Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 2143–2156
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Damascening video databases for evaluation of face tracking and recognition – 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

Received 31 May 2006; received in revised form 10 April 2007
Available online 29 June 2007

Communicated by G. Sanniti di Baja

Abstract

Performance quantification of biometric systems, such as face tracking and recognition highly depend on the database used for testing the systems. Systems trained and tested on realistic and representative databases evidently perform better. Actually, the main reason for evaluating any system on test data is that these data sets represent problems that systems might face in the real world. However, building biometric video databases with realistic background for testing is expensive especially due to its high demand of cooperation from the side of the participants. For example, XM2VTS database contain thousands of video recorded in a studio from 295 subjects. Recording these subjects repeatedly in public places such as supermarkets, offices, streets, etc., is not realistic. To this end, we present a procedure to separate the background of a video recorded in studio conditions with the purpose to replace it with an arbitrary complex background, e.g., outdoor scene containing motion, to measure performance, e.g., eye tracking. Furthermore, we present how an affine transformation and synthetic noise can be incorporated into the production of the new database to simulate natural 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, bandwidth, and most importantly, the time of clients populating the database, and the time of the operators.

Keywords: Biometrics; XM2VTS; Damascened video; Face tracking; Face recognition; Performance quantification

1. Introduction

Biometrics is an increasingly important and 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, it has become an integral part of the infrastructure used for diverse business sectors such as security, finance, health, law enforcement, etc. (Bailly-Baillire et al., 2003; Jain et al., 2004b). 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., 2004a). 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, by using multiple modalities for authentication, and incorporating aliveness detection into the systems. Therefore,

* Corresponding author. Tel.: +46 (0)35 167421; fax: +46 (0)35 148533.
E-mail addresses: Dereje.Teferi@ide.hh.se (D. Teferi), Josef.Bigun@ide.hh.se (J. Bigun).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.06.007

performance of biometric systems in practice has to be measured by using databases simulating realistic environments that systems will face at the time of implementation.

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 will perform evidently better. However, most biometric databases are built in a studio and have near-constant color background such as XM2VTS (blue), CUAVE (green) (Patterson et al., 2002), etc. Though reasonable effort is made to ensure perfectly constant background, this is not the case in many valuable DB recordings, e.g., in XM2VTS, the background color is not within a constant range that can be found with standard technique such as chroma keying, Fig. 1. The reasons for this include non-uniform illumination, and the freedom of the DB-clients to choose dress-color.[1] While building small databases, with few number of clients and few recording sessions, it could be possible to ask DB-Clients to avoid certain colors from their dressing, and apply constant illumination to the background for later separation. It may even be possible to repeat recordings of those clients where separation of the background goes wrong. However, this is difficult for large databases with hundreds of clients and thousands of recordings such as that of XM2VTS.

The need to collect new databases, consuming huge resources and claiming important maintenance resources, to represent real world situations is stated in (Bailly-Baillire 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.*

However, recording a brand new database in real world (non-controlled) backgrounds capturing the multiple scenarios for different applications is difficult, if not impossible, because of the combinatorial explosion. This is due to the high demand the DB-recording puts on the participants to repeatedly appear on intervals to real-life environments (e.g., malls, busy stations, streets, etc.) for recording.

Therefore, building damascened video databases by synthesizing already existing ones, or new studio-recorded databases, with realistic and representative background videos is a viable complement to new DB-recordings to improve performance quantification of research in biometrics. 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 not only to generate various damascened XM2VTS databases with realistic backgrounds, such as shopping centers, moving cars, busy stations, people, additionally, they will be used to add scale variation, noise, motion blur, etc., to test the sensitivity of face tracking and recognition algorithms.

This paper is organized as follows. The next two sections provide information on related works and give some background on the XM2VTS database. Section 4 discusses standard clustering algorithms and the specific demand of adaptive image segmentation for XM2VTS. In section 5 we present the compression technique we implemented to reduce the size of the masks for ease of distribution. Section 6 and its subsections come with our approach to build the damascened XM2VTS database with added variability. The subsections include the different types of damascening techniques applied such as scaling, blur, rotation, etc. Section 7 gives some details on how to access the DXM2VTS database. In Section 8 we present the experiment conducted and finally our conclusions are summarized in Section 9.

2. Related work

An image degradation for the XM2VTS static image database is developed as part of COST 275 project Fratric, 2003. Various degradations methods are applied on the static images of the XM2VTS database. In this degradation process the background of the images is separated by assuming that the blue component of the color space is greater, by at least 25%, than the green and the red components and the lighting is between 30 and 200 (out of 255). This work is an important step although the new degraded database is for the static images only which constitute less than 0.5%, assuming an average of 200 frames per video, of the total size of the XM2VTS database.

In our work, binary masks are created for all videos of the XM2VTS database. Hence, segmentation is no more necessary and the DXM2VTS database can be developed by simply collecting a new background video and by adding various degradations parameters as necessary for the biometric system in question. Moreover, This work can be extended to other databases with different background color than the blue, for example to CUAVE database which has a green background. We have also implemented a method using motion of pixels between successive frames and region of interest (*narrow band*) to improve segmentation efficiency of head-shoulder image sequences.

3. 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

[1] In TV-weather-program like recordings, there are few people who are involved and they can be instructed to avoid certain colors in their appearance.

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, lip dynamics, 3D face surface modeling, speech/lip surface correlation, and speech/lip signal synchronization Messer et al. (1999).

The video recorded is of the type that can be acquired during a normal access claim intercourse between an access point and a client. The recording includes clients speaking some sentences and digits between zero and nine assuming the biometric system may request some specific information from the client. Moreover, the clients are recorded making extreme head rotation to extract head side profile of the subjects.

The subjects of the database were selected from both genders with different age group to enhance development of robust algorithms.

3.1. Performance quantification

The performance of a biometric verification system is generally characterized by two error statistics: false-rejection (FR) and false-acceptance (FA) rates. A false rejection occurs when a system rejects an authorized client; whereas a false acceptance occurs when a system incorrectly accepts an impostor as a client Phillips et al. (2000). An ideal biometric system would be one with zero false-rejection and false acceptance rates. However, biometric systems are not perfect.

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-Baillire 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 (ICPR-2000) and the 2003 conference on Audio and Video Based Person Authentication (AVBPA-2003) are reported in (Matas et al., 2000; Messer et al., 2003), respectively.

However, XM2VTS and many other face image and video databases are designed mainly to measure performance of face recognition methods and thus the images contain only one face. Images with near-constant background containing one-face each do not, for example, require the system to separate and track the faces from complex backgrounds. They are suitable for training recognition systems rather than testing them 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 by challenging them with image sequences containing real-world backgrounds and simulations of effects such as noise and blur.

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

4. 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 and 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 algorithms not only specific for the used features but often, also specific to the applications, to perform reasonable image segmentation. A survey of several image segmentation techniques are discussed in (Cheng et al., 2001; 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, k-means, 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 some of the standard image segmentation techniques and discuss 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.

4.1. Standard algorithms

There are several general purpose segmentation algorithms that have been developed in the past. Some of them are included in common mathematical packages such as Matlab and Mathematica and they are also used as modules in image processing. Two algorithms: chroma keying and fuzzy c-means are discussed below.

4.1.1. Chroma keying

Chroma keying is a technique used to replace a certain color (or a small range of color) in an image with another image part. It is usually applied to superimpose one video image onto another, e.g., weather forecast broadcasting or creating special effects on a movie. A blue of green color is typically used for chroma keying.

Chroma keying is commonly implemented in the HSV (hue, saturation, value) color space. If the image is in RGB (red green blue) color space then it can be converted to the HSV color space as follows.

Let *Max* and *Min* be the Maximum and the Minimum of the RGB values of a pixel respectively, then

$$H = \begin{cases} 0, & \text{if Max} = \text{Min} \\ 60 \times \frac{G-B}{\text{Max}-\text{Min}} + 0, & \text{if Max} = R \text{ and } G \geqslant B \\ 60 \times \frac{G-B}{\text{Max}-\text{Min}} + 360, & \text{if Max} = R \text{ and } G < B \\ 60 \times \frac{B-R}{\text{Max}-\text{Min}} + 120, & \text{if Max} = G \\ 60 \times \frac{R-G}{\text{Max}-\text{Min}} + 240, & \text{if Max} = B \end{cases}$$

$$S = \begin{cases} 0, & \text{if Max} = 0 \\ 1 - \frac{\text{Min}}{\text{Max}}, & \text{otherwise} \end{cases}$$

$$V = \text{Max} \tag{1}$$

where $H \in [0, 360)$ and S, V, R, G, $B \in [0, 1]$.

Then, the image is replaced with a new background for those parts/pixels of the HSV image where the *Hue* value is within a predefined angle. Fig. 1 illustrate some results of the chroma keying technique.

The problem with chroma keying, in the case of the XM2VTS database, is that the background illumination is not constant and the clients were not instructed to wear clothes other than blue resulting in incorrect segmentation, Fig. 1.

4.1.2. FCM – Fuzzy c-means

Fuzzy c-means is a clustering technique where each feature vector x_i (pixel in our image) is a member of all the clusters only to differ by the weight of membership $u_{ij} \in [0, 1]$. In other words, u_{ij} can be seen as the distance from a feature vector (pixel) x_i to the center of cluster j normalized by the sum of distances to all class centers. Therefore, the sum of the membership values of a pixel is 1 Bigun (2006). That is:

$$\sum_{i=1}^{c} u_{ij} = 1, \tag{2}$$

where, $j = 1, \ldots, N$, N is the number of feature vectors and $c =$ number of classes.

Here, we assume that $1 < c < N - 1$ because when $c = 1$ and $c = N$ then we get trivial partitioning of either all vectors in one class or each feature vector in its own separate class respectively.

FCM is an iterative algorithm which aims to find cluster centers that minimize the dissimilarity function shown in Eq. (3).

(a) Original images (b) Chroma keyed images

Fig. 1. Segmentation by chroma keying. (a) Original images and (b) chroma keyed images.

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{c} u_{ij}^m \|x_i - c_j\|^2, \tag{3}$$

where $m \in [1,\infty)$, the weighting exponent controlling the level of fuzziness, u_{ij}^m is the degree of membership of a pixel x_i to cluster j, $\|x_i - c_j\|$ is the euclidean distance between the ith pixel and the jth cluster center.

The clustering is performed by minimizing Eq. (3) above while updating u_{ij} and c_j as follows:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}, \tag{4}$$

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}. \tag{5}$$

The general procedure to implement the FCM algorithm can be stated as:

(1) **Initialize:** Fix the number of classes and the number of iterations k, and initialize the membership matrix U^0.
(2) **Calculate cluster centers:** at iteration k update cluster center $C^k = [c_j]$ and $U^{(k)}$ according to Eq. (5).
(3) **Update partitions:** Update $U^{(k)}$ and $U^{(k+1)}$ according to Eq. (4).
(4) **Terminate?** Stop processing if $\|U^{(k+1)} - U^{(k)}\| \leqslant \epsilon$ or repeat process from step 2.

Fig. 2 illustrates segmentation results of three images from the XM2VTS database by using FCM algorithm with 4 classes. FCM is an efficient algorithm for clustering images by adjusting the number of classes as necessary. However, using FCM with a constant class size to segment thousands of images, as in the case of the XM2VTS database, will not be an appropriate choice. Hence, the need for a robust segmentation algorithm that can not only adjust the number of classes automatically but also that can be easily modified to work for segmentation of image sequences as discussed below.

4.2. Our segmentation approach

Here we present the technique we used to separate a near-constant color background from a face-shoulder image sequence. We applied it on the XM2VTS database to create the necessary image-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. The refinement process iteratively adjusts the class assignments and their centroid. Once the first frame is segmented, motion of pixels between frames and region of interest are used to improve segmentation of the remaining frames. These procedures are presented below.

4.2.1. 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. It is computed as:

$$h_g(x,y) = \frac{1}{2\pi\sigma}\exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right),$$
$$G(x,y) = \frac{h_g(x,y)}{\sum_x\sum_y h_g}. \tag{6}$$

The smoothing is done by convolving the image I with the filter G above

$$R = I \otimes G = \iint I(u,v)G(x-u,y-v)\mathrm{d}u\mathrm{d}v, \tag{7}$$

where, u and v are indices of the image I and x and y are indices of the Gaussian G.

The segmentation algorithm is applied on the smoothed image. The proposed segmentation does not require prior

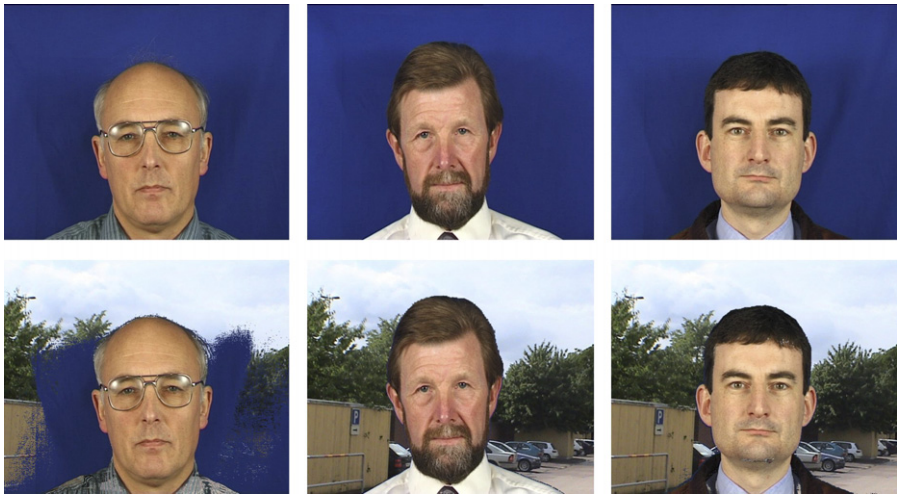


Fig. 2. Segmentation by FCM with 4 classes.

knowledge of the number of clusters in the image as this is automatically determined by the algorithm. Automatic determination of class size is necessary in this procedure 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.

4.2.2. The feature space and the clustering criteria

The segmentation process needs a *feature space*, which is normally a vector space, equipped with a *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 $\boldsymbol{f}_{ij}$ represent the color vector of a pixel at row i, column j of an image then

$$\boldsymbol{f}_{ij} = (f_{ij}^1, f_{ij}^2, f_{ij}^3)^{\mathrm{T}} \tag{8}$$

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

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

$$\|\boldsymbol{f}\|_\infty = \|(f^1, f^2, f^3)^{\mathrm{T}}\|_\infty = \max_k\{f^k\}, \tag{9}$$

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

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

$$\begin{aligned}\tilde{\boldsymbol{f}}_{ij} &= \left(\tilde{f}_{ij}^1, \tilde{f}_{ij}^2, \tilde{f}_{ij}^3\right)^{\mathrm{T}} \\ &= \left(f_{ij}^1 - f_{ij}^2, f_{ij}^2 - f_{ij}^3, f_{ij}^3 - f_{ij}^1\right)^{\mathrm{T}}.\end{aligned} \tag{10}$$

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

$$\boldsymbol{\delta}_{ij}^l = \left(\left\|\boldsymbol{f}_{ij} - \mathbf{c}^l\right\|_\infty, \quad \left\|\tilde{\boldsymbol{f}}_{ij} - \tilde{\mathbf{c}}^l\right\|_\infty\right)^{\mathrm{T}}, \tag{11}$$

where $\mathbf{c}^l$ is the cluster center vector of the cluster with label l and $\boldsymbol{f}_{ij}$ represents the rgb-color vector of a pixel. The vectors $\tilde{\mathbf{c}}^l, \tilde{\boldsymbol{f}}_{ij}$ 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. 3. 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,

$$0 \leqslant \boldsymbol{\delta}_{ij}^l \leqslant \boldsymbol{\tau} \tag{12}$$

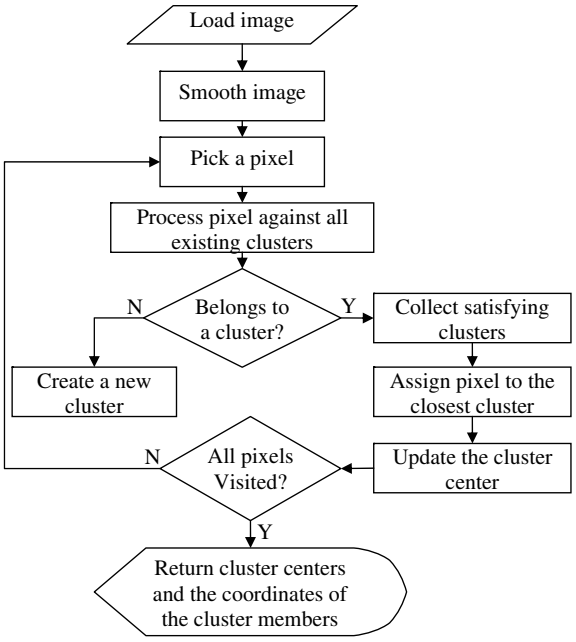


Fig. 3. The segmentation algorithm.

where $\boldsymbol{\tau} = (\tau^1, \tau^2)^{\mathrm{T}}$ 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 (scan-line, random, and hilbert curve) scan directions, Fig. 4. 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 scan-line 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

$$\left\|\boldsymbol{\delta}_{ij}^l\right\|_\infty = \left\|\left(\|\boldsymbol{f}_{ij} - \boldsymbol{c}^l\|_\infty, \quad \|\tilde{\boldsymbol{f}}_{ij} - \tilde{\boldsymbol{c}}^l\|_\infty\right)^{\mathrm{T}}\right\|_\infty. \tag{13}$$

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.

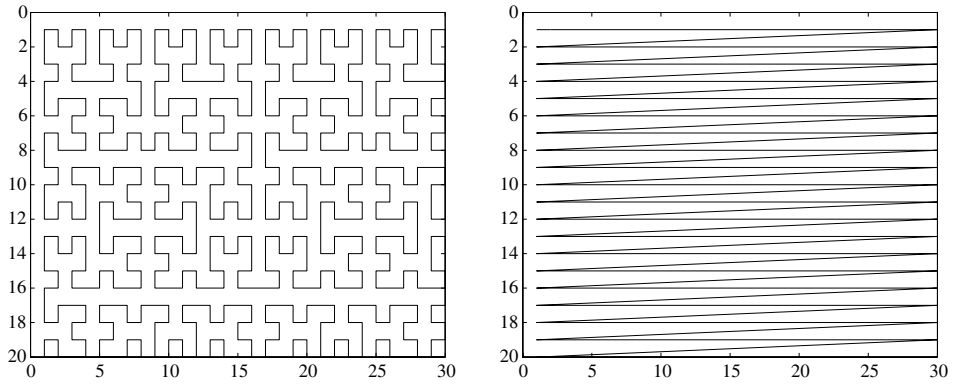


Fig. 4. Visiting orders of hilbert and scan-line traversals for a 20×30 image.

Table 1
Cluster matrix

Cl. No	Red	Green	Blue	No. of pixels	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
⋮	⋮	⋮	⋮	⋮	⋮

4.2.3. Cluster refinement

Cluster refinement is a procedure applied on the image to refine the cluster centers as well as their member pixels. This procedure is necessary as the segmentation above terminates upon visiting each pixel only 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. Moreover, the cluster matrix that we obtain from the first iteration of the first frame is used as a starting cluster list for the rest of the frames within the image sequence as there is minimal variation on the result from the first iteration of the frames. The segmentation result converges by running this refinement procedure repeatedly while automatically adjusting the class centroids and member pixels as well as the number of classes over each iteration.

Once the clusters are refined, the background class is identified as 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.

4.2.4. 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 and the change in pixel value between consecutive frames (motion of pixels) to improve the efficiency of our segmentation algorithm for 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.

4.2.4.1. Motion. Consecutive frames in an image sequence contain a lot of redundancy. This redundancy is often exploited for the purpose of video compression. Here, we use it for the purpose of improving the segmentation efficiency of the image sequences.

A complete segmentation procedure is performed only on the first image. Segmentation for the rest of the frames is performed only for those points where there is motion, see Fig. 5.

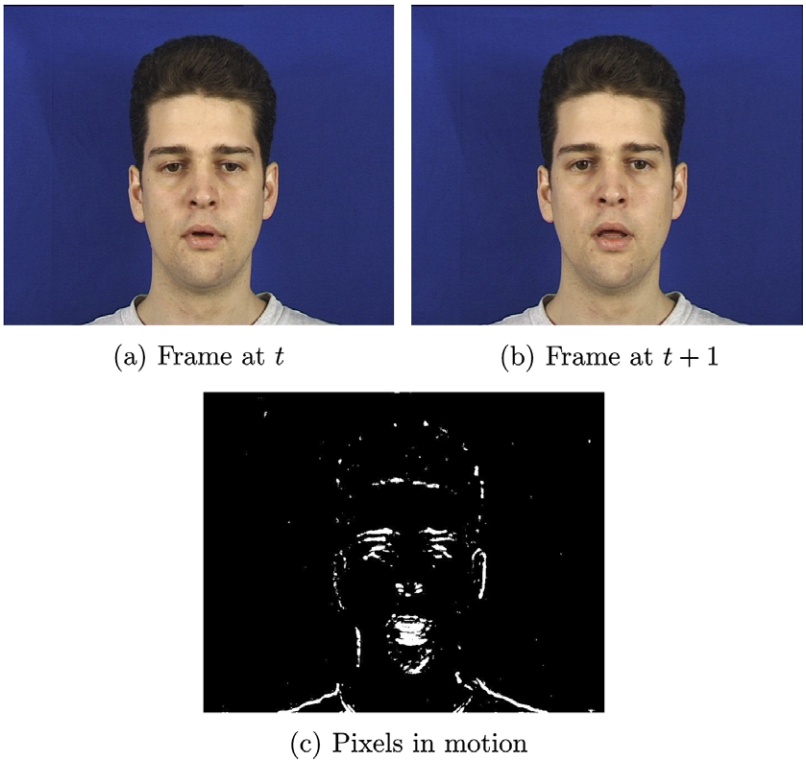
The change in consecutive frames is calculated by using absolute difference between the two frames:

$$MAD = |\boldsymbol{F}^{m+1}(k,l) - \boldsymbol{F}^{m}(k,l)|, \tag{14}$$

where $\boldsymbol{F}^{m}$ and $\boldsymbol{F}^{m+1}$ are frames at time t and $t+1$.

The segmentation efficiency can be further improved by taking only those points in motion which lie around the boundary of the head. Since the motion within the face, such as that of the mouth and the eyes, are not in our interest. This can be done by creating a *narrow band* around the head shoulder of the binary mask of the first frame.

4.2.4.2. Narrow band. 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 those pixels in motion, Fig. 5, and corresponding to the pixel coordinates of the narrow band, from the binary mask of the first frame, are passed through the segmentation scheme presented above.



(a) Frame at t (b) Frame at $t+1$

(c) Pixels in motion

Fig. 5. Pixels in motion between consecutive frames.

This method is a supplement to the improvement made by using motion of pixels and it works only for the speaking head shots of the XM2VTS database. This is because the motion of the subject is only within the marked narrow band, Fig. 6. This method does not improve efficiency for the rotating head image sequences as the motion of the subjects cover almost all areas of the frame.

We know, from Fig. 6b, 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 *moving pixels* within the narrow band. Then, the part that belongs to the background is set to one and the foreground to zero. This result is subsequently used as a binary mask for the specific frame in the image sequence.

5. 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 performance of two of these algorithms, MPEG-4 and RLE – Run Length Encoding, are presented below.

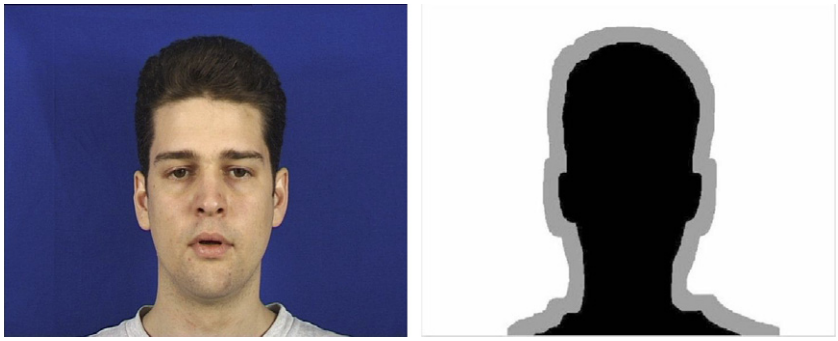


Fig. 6. Narrow band around the foreground.

Table 2
RLE vector

First symbol	Run-count of first symbol	Run-count of next symbol	⋮	⋮	⋮	⋮	End-of-frame delimiter	⋮	⋮
1	37234	35	712	.	.	.	0	37049	⋮

5.1. MPEG-4 compression

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 2000The MPEG home page.

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/s is only 10% whereas at a data rate of 256 kb/s 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.

5.2. RLE compression

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 along with their run count, 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.

The size of the RLE vector is 70–80% less than the size of the original binary mask. Moreover, the compression is lossless, the results are 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.

6. Building the damascened database with added variability

The next step is building the damascened video from the compressed mask. First, the mask is decompressed using a reverse-algorithm of the one that is used for the compression. This generates the binary mask for each frame in the image sequence. Then, the damascening process is performed on each image.

Parallel frames are extracted from the original video (XM2VTS in this case), its corresponding mask, and the new background video, one at a time to build the damascened video. 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 original frame and adding the result to get the required frame of the damascened image sequence. At this point, variability is added as required on the damascened frame to simulate real recording environments, Fig. 7.

In reality, video 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 add variability to the damascened video according to the following models.

6.1. Changing background

The decompressed binary mask is used to change the background of the video with realistic scenery. The background of every frame from the video database can be replaced by either a static image or frames from a video by simple multiplication with its corresponding binary mask.

6.2. Smoothing blur

Each frame from the image sequence can be blurred by using either an averaging filter Eq. (15) or a Gaussian filter Eq. (16). The averaging filter gives equal weight to all the pixels when convolving. Whereas, the Gaussian filter gives more weight to the central pixel depending on the variance σ when smoothing, Fig. 8.

The averaging filter is an $n \times n$ matrix with elements:

$$h(x,y) = \frac{1}{n^2} \quad x = 1,\ldots,n \quad \text{and} \quad y = 1,\ldots,n. \tag{15}$$

Whereas, the Gaussian filter is an $n \times n$ matrix with elements:

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

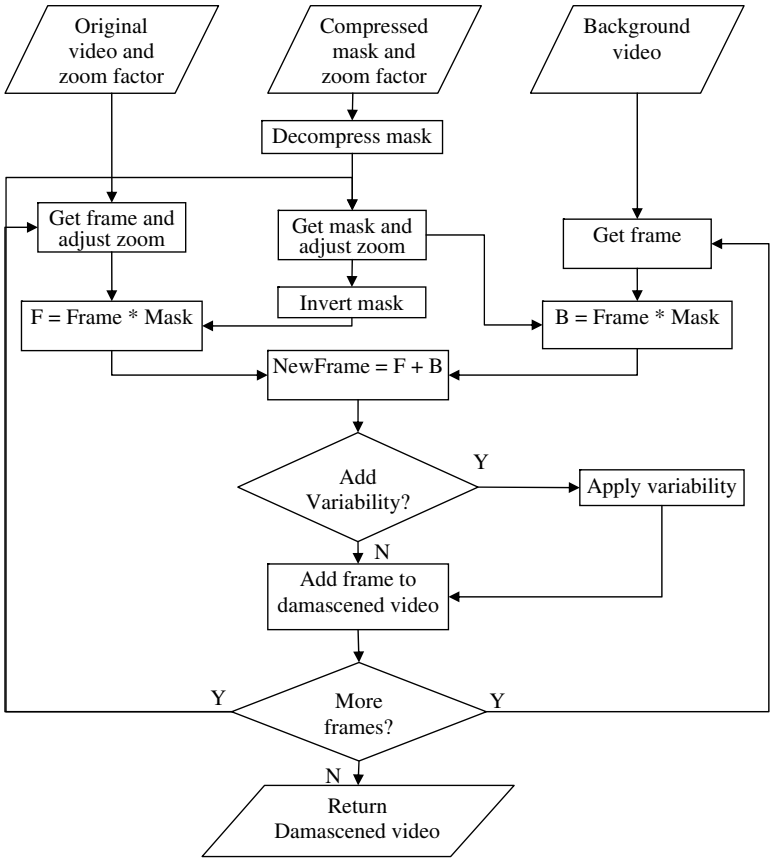
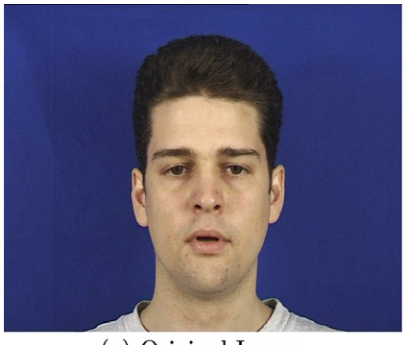


Fig. 7. Algorithm for damascening a video.



(a) Original Image

(b) Averaging blur

(c) Gaussian blur

Fig. 8. Smoothing blur.

$$h_g(x,y) = \frac{1}{2\pi\sigma}\exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right),$$
$$h(x,y) = \frac{h_g(x,y)}{\sum_1^n\sum_1^n h_g}. \quad (16)$$

The image is convolved with the filter matrix as follows:

$$R = I \otimes h = \iint I(u,v)h(x-u,y-v)\mathrm{d}u\mathrm{d}v. \quad (17)$$

6.3. Motion blur

Motion blurs are used to simulate the blur created by fast-moving objects in a certain direction. It is created by convolving an image I with a motion blur filter. The filter translates the pixels by n pixels in the direction of an angle θ. by adjusting n and θ one can simulate motion for linear movements with an angle of θ degrees in a counterclockwise direction, see, Fig. 9.

6.4. Imaging noise

Digital images are prone to different types of noise. Depending on how the image is created, several types of noise can be introduced into the image. For example:

- If the image is scanned from a photograph, the scanner itself or low quality of the photograph can create noise.
- If the image is acquired directly in a digital format, the quality of the camera can introduce noise.
- Electronic transmission of image data can introduce noise.

To simulate some of them, we propose to add two types of noise, Gaussian and Salt and Pepper, as necessary with different density and intensity levels, Fig. 10.

6.5. Foreground scaling

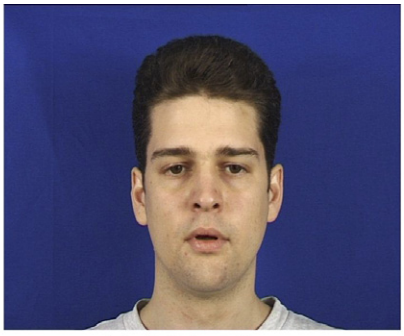
Sometimes, in real life, faces may not appear by themselves but in a crowd. In the XM2VTS database the face covers too much of the image space to add other faces. The foreground image (head shoulder) can be scaled down so that other faces can be added at the background as illustrated in Fig. 11.

6.6. Affine transformation

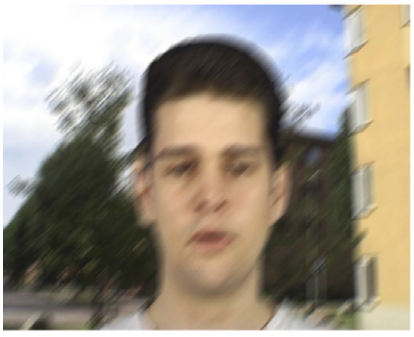
Affine transformation is a linear transformation and a translation from one vector space to another. That is, let V and W be vector spaces over the same field F. An affine transformation is a mapping A: $V \to W$ such that (see Fig. 12):

$$A(v) = L(v) + w \quad \text{where } v \in V. \quad (18)$$

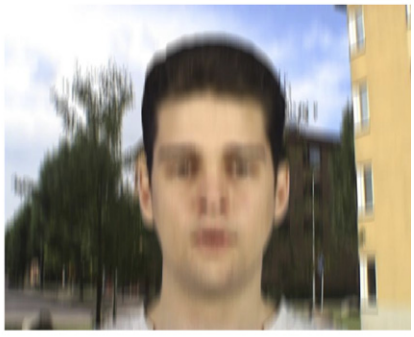
A transformation matrix, Eq. (19), can be used to transform point (x,y) in an image to (x',y') and rotate it by θ:



(a) Original Image

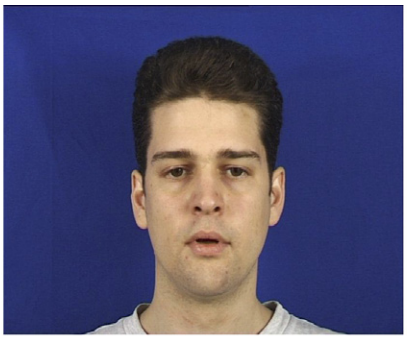


(b) Motion blur $n = 30$ and $\theta = 45$



(c) Motion blur $n = 30$ and $\theta = 90$

Fig. 9. Effects of motion blur.

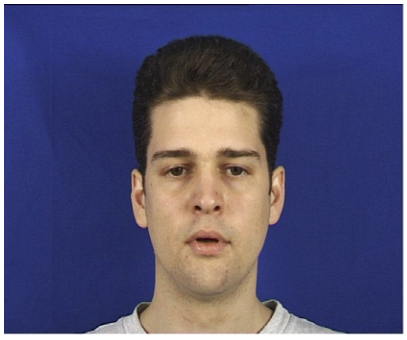


(a) Original Image

(b) Gaussian Noise

(c) Salt and pepper noise

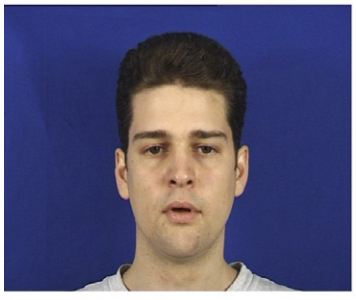
Fig. 10. Gaussian and salt and pepper noise.



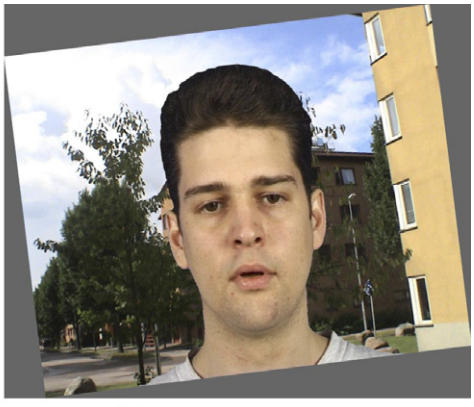
(a) Original Image

(b) Foreground scaled down by 50%

Fig. 11. Foreground scaling.



(a) Original Image



(b) Scaled and rotated

Fig. 12. Affine transformation: scaling and rotation.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & h \\ -\sin(\theta) & \cos(\theta) & k \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (19)$$

representing the transformation:

$$\begin{aligned} x' &= x \cdot \cos(\theta) + y \cdot \sin(\theta) + h \quad \text{and} \\ y' &= -x \cdot \sin(\theta) + y \cdot \cos(\theta) + k \end{aligned} \quad (20)$$

7. Accessibility

A set of standard video backgrounds such as offices, outdoors, malls, moving cars, etc., are collected and offered with the compressed mask of the XM2VTS database. Moreover, a software is provided for the damascening process. These are available for download from Halmstad University.[3] These download does not include the original XM2VTS database. The licence for the XM2VTS database can be requested from the University of Surrey.[4]

The use of comparable quality background video with equal frame rate is suggested to make the damascened video look more natural. The mask, the background, and the original video data are used to generate the damascened video database with required level of noise and variability to simulate real life scenario changes.

8. Experimental results

A series of experiments are conducted on selected XM2VTS database sequences to determine the optimal threshold for distance to cluster centers and hue variations, $\tau = (\tau_1, \tau_2)^{\mathrm{T}}$ 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. Moreover, the Gaussian filter used to smooth the image frames before segmentation is set to be of size (7×7) and $\sigma = 2.5$.

Using these parameters, 94% of the image sequences of the XM2VTS database are segmented successfully. The remaining 6% of the image sequences are segmented semi-automatically by human intervention only on the first frame of the image sequence. Segmentation of consecutive frames is done automatically by adjusting the pixel values of the image based on the adjustments made on the first frame and by using motion of pixels and narrow band suggested in Section 4.2.4.

A software to generate damascened XM2VTS database from the binary masks is developed. A set of representative degradation parameters such as salt and pepper noise, motion and averaging blur, foreground scaling and affine transformation are included in the damascening system to simulate motion of camera, rotation, blurring etc. in the output video.

9. 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 audio-visual) database in existence, at least in academic research. A method is presented in this paper to store a binary mask for all the 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.

Duplicating the exact lighting environment of the studio recorded XM2VTS database while recording a new outdoor background video may not be possible. However, as this is the first work in damascening image sequences for improving performance evaluation of face tracking and recognition systems, we believe DXM2VTS can be a viable complement to XM2VTS in extending its usefulness in biometric research.

Acknowledgement

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

References

Bailly-Baillire, E., Bengio, S., Bimbot, F., Hamouz, M., Kittler, J., Mariethoz, J., Matas, J., Messor, K., Popovici, V., Poree, F., Ruiz, B., Thiran, J., 2003. The BANCA Database and Evaluation Protocol. In: Proc. of Audio and Video Based Person Authentication: AVBPA 2003, LNCS 2688, pp. 625–638.

Bigun, J., 2006. Vision with direction: A systematic introduction to image processing and computer vision. Springer-Verlag, Berlin, Heidelberg.

Cheng, H., Jiang, X.H., Wang, J., 2001. Colour image segmentation: Advances and prospects. Pattern Recogn. Lett. 34, 1277–1294.

Fratric, I., 2003. Degradation of the XM2VTS Face Images Database, STSM report, COST Action 275. Biometric-Based Recognition Over the Internet.

Jain, A., Ross, A., Prabhakar, S., 2004a. An Introduction to Biometric Recognition. Proc. IEEE Trans. Circuits and Systems for Video Technol., Special Issue on Image and Video Based Biometrics, 14 (1).

Jain, A., Pankanti, S., Prabhakar, S., Hong, L., Ross, A., 2004b. Biometrics: A Grand Challenge. In: Proc. of the 17th Int. Conf. on Pattern Recognition, ICPR 2004, 2, pp. 935–942.

[3] The damascening software, some background videos and the compressed binary masks can be downloaded from http://www2.hh.se/staff/josef/downloads.

[4] The XM2VTS Database website http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/.

Lucchese, L., Mitra, S.K., 2001. Colour Image Segmentation: A State-of-the-art Survey. In: Proc. of Image Processing, Vision, and Pattern Recognition. Proc. of the Indian National Science Academy (INSA-A), 67A(2), pp. 207–221.

Matas, J., Hamouz, M., Jonsson, K., Kittler, J., Li, Y., Kotropoulos, C., Tefas, A., Patas, I., Tan, T., Yan, H., Smeraldi, F., Bigun, J., Capdevielle, N., Gerstner, W., Ben-Yacoub, S., Abdeljaoued, Y., Mayoraz, E., 2000. Comparison of Face Verification Results on the XM2VTS Database. Proc. Int. Conf. Pattern Recogn. ICPR-15. IEEE Computer Society.

Messer, K., Matas, J., Kitler, J., Luettin, J., Maitre, G., 1999. XM2VTSDB: The Extended M2VTS Database. In: Proc. 2nd Int. Conf. on Audio and Video-based Biometric Person Authentication (AVBPA'99), pp. 72–77.

Messer, K., Kittler, J., Sadeghi, M., Marcel, S., Marcel, C., Bengio, S., Cardinaux, F., Sanderson, C., Czyz, .J., Vandendorpe, L., Srisuk, S., Petrou, M., Kurutach, W., Kadyrov, A., Paredes, R., Kepenekci, B., Tek, F., Akar, G., Deravi, F., Mavity, N., 2003. Face Verification Competition on the XM2VTS Database. In: Proc. of the Fourth Int. Conf. on Audio and Video Based Biometric Person Authentication (AVBPA'03), pp. 64–974.

Ortega-Garcia, J., Bigun, J., Reynolds, D., Gonzalez-Rodriguez, J., 2004. Authentication Gets Personal with Biometrics. IEEE Signal Process. Mag. 21 (2), 50–62.

Patterson, E., Gurbuz, S., Tufekci, Z., Gowdy, J., 2002. CUAVE: A New Audio-Visual Database for Multimodal Human-Computer Interface Research. In: Proc. Int. Conf. on Acoustics, Speech, and Signal Process. (ICASSP 2002), pp. 2017–2020.

Phillips, P., Martin, A., Wilson, C., Przybocki, M., 2000. An Introduction to Evaluating Biometric Systems. IEEE Comput. 33 (2), 56–63.

The MPEG home page., <http://www.chiariglione.org/mpeg/>.

Yang, M.-H., Kriegman, D., Ahuja, N., 2002. Detecting Faces in Images: A Survey. IEEE Trans. Pattern Anal. Machine Intell. 24 (1), 34–58.