td>474</td>
<td>501</td>
<td>247</td>
<td>410</td>
<td>392</td>
</tr>
<tr>
<td>Remaining</td>
<td>474</td>
<td>506</td>
<td>298</td>
<td>451</td>
<td>355</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Imgs</th>
<th>Susp 1</th>
<th>Susp 2</th>
<th>Susp 3</th>
<th>Susp 4</th>
<th>Susp 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR4</td>
<td>516</td>
<td>511</td>
<td>383</td>
<td>228</td>
<td>545</td>
</tr>
<tr>
<td>LR1</td>
<td>546</td>
<td>507</td>
<td>387</td>
<td>256</td>
<td>555</td>
</tr>
<tr>
<td>LR2</td>
<td>546</td>
<td>507</td>
<td>387</td>
<td>256</td>
<td>555</td>
</tr>
<tr>
<td>LR3</td>
<td>521</td>
<td>550</td>
<td>406</td>
<td>267</td>
<td>592</td>
</tr>
<tr>
<td>LR4</td>
<td>562</td>
<td>517</td>
<td>400</td>
<td>243</td>
<td>540</td>
</tr>
<tr>
<td>LR5</td>
<td>567</td>
<td>465</td>
<td>379</td>
<td>249</td>
<td>518</td>
</tr>
<tr>
<td>Remaining</td>
<td>534</td>
<td>544</td>
<td>402</td>
<td>255</td>
<td>584</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Imgs</th>
<th>Susp 1</th>
<th>Susp 2</th>
<th>Susp 3</th>
<th>Susp 4</th>
<th>Susp 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR5</td>
<td>492</td>
<td>359</td>
<td>349</td>
<td>382</td>
<td>175</td>
</tr>
<tr>
<td>LR1</td>
<td>439</td>
<td>350</td>
<td>359</td>
<td>384</td>
<td>223</td>
</tr>
<tr>
<td>LR2</td>
<td>457</td>
<td>382</td>
<td>359</td>
<td>384</td>
<td>223</td>
</tr>
<tr>
<td>LR3</td>
<td>491</td>
<td>327</td>
<td>423</td>
<td>405</td>
<td>269</td>
</tr>
<tr>
<td>LR4</td>
<td>417</td>
<td>371</td>
<td>385</td>
<td>381</td>
<td>261</td>
</tr>
<tr>
<td>LR5</td>
<td>429</td>
<td>387</td>
<td>344</td>
<td>364</td>
<td>214</td>
</tr>
</tbody>
</table>