Building Video Databases to Boost Performance Quantification – The DXM2VTS Database

Dereje Teferi*, Josef Bigun*
*School of Information Science, Computer, and Electrical Engineering (IDE)
Halmstad University
P.O.Box 823, SE-301 18
Halmstad, Sweden
{Dereje.Teferi,Josef.Bigun}@ide.hh.se

Abstract

Building a biometric database is an expensive task which requires high level of cooperation from a large number of participants. Currently, despite increased demand for large multimodal databases, there are only a few available. The XM2VTS database is one of the most utilized audio-video databases in the research community although it has been increasingly revealed that it cannot quantify performance of a recognition system in the presence of complex background, illumination, and scale variability. However, producing such databases could mean repeatedly recording a multitude of audio-video data outdoors, which makes it a very difficult task if not an impossible one. This is mainly due to the additional demands put on participants. This work presents a novel approach to audio-visual database collection and maintenance to boost the performance quantification of recognition methods and to increase the efficiency of multimodal database construction. To this end we present our segmentation procedure to separate the background of a high-quality video recorded under controlled studio conditions with the purpose to replace it with an arbitrary complex background. Furthermore, we present how an affine transformation and synthetic noise can be incorporated into the production of the new database to simulate real noise, e.g. motion blur due to translation, zooming and rotation. The entire system is applied to the XM2VTS database, which already consists of several terabytes of data, to produce the DXM2VTS – Damascened XM2VTS database essentially without an increase in resource consumption, i.e. storage space, video operator time, and time of clients populating the database. As a result, the DXM2VTS database is a damascened (sewn together) composition of two independently recorded real image sequences that consist of a choice of complex background scenes and the the original XM2VTS database.

1 Introduction

Biometrics is an important and increasingly challenging research topic. It is the automatic recognition of individuals based on who they are (e.g. face, iris, fingerprint etc) and/or what they can do (e.g. voice, signature) instead of what they hold (e.g. ID cards) and/or what they remember (e.g. PIN, password) (Ortega-Garcia et al., 2004). Biometric data such as face, fingerprints, iris, ear, hand, and voice are used to track and analyze features and uniquely identify and verify people. Such systems are built through years of research, testing, experimentation and innovation. Their performance quantification depends on, among other things, the size and variety of the database used.

Biometrics technology gives a high level of authentication and is successful in many ways but has not yet been fully trusted by users due to fraud and impersonation. Even so, they have become an integral part of infrastructure used for diverse business sectors such as security, finance, health, law enforcement etc (Jain et al., 2004a), (Bailly-Baillière et al., 2003). Practical biometric systems should be accurate to a specified level, fast, harmless to users, and accepted by users as well (Ortega-Garcia et al., 2004), (Jain et al., 2004b). User acceptance and high confidence can be achieved by training, testing and evaluating these biometric systems on variety of large databases recorded in real world environment, using multiple modalities for authentication, and incorporating aliveness detection into the systems.

One of the major problems in the development of biometric systems is the lack of public large databases acquired under real-world environment for training, testing and evaluation. Those systems trained and tested on databases recorded in realistic working and living environments perform evidently better. However, most biometric databases are built in a studio and have near-constant color background such as XM2VTS (blue), CUAVE (green) etc. Although one color background does not represent real life scenery, it facilitates segmentation of the background to replace it with a realistic one. However, this has not been done before for a variety of reasons. Whereas the background is illuminated uniformly in CUAVE enabling chroma-keying of different backgrounds (Patterson et al., 2002), this is not the case in XM2VTS as the background is not uniformly blue, Fig.1.

The need to collect new databases, consuming huge resources and claiming important maintenance resources, to represent real world situations is stated in (Bailly-Bailliére et al., 2003) as:

> *The XM2VTS database, together with the Lausanne protocol contains 295 subjects recorded over 4 sessions. However, it was not possible to use it as the controlled recording environment was not realistic enough compared to the real world situations when one makes a transaction at home through a consumer web cam or through an ATM in a variety of surroundings. Therefore it was decided that a new database for the project would be recorded and a new experimental protocol using the database defined.*

Recording a database in real world (non-controlled) environments representing multiple scenarios is difficult if not impossible. This is due to the high demand it puts on the participants to repeatedly appear on weekly or monthly intervals to real-life environments (e.g. malls, busy stations, streets, etc) for recording. To this end, we propose a method to segment the image sequences of the XM2VTS database and create a binary mask of the background and the foreground. These masks will be used to generate various damascened XM2VTS databases with realistic background, such as shopping centers, moving cars, people, additionally, they will be used to add scale variation, illumination, motion and zooming blur to improve performance quantification of biometric systems.

This paper is organized as follows. The next two sections give some background on the XM2VTS database and its specific demands on image segmentation. In section four and its subsections we present our approach. The subsections include the three stages of the proposed system: segmentation, compression of the binary mask, and building the damascened database. In section five we present the experiment conducted and finally our conclusions are summarized.

2 The XM2VTS Database

The XM2VTS database is a 295 subject audio-video database that offers synchronized video and speech data as well as side view images of the subjects. Its design is based on the experience of its predecessor, the M2VTS database which is comprised of 37 subjects. The database acquisition started with 360 subject and 295 of them completed the whole session of four recordings. These subjects are recorded inside a studio in four separate sessions in a uniformly distributed period of five months. This ensures the natural variability of styles and moods of subjects over different times. The database is intended for researchers in areas related to multimodal recognition systems (Messer et al., 1999).

2.1 Performance quantification

XM2VTS has associated evaluation protocols to assess the performance of vision and speech based systems developed using this database as a training and test bed. It has been used in the published studies of many researches and therefore it is considered as a useful comparison yard-stick (Bailly-Bailliére et al., 2003). The results from two face verification contests on the XM2VTS database that were organized in conjunction with the 2000 International Conference on Pattern Recognition and the AVBPA 2003 are reported in (Matas et al., 2000) and (Messer et al., 2003) respectively.

However, XM2VTS and many other face image databases are designed mainly to measure performance of face recognition methods and thus the images contain only one face. Near-constant background one-face images does not require the system to track the faces from complex backgrounds. They are suitable for training recognition systems rather than testing as the tacit reason for comparing classifiers on test sets is that these data sets represent problems that systems might face in the real world. Moreover, it is assumed that superior performance on these benchmarks may translate to superior performance on other real-world tasks (Yang et al., 2002). To this effect, it is important to boost the performance quantification of recognition systems trained on the XM2VTS database by synthesizing the image sequences with a real-world background and a simulation of natural noise.

Accordingly, research in biometrics need many databases similar to that of XM2VTS, with varying background, to build safer and better systems for the ever increasing security requirements of the public at large.

3 Image Segmentation

Image segmentation is used and applied in many computer vision research and applications. It is a problem for which no general solution exist. It is broadly defined as partitioning of pixels into groups with similar properties.

Generally color (in different color spaces), edge, texture and relative spatial location are the factors that define the properties of an image. These factors have been used to develop many algorithms for image segmentation. A survey of many image segmentation techniques is discussed in (Cheng et al., 2001) and (Lucchese and Mitra, 2001). Although many of the algorithms developed do not require training, some application specific knowledge needs to be provided as a parameter, typically a threshold on within-class variance or the number of classes.

There exists a multitude of clustering techniques such as hierarchical clustering, kmeans, and fuzzy c-means that are standard in many mathematical software packages. However, they need to be adapted to the needs of automatic image segmentation e.g. the spatial continuity of the assigned classes are not guaranteed which in turn requires post processing, or when thousands of images are individually segmented the number of classes parameter that is needed as input may vary, which in turn prompts for manual intervention. Below, we summarize the various stages and the adaptation that we have undertaken to minimize the manual intervention during the automatic segmentation of hundreds of thousands of images included in the XM2VTS database.

4 Segmentation, Adding Variability and Building the New Database

4.1 Segmentation

Histogram thresholding method is tested to see if we can separate the background of image sequences from the XM2VTS video database. However, finding a single threshold group that works for hundreds of thousands of images can not be achieved due to varying illumination of the background within an image as well as from image sequence to image sequence. As a result, the segmentation showed poor performance, Fig. 1.

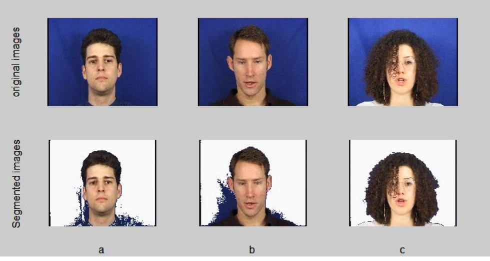

Figure 1: Segmentation by histogram thresholding

Here we present the technique we used to separate a near-constant color background from a face-shoulder image of a video. We applied it on the XM2VTS database to create the necessary mask for the rest of the application. The segmentation technique is an iterative procedure that uses two feature spaces. The first iteration creates a crude set of classes and then a refinement process follows that iteratively adjusts the class assignments and their centroid. This procedure is presented below.

4.1.1 Clustering

A low-pass filtering using a Gaussian filter is applied to smooth the image before segmentation to increase the signal to noise ratio. The Gaussian filter G is a separable filter and has the parameter σ:

$$G = exp(-(x^2 + y^2)/2\sigma^2) \qquad (1)$$

The segmentation algorithm is applied on the smoothed image. The proposed segmentation does not require *a priori* knowledge of the number of clusters in the image as this is automatically determinable as will be explained below. Automatic determination of class numbers is necessary in our work as we have thousands of image sequences to segment and the number of clusters may vary from image sequence to image sequence depending on the color and texture of the subjects' clothing.

4.1.2 The feature space and the clustering criteria

The segmentation process needs a *feature space*, which is normally a vector space, equipped with a

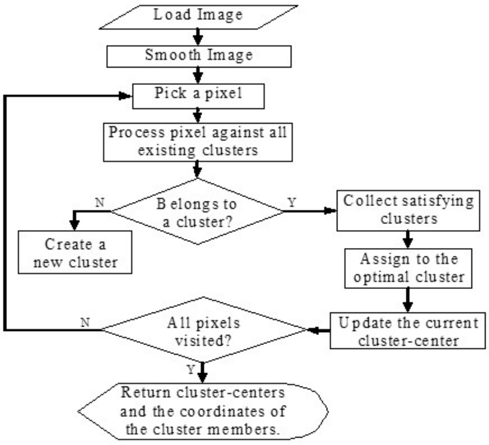
Load Image
Smooth Image
Pick a pixel
Process pixel against all existing clusters
Belongs to a cluster?
N
Y
Create a new cluster
Collect satisfying clusters
Assign to the optimal cluster
Update the current cluster-center
All pixels visited?
N
Y
Return cluster-centers and the coordinates of the cluster members.

Figure 2: The segmentation algorithm

metrics to measure distances. We use here two such vector spaces jointly. The first one consists of the ordinary rgb-color space equipped with the max-norm. To be precise, let $\mathbf{f}_{ij}$ represent the color vector of a pixel at row i, column j of an image then

$$\mathbf{f}_{ij} = (f_{ij}^1, f_{ij}^2, f_{ij}^3)^T \tag{2}$$

where the superscripts 1, 2, and 3 represent the color components red, green, and blue.

The max-norm of a vector $\mathbf{f}$ is then defined as,

$$\|\mathbf{f}\|_\infty = \|(f^1, f^2, f^3)^T\|_\infty = \max_k\{f^k\} \tag{3}$$

where $\quad k \in \{1, 2, 3\}$

Having the same metrics as above, the second feature vector space is constructed from the first one as follows:

$$\begin{aligned}\tilde{\mathbf{f}}_{ij} &= (\tilde{f}_{ij}^1, \tilde{f}_{ij}^2, \tilde{f}_{ij}^3)^T \\ &= (f_{ij}^1 - f_{ij}^2, f_{ij}^2 - f_{ij}^3, f_{ij}^3 - f_{ij}^1)^T \end{aligned} \tag{4}$$

This feature space is introduced to split regions that have too high hue variations since it more directly measures the direction of an rgb-color vector, the hue, than the first feature space.

To describe the clustering algorithm, we need to define *the distance to cluster-center vector* of a pixel at row i, column j as

$$\delta_{ij}^l = (\|\mathbf{f}_{ij} - \mathbf{c}^l\|_\infty, \|\tilde{\mathbf{f}}_{ij} - \tilde{\mathbf{c}}^l\|_\infty)^T \tag{5}$$

where $\mathbf{c}^l$ is the cluster-center vector of the cluster with label l and $\mathbf{f}_{ij}$ represents the rgb-color vector of a pixel. The vectors $\tilde{\mathbf{c}}^l$, $\tilde{\mathbf{f}}_{ij}$ are the cluster-center vector, and the feature vector of the pixel at row i, column j but measured in the second feature space.

The clustering and segmentation is achieved by iteratively updating the partitioning, and the cluster-centers, for every advancement of the pixel position, Fig. 2. The pixel position advancement follows the scan-direction, which attempts to achieve a spatial continuity of the labels being obtained in the segmentation process. A pixel is considered to belong to a cluster, if its distance to cluster-center vector is within certain limits,

$$0 \le \delta_{ij}^l \le \tau$$

where $\tau = (\tau^1, \tau^2)^T$ represents a threshold vector, which determines the maximum variation of the distance to cluster-center vector allowed within the same cluster, in each feature space independently. It is worth noting that in traditional clustering algorithms used in image segmentation, the pixel positions have no-influence on cluster-assignments. Yet, since the physics of imaging is continuous w.r.t. pixel positions, the class-labels should change smoothly, prompting for an additional step attempting to remove "salt-pepper" classes scattered around large classes.

The procedure has been implemented by visiting pixels in three different (sequential(scan-line), random, and hilbert curve) directions. For the XM2VTS database, we have verified that the segmentation responded better for the hilbert and the scan-line traversals. This is mainly attributed to the continuity of intensity changes in the background of the images. The sequential visiting order is chosen for efficiency reasons only. This order of presenting pixels to the clustering scheme attempts to enforce continuity at least line-wise.

A pixel's distance to all available cluster-centers are computed and the pixel is assigned to the "closest" cluster, the nearest neighbor assignment. To give a meaning to "closeness", in the combined feature spaces, the metrics of the two feature spaces are merged in a straightforward manner as

$$\|\delta_{ij}^l\|_\infty = \|(\|\mathbf{f}_{ij} - \mathbf{c}^l\|_\infty, \|\tilde{\mathbf{f}}_{ij} - \tilde{\mathbf{c}}^l\|_\infty)^T\|_\infty \tag{6}$$

Finally a cluster-list containing all the clusters and their centroids is generated from this procedure, Table 1. This is used as an initial cluster for the cluster refinement procedure described next.

Table 1: Cluster matrix

Cl	Red	Green	Blue	N_pxls	Members
1	0.153	0.133	0.134	4837	List_1
2	0.159	0.221	0.518	37341	List_2
3	0.186	0.259	0.614	27735	List_3
4	0.120	0.153	0.293	1688	List_4
.	.	.	.	.	.
.	.	.	.	.	.

4.1.3 Cluster refinement

Cluster refinement is a procedure applied on the image to refine the cluster-centers as well as the their member pixels. This procedure is necessary in our method as the segmentation above terminates upon visiting each pixel once. We use the same procedure as above when refining the clusters except that the cluster matrix we obtained from the initial segmentation is used as the start cluster list for the refinement. The segmentation result converges by running this procedure repeatedly while automatically adjusting the class centroids as well as the number of classes over each iteration.

Once the clusters are refined, the background class is identified as the the cluster with the largest member pixels. We verified that this hypothesis indeed holds very well for the XM2VTS database, although more refined approaches, such as the cluster containing most boundary pixels, and the cluster-center closest to a certain set of example background colors, can be used either separately or in combination. Finally, a 2D mask is created by identifying the background cluster pixels and assigning them the logical value one whereas the members of the remaining clusters (the foreground) are set to zero.

4.1.4 Improving the efficiency of segmentation

The speaking head shots of the XM2VTS database comprise about 75% of the overall image sequences. They are used for speech and/or face-recognition applications. These shots contain only small amount of motion of the person as a natural behaviour of the subjects when speaking. We exploited this feature to improve the efficiency of our segmentation algorithm for the speaking head shots of the XM2VTS database. That is, the segmentation result of the first image can be used effectively for the segmentation of the remaining images within an image sequence.

The binary mask we obtained from the segmentation of the first frame is used to create a narrow band around the border of the foreground image part, which we below refer to as *the narrow band*. In the consecutive frames of the image sequence only the image part corresponding to the pixel coordinates of the narrow band, from the binary mask of the first frame, is passed through the segmentation scheme presented above.

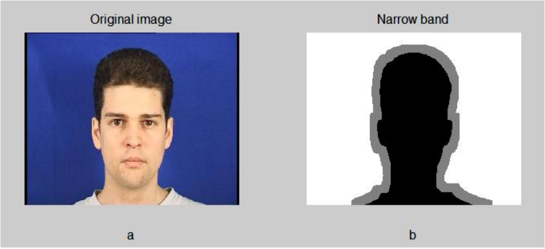

Figure 3: Narrow band around the foreground

The motion of the subject is only within the marked narrow band, as shown in Fig. 3. Therefore we know that the zero, or the black label, represents the foreground whereas the white part is the background in all frames of the video. Accordingly, the segmentation process for the consecutive frames is applied on the narrow band and the part that belongs to the background is set to one and the rest to zero. This result is subsequently used as a binary image mask for the specific frame in the image sequence.

This procedure could not be applied for the rotating heads image sequences as the motion of the subjects cover almost all areas of the frame. Thus, for the case of the rotating-head shots, the segmentation procedure is applied on the whole image to create the binary mask of each frame.

4.2 Compression of the binary mask

Each binary mask of a video is basically an *fr* dimensional binary image of size (r,c,fr), where (r,c) is the size of an image in the sequence and fr is the number of frames in the video. Storing these binary masks of every frame of the image sequence is possible but not efficient especially for distribution as the size of video databases, such as that of XM2VTS, is very large. Thus, compression is found to be necessary. There are several compression algorithms available. The test results of MPEG-4 and RLE – Run Length Encoding are presented below.

4.2.1 Compression using MPEG-4 encoding

MPEG-4 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group) in 1999. The fully backward compatible extensions under the title of MPEG-4 Version 2 became an international standard in 2000[1].

The binary mask of each image sequence is converted to an Audio/Video Interleaved (AVI) file and compressed[2]. The compression level (lossy) at a data rate of 512 Kb/sec is only 10% whereas at a data rate of 256 Kb/sec is about 54%. However, the compression is not only lossy but also visually degraded on the latter case and therefore not convenient for our purpose.

4.2.2 Compression using RLE

Run Length Encoding (RLE) is a process of searching for repeated runs of a symbol in an input stream and replacing them by a single instance of the symbol and its run count. The repetitive appearance of 1's and 0's in the binary masks makes it convenient for RLE compression. Moreover, the binary mask, in our case contains only two values, 1's and 0's, and therefore modifying the RLE algorithm by storing the first symbol once and then only the run count of the consecutive symbols is possible. That is, we only store the run count of the symbols and not the symbols themselves, Table 2. This again improves the RLE algorithm by reducing the size of the compressed mask substantially. Furthermore, we use a zero delimiter as a separator between frame masks.

Table 2: RLE vector

First symbol	count of first symbol	count of next symbol	.	.	end of frame	.
1	37234	35	.	.	0	.

The size of the RLE vector is 70 – 80% less than the size of the original binary mask. Moreover, it is easy to store, distribute and process with less memory requirement.

Therefore, the modified RLE algorithm is implemented in this system as it performed better for the specific case of compressing the binary masks of a face-shoulder video database.

4.3 Building the damascened database

The next step is building the damascened XM2VTS image sequence from the compressed mask. The mask is decompressed using a reverse-algorithm of the one that is used for compression. This generates the narrow band matrix. Then, we use each row of the narrow band matrix as the values of the narrow band of a binary image mask to generate the binary mask of the whole image sequence.

Parallel frames are extracted from the mask, its corresponding XM2VTS video, and the new background video, one at a time to build the damascened database. In practice, sewing together, or damascening, the two real image sequences according to the mask, can be easily achieved by multiplying the binary mask with the background, its inverse with the XM2VTS frame and adding the result to get the required frame of the synthetic image sequence. At this point, distortion is added as required on the synthetic frame to simulate real recording environments, Fig. 4. Real videos can be distorted or blurred for a variety of reasons. For example, while recording objects moving with high speed, the camera must often be moved manually to keep the moving object in the center. When tracking a still object when the camera is moving, e.g. in a car, high speed motion blur is generated for the rest of the objects that are not tracked. Noise could also occur from poor quality of camera used or due to weather conditions at the time of recording.

To simulate a significant portion of the natural noise, we suggest to apply distortions to the synthetic image sequence according to the following models:

- Horizontal blur: translation of pixels in the x directions;
- Vertical blur: translation of pixels in the y directions;
- Zoom blur: translation of pixels in the radial direction;
- General motion blur: affine transformation
- Imaging noise: Salt and pepper noise at different noise intensity levels.

A set of standard video backgrounds such as offices, outdoors, malls, moving cars is collected and offered with the compressed mask of the XM2VTS database for distribution. These data can be used to generate damascened XM2VTS database with required level of noise and the variability of the background due to the real life scenario changes.

5 Experimental results

A series of experiments are conducted on 55 randomly selected XM2VTS database sequences to de-

[1] The MPEG Home Page. `http://www.chiariglione.org/mpeg/`

[2] Compressed using MPEG-4 encoding by an open source software virtualDub, `http://www.virtualdub.org/`

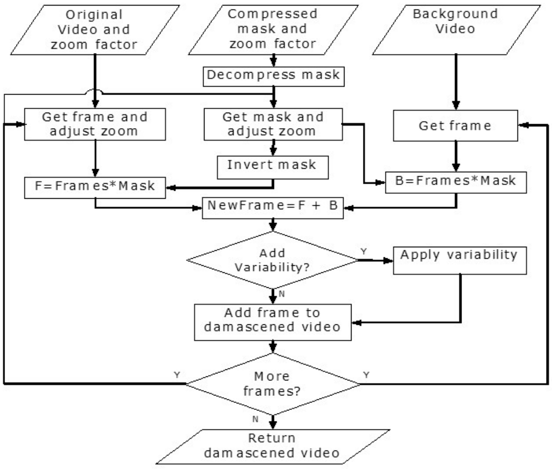

Figure 4: Algorithm for building the new video

termine the optimal threshold for distance to cluster-centers and intensity variations, τ_1, τ_2 where the background would be separated as one cluster. These values are empirically found to be 0.35 and 0.23, assuming an image where the rgb-color components of pixels are between 0 and 1. In addition, the motion of the subjects while speaking is found to be within a 24 pixel wide narrow band. The Gaussian filter used to smooth the image frames before segmentation is set to be of size (7x7) and $\sigma = 2.5$. Moreover, only two iterations are found to be necessary to separate the background of the XM2VTS images.

Using these parameters, 94% of the image sequences of the database are segmented successfully. Since the segmentation of consecutive images is done only on the narrowband that is collected from the first frame of the image sequence, the remaining 6% of the image sequences are segmented semi-automatically by human intervention only on the first frame to find the narrow band.

A software to generate synthetic XM2VTS database from the binary masks is built using Matlab 7.1. A set of representative distortion parameters such as salt and pepper noise, blurring and affine transformation are included on the software to simulate motion of camera, rotation, blurring etc in the output synthetic video as shown in Fig. 5.

6 Conclusion

The segmentation algorithm in our system uses a variant of the max-norm applied to two color spaces simultaneously. These metrics are combined together to yield a segmentation accuracy of 94% at the level of image sequences, i.e. when checked visually, the boundaries found appeared natural to a human observer (the authors).

Due to its public availability, currently, the XM2VTS database is one of the largest, and probably the most utilized biometric multimodal (face and audio) databases in existence, at least in academic research. A method is presented in this paper to store a binary mask for the speaking image sequences of this database. The result which contains a collection of compressed masks is suggested to be used to sew together the XM2VTS database head-shoulders with complex image sequences to obtain damascened image sequences. The boundaries have a high accuracy whereas the damascened sequence contains realistic, yet controllable distortions, such as the amount of the motion blur.

Acknowledgment

This work has been sponsored by the Swedish International Development Agency (SIDA).

References

E. Bailly-Bailliére, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messor, V. Popovici, F. Porée, B. Ruiz, and J. Thiran. The BANCA database and evaluation protocol. In *Audio and Video Based Person Authentication: AVBPA 2003, LNCS 2688*, pages 625–638, 2003.

H.D. Cheng, X.H. Jiang, and J. Wang. Colour image segmentation: Advances and prospects. *Pattern Recognition Letters*, 34:1277–1294, 2001.

A.K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A. Ross. Biometrics: A grand challenge. In *Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004*, volume 2, pages 935–942, 2004a.

A.K. Jain, A. Ross, and S. Prebhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics*, 14(1), January 2004b.

L. Lucchese and S.K. Mitra. Colour image segmentation: A state-of-the-art survey. In *Image Processing, Vision, and Pattern Recognition, Proc. of the Indian National Science Academy (INSA-A)*, volume 67A(2), pages 207–221, March 2001.

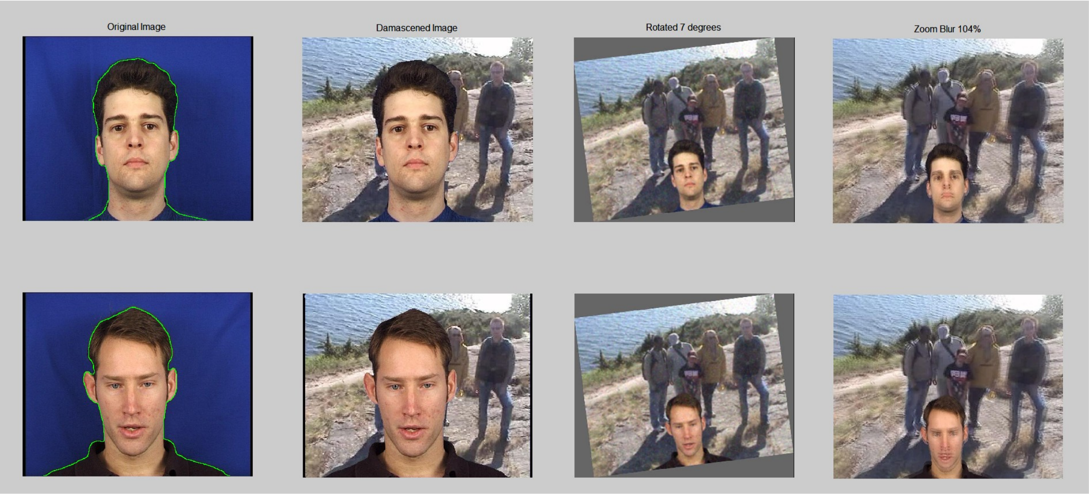

Figure 5: Image frames from the damascened video

J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Patas, T. Tan, H. Yan, F. Smeraldi, J. Bigun, N. Capdevielle, W. Gerstner, S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Comparison of face verification results on the XM2VTS database. In *Proceedings of International Conference on Pattern Recognition. ICPR-15*. IEEE Computer Society, 2000.

K. Messer, J. Matas, J. Kitler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *2nd International Conference on Audio and Video-based Biometric Person Authentication (AVBPA'99)*, pages 72–77, 1999.

K. Messer, J. Kittler, M. Sadeghi, S. Marcel, C. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, J. Czyz, L. Vandendorpe, S. Srisuk, M. Petrou, W. Kurutach, A. Kadyrov, R. Paredes, B. Kepenekci, F.B. Tek, G.B. Akar, F. Deravi, and N. Mavity. Face verification competition on the XM2VTS database, 2003.

J. Ortega-Garcia, J. Bigun, D. Reynolds, and J. Gonzalez-Rodriguez. Authentication gets personal with biometrics. *IEEE Signal Processing Megazine*, 21(2):50–62, March 2004.

E.K. Patterson, S. Gurbuz, Z. Tufekci, and J.N. Gowdy. CUAVE: A new audio-visual database for multimodal human-computer interface research. In *International Conference on Acoustics, Speech, and Signal Processing, Proceedings. (ICASSP 2002)*, pages 2017–2020, May 2002.

M-H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligece*, 24(1): 34–58, January 2002.