Semantic video fingerprinting and retrieval using face information

Costas Cotsaces *, Nikos Nikolaidis and Ioannis Pitas

Department of Informatics

University of Thessaloniki

Thessaloniki 54124, Greece

tel:+302310996361

Abstract

The management of large video databases, especially those containing motion picture and television data, is a major contemporary challenge. A very significant tool for this management is the ability to *retrieve* those segments that are perceptually similar to a query segment. Another similar but equally important task is determining if a query segment is a (possibly modified) copy of part of a video in the database. The basic way to perform these two tasks is to characterize each video segment with a unique representation called a *signature*. Using semantic information for the construction of the signatures is a good way to ensure robustness in retrieval and fingerprinting. Here a ubiquitous semantic feature, namely the existence and identity of human faces, will be used to construct the signature. A fast algorithm has been developed to quickly and robustly perform these two tasks on very large video databases. The prerequisite face recognition was performed by a commercial system. Having verified the basic efficacy of our algorithm on a database of real video from motion pictures and television series, we then proceed to further explore its performance in an artificial digital video database, which was created using a

probabilistic model of the video creation process. This enabled us to explore variations in performance based on parameters that were impossible to control in a real video database. Furthermore, the suitability of the proposed approach for very large databases was tested using (artificial) data corresponding to hundreds or thousands of hours of video.

Key words: video fingerprinting, video retrieval, face recognition, semantic features

---

1 Introduction

Advances in digital video technology in the last decade, particularly in synergy with computer technology, have resulted in an explosion in the amount of available digital video. This is often in the form of large *video databases* that contain hundreds or thousands of hours of digital video. Digital video has opened up the potential of using video sources in ways other than the traditional serial playback. However, this requires the development of new technologies for accessing and manipulating digital video [1]. Additionally, the amount of digital video data, which has the potential of becoming much greater than that of traditional analog video, necessitates the development of digital video management tools for handling massive video databases. Finally, the ease with which digital video (like all digital media) can be flawlessly copied, makes the development of appropriate rights protection and authentication tools highly desirable.

Video *retrieval* is a fundamental technology for the management of digital

* Corresponding author.
Email address: `pitas@zeus.csd.auth.gr` (Ioannis Pitas).

video. Given a video database, the goal of video retrieval is to locate one or more video segments that the user is interested in. Video retrieval methods are comparable to those used for the retrieval of other types of multimedia objects, such as images and usually follow one of two paradigms. In the case of *query-by-keyword*, the image or video database is annotated with keywords or other metadata. This annotation can be performed manually, semi-automatically or automatically. The user then enters the keywords that best describe what he is searching for or he interacts with a user interface that produces some other appropriate metadata. These metadata are then used to perform a textual or symbolic search in the database. On the other hand, *query-by-example* means that the images or videos in the database are characterized (almost always using automatic methods) with an appropriate set of features, which constitute a reduced dimensionality representation of the digital item. We call this representation a *signature*. The user then inputs or selects an image or video similar to the one that he is searching for. Then, a set of features is extracted from the user image or video and used to find images or videos with similar features in the database, sometimes using advanced indexing techniques.

Another technology which is useful for the management of video, particularly with respect to rights protection, is *fingerprinting* [2,3], also known as *perceptual hashing* or *replica detection*. This is defined as the identification of a video segment using a representation called *fingerprint* (or sometimes *perceptual hash*), which is extracted from the video content. The fingerprint must uniquely identify a video segment, but does not necessarily need to represent its content. Additionally, it must remain the same when a video segment is manipulated, usually by common video processing operations such as resizing, cropping, histogram equalization, compression etc. Fingerprints can be used

for establishing whether two given segments are either identical or derived from each other, and also for establishing whether a video segment is identical with (or derived from) any segment within a given video database. The major difference between fingerprinting and retrieval is that the similarity criterion is usually looser in retrieval, since the user is often interested not only in copies of a video segment, but also in video segments that are perceptually similar to it. Apart from differences in their use, a fingerprint and a signature can be said to be essentially the same thing, and in the following we will use the term "signature" to refer to both.

In the case of retrieval, the user is usually interested either in the n best matches, or in those matches whose goodness is above a certain threshold, or simply in a list of matches arranged from best to worst. Alternatively, the user may be interested in only one match, the best one. In the case of fingerprinting the user is interested in finding whether a specific segment exists within a database, which is similar to the task of retrieval of only one match. However, fingerprinting algorithms differ in that they are required to be able to return an empty result set (in both the above cases), when no identical segments are identified.

The following attributes are generally desirable for a query-by-example retrieval and/or fingerprinting algorithm:

(1) *Uniqueness*, which means that two different videos should have different signatures, while two semantically equal videos should have the same signature. A broader way to express the above is to say that the distance between the signatures of two different video segments is roughly proportional to the perceptual distance between the videos themselves. In the

case of fingerprinting semantic identity between two videos is equivalent to the ability to transform one into the other through common manipulations.

(2) *Robustness* with respect to noise and content manipulation, e.g. compression. This includes, especially in the case of fingerprinting, malicious manipulation that attempts to circumvent intellectual rights protection mechanisms. One way to achieve robustness is to extract features from the video that are semantic in nature, and that are also as temporally invariant as possible.

(3) *Temporal Segmentation Invariance [4]*, which describes whether the comparison between two video segments is invariant with respect to their exact temporal limits. The algorithm should not require that a video segment starts and ends in a set of predefined times (e.g. shot boundaries). One way to achieve this is by having a signature that is not defined only for specific discrete partitions of the video, but is defined for every frame.

(4) *Indexability* [5]. It describes the ease of seeking and retrieving a specific video segment in a large database based on its signature. This is conditional on two factors: a) high granularity of the indexed quantities (to avoid serial searches as much as possible), and b) low fingerprint dimensionality (since, in general, multidimensional indexing techniques are very inefficient) [6] .

(5) *Segment Length Independence.* Ideally, an algorithm should function equally well for the retrieval of query segments of all sizes. However, larger segments contain more information than smaller ones, and thus algorithms usually produce better results with larger segments.

Various approaches have been tried in order to perform retrieval or finger-

printing on video sequences. Below we give a list of the major families of such algorithms (mostly for retrieval), along with their inherent limitations:

(1) *Color Statistics.* The most common of them is the color histogram. They are usually considered within shots [7,8,9,10], which raises problems regarding temporal segmentation invariance. It also raises consistency issues regarding shot detection. It is also difficult to achieve good indexing performance, since color is not necessarily a discriminant feature for video. Furthermore, color histograms tend to have a high dimensionality.

(2) *Keyframe Comparison.* This involves segmenting both the query video sequence and the video database into appropriate segments (e.g. shots), extracting characteristic frames (keyframes) from each segment and then treat the video retrieval problem as an image retrieval problem [11]. This approach has problems regarding the consistency of the temporal segmentation, but these are compounded by consistency problems regarding the selection of the keyframes.

(3) *Global / Object Motion.* There are two ways to use motion as a signature. One is to use general optical flow statistics for specific video segments [12,13,14,10], and the other to use motion trajectories of specific objects [15]. The first way has several problems: dependence on the temporal segmentation involved (e.g. shot boundary detection), vulnerability to compression (especially MPEG1 / MPEG2 encoding), indexing issues. The second one avoids shot dependence and compression problems. However, it is rather difficult to devise an appropriate indexing method for it. Furthermore, it is susceptible to inconsistencies arising from perturbations in object trajectories due to the instability of object detection caused by video noise. Finally, this approach obviously does not work

when there is little motion in a video segment (e.g. a dialog).

This work is based on the use of signatures utilizing the existence of faces of distinct individuals, such as actors, in order to robustly characterize a video segment. This is an obvious basis for characterization in many high-value and common video types like motion pictures and television series, although it may not be applicable in other types, e.g. sports videos, news and documentaries. We present the use of such signatures in fast (logarithmic-time) algorithms for video retrieval and fingerprinting, and investigate the effectiveness of such algorithms. We do not concern ourselves with face detection and recognition, since ample work has been performed on both subjects [16]. We also investigate the consistency and robustness of using such algorithms in retrieval and fingerprinting tasks, depending on the characteristics of the underlying face detection and recognition algorithms. Preliminary work on the viability of this method has been presented in [17].

A number of works until now [18,19,20,21] have been published on the subject of the use of face-related information for video indexing. However, they have not dealt with the organization and efficient indexing of such information — their topic was face recognition with a view to its eventual application on indexing. Thus, they actually present an excellent foundation in providing input to our work, in the form of detected and/or recognized faces. This is especially true for the works of Satoh [19] and of Eickeler et al [18], who perform identity recognition on the faces they detect. There has been some work that actually utilizes face information as a video signature, for example Chan et al. [22] who characterize video shots using face information, and Viallet and Bernier [23] who evaluate the similarity between different shots based on face information. None of the above, however, address video retrieval and fingerprinting in large

databases. It should be noted that here we do not propose a face detection and recognition method, but we investigate the effect of different parameters of the face detection and recognition process on the retrieval and fingerprinting performance of our method. The data used for this purpose are constructed by a probabilistic model describing the appearance of faces in videos. These data are modified by modelling the performance of face detection and recognition modules. Since this performance is never perfect, face detection and recognition are viewed as being equivalent to the addition of noise to the video signature. The practicality of our system has been already verified by implementing a real system and testing its retrieval performance on a database of real video. Our algorithm is robust to video noise and manipulations because it is based on semantic information, which is largely unaffected by such changes. It is also robust to changes in the boundaries of query segment and to malfunctions of the face detector and recognizer. Finally, it is well suited to large video databases, having been tested on artificial data corresponding to thousands of hours of video.

The paper is organized as follows: In Section 2, we define the video similarity metric we have based our algorithm on. The algorithm itself is described at Section 3. In Section 5 we demonstrate the practicality of our approach through experiments that have been performed on a database of real video. Section 4 is the most important part of the present work, as it explores the performance of our algorithm with respect to the performance of face detection and recognition. Conclusions are presented in the Section 6.

2 Video Similarity Metric

In the following, we first rigorously define the way we characterize a video using a signature which is derived from the existence of individual faces in the video. We consider the detection and identification of these faces as a to be a task that has either been solved or is to be solved outside the scope of this work. That is, we assume that an appropriate face detection and recognition module already exists. Subsequently, we give a measure for defining the similarity of two video segments based on their signatures.

2.1 Format of Signature

Our aim is to characterize a video $\mathbf{V}$, consisting of N consecutive frames f_n such that $\mathbf{V} = \{f_1\ f_2\ \ldots\ f_N\}$, through an appropriately constructed signature. Assume that there exist M individual persons s_m in $\mathbf{V}$, so that $\mathbf{S} = \{s_1\ s_2\ \ldots\ s_M\}$ is the set of all persons that have been imaged in the video. With no loss of generality, we can limit $\mathbf{S}$ to contain only the individuals of interest — excluding for example the extras in a motion picture. We then apply to $\mathbf{V}$ a face detector and recognizer F, whose output $G(n,m)$ is the certainty that person s_m appears in frame f_n (i.e. $G(n,m) = \text{Prob}\{s_m \text{ is imaged in } f_n\}$). We will name this certainty *recognizability.* $G(n,m)$ can either be a hard (binary) decision, i.e. $G(n,m) \in \{0,1\}$ or a soft one, in which case $G(n,m) \in [0,1]$. In order to reduce the amount of information that is required to represent $\mathbf{V}$through $G(n,m)$, for each person s_m all frame intervals $I_i^m = [a_i^m, b_i^m]$ such that $G(n,m) > 0,\ n \in [a_i^m, b_i^m]$ and $I_i^m \not\subset I_j^m, \forall i \neq j$ are found. I_i^m then define a *face occurrence* $F_i^m = \overline{G(n,m)}\big|_{n=a_i^m}^{b_i^m}$ which is the average rec-

ognizability of a specific person within the interval I_i^m. $G(n,m)$ can thus be approximated by a function $F(n,m)$ which contains the unit step function $u(n)$ and $[a_i^m, b_i^m]$ which is the i-th interval that contains the face of the m-th person:

$$F(n,m) = \sum_i F_i^m \left[u(n - a_i^m) - u(n - b_i^m)\right] \tag{1}$$

Each signature triplet $(F_i^m, a_i^m, b_i^m),\ i = 1, \ldots, N$ corresponding to a person S_m is a pulse in the video time domain, as shown in Figure 1. Thus the signature for a single person is a pulse series, and the complete signature is a superposition of M pulse series, as shown in Figure 2(a).

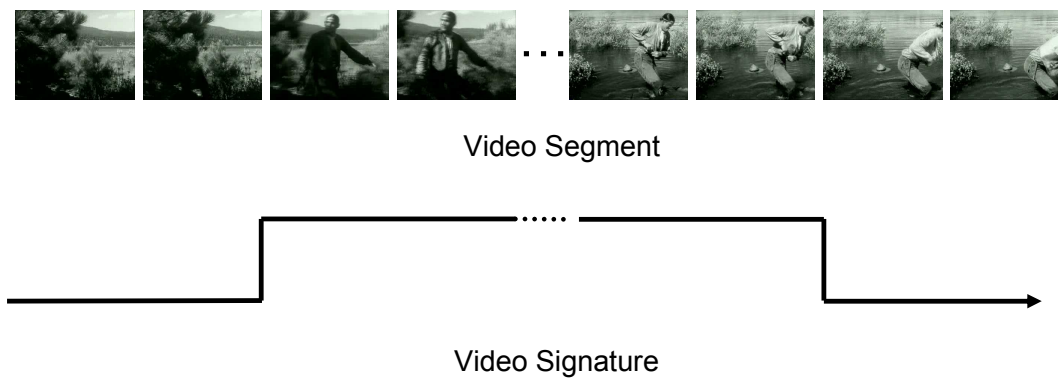

Fig. 1. Example of the relation between the appearance of a person's face in a scene and the corresponding signature quartet.

Therefore, the video $\mathbf{V}$ is characterized by a *signature* consisting of quartets of values $(s_m, F_i^m, a_i^m, b_i^m), m = 1, \ldots, M,\ i = 1, \ldots, N$. A unique face appearance, i.e. the information that person s_m has been detected from frame a_i^m to frame b_i^m with a confidence of F_i^m is represented by a quartet. A video contains $\sum_{m=1}^{M} g_m \ll N \times M$ quartets, where g_m is the number of appearances of person s_m in the video, and N and M are the total numbers of frames and persons in the video. In practice, in order to reduce the amount of redundant data in the signature, it is better to discard face occurrences that are too short

and to unify proximate occurrences of the same face.

2.2 Signature Similarity

Our objective is to compare two video segments $\mathbf{V}_1$ and $\mathbf{V}_2$. Assuming a common set of faces $\mathbf{S}$, Equation (1) can be used to characterize $\mathbf{V}_1$ and $\mathbf{V}_2$ with their respective $F_1(n,m)$ and $F_2(n,m)$. Since we do not know in advance the temporal alignment of the two videos, F_2 is moved by an arbitrary displacement d with respect to F_1. We will define as *co-occurence* C the evidence that the two videos are the same, based on their signatures. In the case of a binary decision recognizer such evidence exists if and only if a specific person m exists at a specific frame n in both signatures, i.e. $C_{hard}(d,n,m) = F_1(n,m) \cdot F_2(n+d,m)$. If the detector produces a measure of recognizability, the evidence that a specific person occurs in both signatures depends on this recognizability. The evidence of co-existence is only as good as the worst recognizability of the two signatures, and thus $C(d,n,m) = \min(F_1(n,m), F_2(n+d,m))$. The overall evidence of similarity of $F_1(n,m)$ and $F_2(n,m)$ for a specific displacement can be computed by summing $C_{hard}(d,n,m)$ over all frames and persons. In order to achieve invariance with respect to the lengths of the two video segments (respectively N_1 and N_2, assuming $N_1 \leq N_2$), C can be regularized by dividing by N_1 and the number of possible persons, M. In the case of a hard detector (whose output is 0 or 1) this corresponds to:

$$C_{hard}(d) = \sum_{n=1}^{N_1} \sum_{m=1}^{M} \frac{F_1(n,m) \cdot F_2(n+d,m)}{N_1 M} \tag{2}$$

which is the correlation between the binary signals F_1 and F_2. In the case of a detector that produces detection certainties, we have:

$$C_{soft}(d) = \sum_{n=1}^{N_1} \sum_{m=1}^{M} \frac{\min(F_1(n,m), F_2(n+d,m))}{N_1 M} \tag{3}$$

The formulation of C can be explained as a representation of the overlap between the rectangles that correspond to the quartets which refer to the same person in the two signatures. The similarity of the two signatures is defined as the maximum value of co-occurence $C_{max} = \max_d C(d)$, obtained when sliding one signature in relation to the other. The computation of $C(d)$ and C_{max} is similar to the computation of the convolution between the two face signature signals. This has the effect that small changes in the signature, such as splits, shifts, changes in height or in width of the quartet rectangles do not affect C_{max}. Having established a method for computing the similarity between two signature segments, searching for a specific video in a database entails simply comparing a candidate segment with the whole database and declaring a match when the similarity exceeds a certain threshold. However doing this exhaustively is computationally extremely expensive. Thus we have developed an algorithm that does this in near-logarithmic time with respect to the size of the database. Finally

3 Retrieval-Fingerprinting Algorithm

In the following we will give a complete description of our algorithm, which is schematically illustrated in Figure 2. Its basic ideas are the following:

(1) Indexing of the quartet database to enable realistically quick access.

(2) Search based on the most salient faces in the video segment (Figure 2(a)).

(3) Find pairs of faces corresponding to the same persons in the query video and in the database (Figure 2(b)).

(4) Computation of the similarity (Equation 3) only on those points in the signature where it has the potential to be maximum (Figure 2(c) and Figure 2(d)), as proved in Appendix A.

3.1 Algorithm Description

As already mentioned, our method characterizes video segments based on whether the faces of specific persons appear on them. We assume that an appropriate face detector and a face recognizer already exist, and their results are taken to be a given and are not a subject of investigation themselves. They are only of interest inasmuch as they influence the performance of retrieval and fingerprinting algorithms.

In the following, when we declare a sub- or super-scripted Q, we will assume it is a quartet of the form $Q = \{s, F, a, b\}$, where s, F, a and b have the same sub- and super-scripts as Q. Sets of quartets will be noted in bold.

When the video database is initialized, a database index I_{sa} is created over all the signature quartets $\mathbf{Q}^{db}$ in the database, indexing them first on the person identity s and then on the start frame a. It is assumed that the videos are arranged sequentially in the database. Two other indexes I_a and I_b are created based on the quartets' start frame a and end frame b alone. These indexes are crucial for enabling near-logarithmic access (with respect to the size of the database) to the quartets in the database. This is further explored

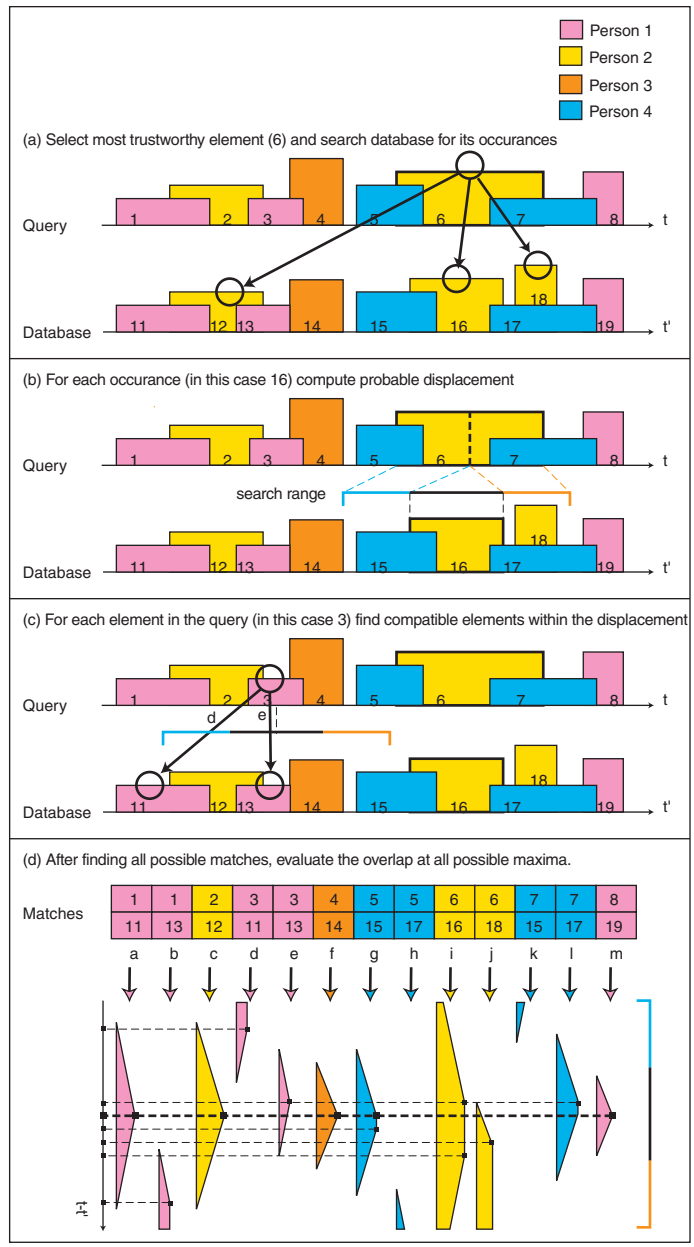

Fig. 2. Graphical overview of the signature search and matching algorithm. Different shades of grey correspond to distinct individuals. Signature quartets are represented by numbered rectangles.

in Section 3.2.

The following algorithm (illustrated in Figure 2) is proposed for finding matching segments in the database with respect to a query segment $\mathbf{V}_{query}$, which

is characterized by a signature consisting of a set of quartets $\mathbf{Q}^{query}$:

(1) Find the quartet in $\mathbf{Q}^{query}$ that has the greatest area (duration $\times$ recognizability) in order to use it as a base for searching, and name it the *trusted* quartet Q^{trust}. Thus the trusted quartet has the following property:

$$F^{trust}(b^{trust} - a^{trust}) = \max_j F_j^{query}(b_j^{query} - a_j^{query}) \tag{4}$$

(2) Find (through the index I_{sa}) all quartets $\mathbf{Q}^{base}$ in the database that refer to the same person as Q^{trust}:

$$\mathbf{Q}^{base} = \{Q^{base} \in \mathbf{Q}^{db} : s^{base} = s^{trust}\} \tag{5}$$

These will be used as the base for evaluating the segments around them, and be named *base* quartets (Figure 2(a)).

(3) For each base quartet $Q_i^{base} \in \mathbf{Q}^{base}$ found in the previous step:

(a) Add the pair consisting of the current base quartet Q_i^{base} and the trusted quartet Q^{trust} into a new list L, which will contain pairs of compatible quartets, i.e. quartets from the candidate segment (in the database) and the query segment which refer to the same person.

(b) Calculate a displacement window $[a_i^{disp}, b_i^{disp}]$, centered on Q_i^{base}, for finding possible matches in the database (Figure 2(b)), where:

$$b_i^{disp} = \frac{(b^{base} - a^{base})}{2} + \frac{(b^{trust} - a^{trust})}{2} \tag{6}$$

$$a_i^{disp} = -b_i^{disp} \tag{7}$$

(c) Then using the current base quartet Q_i^{base}, which we have found in the database, do the following for each query quartet $Q_j^{query} \in \mathbf{Q}^{query}$:

(i) Find (through the database indices I_a and I_b) the set of compatible quartets $\mathbf{Q}_{ij}^{comp}$ in the database, i.e. those quartets that belong to person s_j^{query} and which overlap with a window of size $b_i^{disp} - a_i^{disp}$ which is centered on Q_i^{query}.

(ii) If no quartets are found, increment a counter n. If, for all query quartets examined so far for the current base quartet Q_i^{base} we have $n > T_{reject}$, where T_{reject} a threshold, then proceed to the next database quartet Q_{i+1}^{base} that has the same person with Q^{trust}.

(iii) Add the pairs consisting of Q_j^{query} on the one hand, and all the recovered $Q_{ij}^{comp} \in \mathbf{Q}_{ij}^{comp}$ on the other, into the list L. It should be noted that, because a face cannot exist more than once in each frame, the intervals in $\mathbf{Q}_{ij}^{comp}$ do not overlap.

(d) Extract from list L a pair of quartets that have been accumulated in the above steps. Name the pair Q_l^{left}, Q_l^{right}. Then, for each pair:

(i) Evaluate the area of overlap v_{il} of Q_l^{left}, Q_l^{right} for all displacements d_{il} between the query segment and the candidate segment that correspond to possible maxima of the value of this area. These displacements are proven to be those and only those that have $a^{left} + d_{il} = a^{right}$ or $b^{left} + d_{il} = b^{right}$. The proof is given in in Appendix A.

(ii) Select the maximum match quality $v_i^{optimal} = \max_l v_{il}$ and also keep the corresponding displacement $d_i^{optimal}$. As we have seen in Section 2.2, this equates to finding the maximum similarity (as per Equation (3)) when using this base quartet. If all Q_i^{base} were rejected due to T_{reject}, do not return a $v_i^{optimal}$ (effectively set it to ∞).

(e) Clear the list L.

(4) Select the final similarity $v^{optimal} = \max_i v_i^{optimal}$. If no $v_i^{optimal}$ were returned due to T_{reject}, the algorithm returns a result of *"not found"*. Alternatively, if the user requires more than one match, keep the n highest values of $v_i^{optimal}$. In the case of fingerprinting, if the ratio between this similarity and the area of the original query segment (i.e. the error) is below a threshold T_v, then declare a match, otherwise declare no match. Also keep the corresponding displacement $d^{optimal}$.

(5) Optionally, if no match is found repeat all above for the next most trustworthy quartet. In our experiments we have done so.

Note that, if one wishes to continue verifying the retrieved segment Step 3c can be repeated for the quartets beyond $\mathbf{Q}^{query}$, keeping $d^{optimal}$ but adjusting $v^{optimal}$ and checking if the error exceeds a modified threshold T'_v. This is particularly useful when the query segment is a part of a larger video, e.g. is derived from streaming video.

3.2 Computational Complexity Analysis

The computational complexity of the proposed algorithm is obviously not constant, but depends on the distribution of faces in the query segment and the video database. In the following, M is the number of persons in the database, G_m the number of appearances of person m in the database and G the total number of quartets in the database. Additionally, G_{query} is the number of quartets in the query segment, H_g the number of quartets in the neighborhood of quartet Q_g that correspond to the same person s_g as Q_g, and $\overline{H_g}$ the average of H_g for all quartets.

First, in Step 2 of the algorithm, an indexed search for a specific person is performed, having (due to the index) a complexity of $O(\log M)$. Then, in Step 3, all quartets of this person are processed, giving a complexity of $O(G_m)$. In Step 3c, for each quartet found in Step 3 and for each quarter in the query, an indexed search for nearby quartets is performed, with a complexity of $O(G_{query} \cdot \log G)$. Those quartets that do not have enough matches are rejected. Since it was observed that the percentage of quartets rejected is proportional to G_{query}, the final complexity of Step 3c ends up being $O(\log G)$. In Step 3d, all quartets of Step 3c that have not been rejected are checked for overlap with nearby quartets, with a complexity of $O(H_g)$. Thus the total complexity of the algorithm is $O(\log M \cdot \log G \cdot G_m \cdot \overline{H_g})$.

Of the above factors, only G_m and $\log G$ are significant, since $\overline{H_g}$ is consistently small (usually less than 10) and M is much lower than G, causing $\log M$ to affect performance even less than $\log G$. G, being the number of signature quartets in the video database, is quite a large number (from our data we have estimated one quartet for every 5 seconds of video), but since its logarithm is taken it does not impact performance very much. The biggest factor influencing complexity is G_m, since it is not logarithmic and can be quite high. In practice, since the search is based on only one quartet per query segment (i.e. the "trusted" quartet in Section 3) and all G_m are known in advance for the database, we can avoid using quartets corresponding to persons with very high G_m (i.e., those that refer to very popular actors) as trusted quartets. This optimization was employed in our experiments, with little degradation in retrieval and fingerprinting performance.

4 Experiments on an Artificial Database

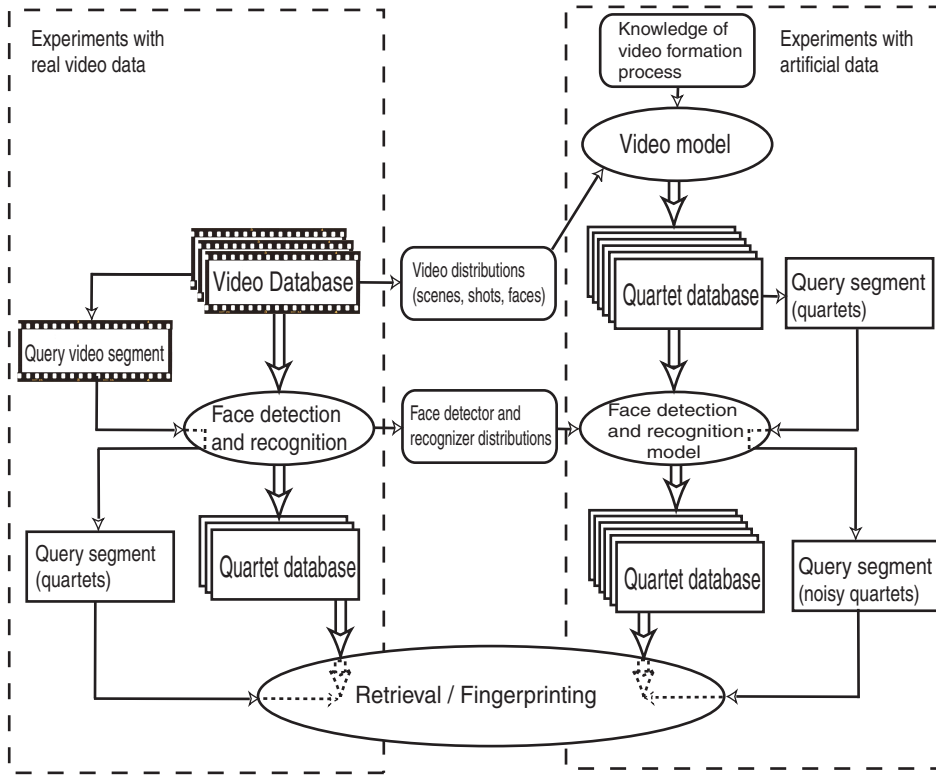

Fig. 3. Schematic flowchart of the creation and use of the artificial video signature database, juxtaposed with the regular retrieval/fingerprinting process.

The interest of this work is to investigate the performance of the proposed video indexing and fingerprinting method when applied on large databases (typically containing hundreds of hours of video). However, the effort of applying different types of face detectors and recognizers on large video databases, in order to derive the data required for the experimental performance evaluation of the proposed indexing and fingerprinting method, is extremely high. Therefore we have elected to perform the quantitative part of the experimental testing of our algorithm on appropriately constructed artificial data, an approach that has been followed before in the field of video indexing, e.g. in [5]. To achieve this, we have formulated a probabilistic model which describes

the ground truth of the appearance of faces in videos, and a second probabilistic model which describes the behavior of the face detection and recognition module when used to derive the signature from the query segment, as illustrated in Figure 3. The output of these models are sets of video signatures consisting of quartets.

This part of our approach has the advantage that we can easily test our algorithm on videos and face detection and recognition methods that have different characteristics by varying the parameters of the model. In the case of the ground truth model, the model parameters can be obtained by manually annotating a small corpus of video data. In the case of the face recognition and detection model, the model parameters could be obtained by running the appropriate face detection and recognition algorithms on a sufficient set of data. However, in our experiments we have varied these model parameters in order to explore the behavior of our method with respect to face detection and recognition algorithms with different characteristics.

4.1 Statistical Modeling of Face Content of Video

We model the appearance of persons in a video by considering the fact that the video is inherently composed of scenes, which are in turn composed of shots. Since scenes are spatio-temporally continuous in the context of the depicted world, and shots are spatio-temporally continuous in the video domain, we can assume different probabilities of appearance of a specific person for each scene and shot.

In order to construct the above model we needed three sets of information:

(1) *The structure of the model*, i.e. the random variables it contains and their interrelations. This was constructed by analyzing the motion picture production process.

(2) *The parameters of the random variables of the model* (mean, standard deviation etc). To estimate these, as well as the actual distributions mentioned below, we have first manually annotated a moderately large corpus of video data (approximately 100 minutes) by marking the faces appearances as well as the scene and shot boundaries present. Then we used these data to compute the statistics of their distributions (means and standard deviations).

(3) *The specific probability distributions of the random variables appearing in this model.* We tried to find appropriate distributions by using a combination of statistical testing and analysis of the physical meaning of the variables. Specifically, we posited several possible distribution models that might explain the distribution of our data, ranging from simple exponential and Gaussian distributions (for continuous variables), and binomial and geometric distributions (for discrete variables), to more complex expressions that stem from our interpretation of the physical and technical/artistic processes that result in film production. Then, using the parameters we have obtained in the previous step, we created candidate distributions and used statistical testing to compare them with the empirical ones extracted from the manually annotated data. More specifically, we applied the Kolmogorov – Smirnoff test (for continuous variables) or the χ^2 test (for discrete variables) [24] on the various candidates, and selected the ones that gave the best match.

In the following we give an outline of the model used. Specifically, we describe

the various random variables that comprise the model, together with their distributions. Although some of these variables are in practice not entirely independent, we will assume that they are. One reason for doing this is to limit the complexity of the model. Also, the dependence of the variables does not really affect the behavior of the retrieval and fingerprinting algorithms when applied to the video signatures produced.

(1) *Importance of each person in the video.* This expresses how significant a specific person (actor, etc) is in the video (e.g. lead actor, actor having a smaller or larger part, extra), and directly influences the probability of his/her appearance in the video. Since only the relative ranking of the importance of different persons is used in the following, this can be modelled with an arbitrary probability distribution. Here we simply use a uniform distribution between 0 and 1.

(2) *Probabilistic model of the scenes and shots of the video.* Three random variables are used for this purpose:

   (a) *The number of shots in a scene.* This was observed to follow a Poisson distribution with $\lambda = 25$.

   (b) *The length of each shot.* This was best approximated by the sum of two random variables, one being uniformly distributed with a mean of 1 second and a standard deviation of 0.5 seconds, and the other being exponential with a μ of 3.5 seconds. The density function of this distribution is shown in Figure 4. The form of the selected distribution model can be attributed to the two tasks a director must achieve in a shot: first to establish the visual context (the uniformly distributed variable) and second to narrate the action (the exponential one). Other works have modelled shot length with different

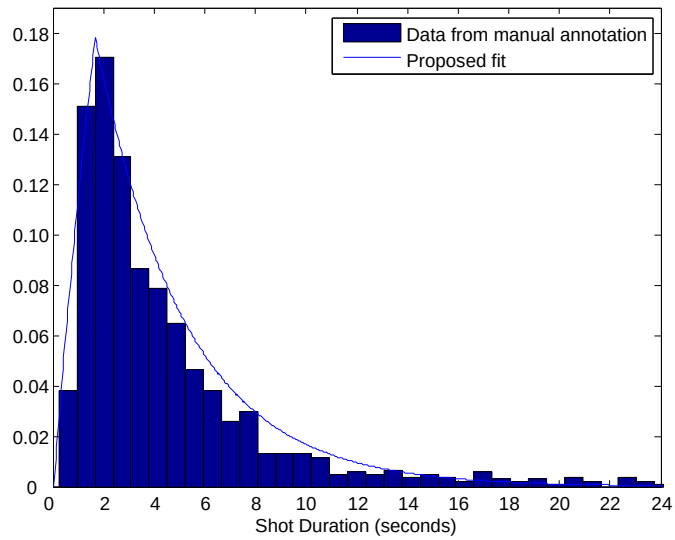

Fig. 4. The random distribution approximating the length of each shot.

distributions, for example Weibull [25] and Lognormal [26]. However we have found that our model gives a better overall fit for the data derived from the manually annotated videos.

(c) *The average size of faces in the shot*, which was found to be best described by a Rayleigh distribution with a σ of 0.15 of the size of the video frame in pixels.

(3) *Number of persons appearing in each scene.* The number of persons was found to be best approximated by a binomial distribution, with a mean of 2.5 and a standard deviation of 1. The actual persons that will appear in the scene are selected by assigning to each a random uniformly distributed number, multiplying it by the person's importance in the video, and selecting the ones with the highest scores.

(4) *Importance of each person in the scene.* This is again given by a random variable uniformly distributed between 0 and 1, as in the case of the whole video (see case 1 above), and for the same reasons.

(5) *Number of persons in each shot.* Again, as in the case of scenes, the

number of persons in the shot was found to be best approximated by a binomial distribution, with a mean equal to 0.5 times the number of persons in the corresponding scene, and a standard deviation equal to 0.25 times the number of persons in the scene. The binomial distribution represents the number of successes of n experiments when each experiment has a probability of success equal to θ. Furthermore, the mean of the binomial distribution is equal to $n\theta$ and its standard deviation equal to $n\theta(1-\theta)$. Thus, θ equals 0.5 and n equals the number of persons in the scene. Therefore the experimental observation can be interpreted as demonstrating that each person in the scene has a probability of 0.5 to appear in the shot. Of course, in practice, the actual probabilities for each person are different, as they depend on his importance in the scene.

(6) *Importance of each person in the shot.* This depends on the importance of the specific person in the corresponding scene. Effectively, we multiply the importance of the person in the scene with another uniform distribution between 0 and 1.

(7) *Appearances of persons.* In order to describe a certain appearance of a person in a shot, four random variables are needed. These are person identity, face size (in the frame), frame of appearance and frame of disappearance. In more detail:

   (a) *Person identity s.* This is randomly chosen from the persons in the shot, with the chances of selection being proportional to a person's importance in the shot.

   (b) *Average face size in the appearance L.* This was found to be approximated adequately by a random variable following uniform distribution between 0 and 2, multiplied by the average face size in the shot. Face size is used in order to derive an estimate of the recogniz-

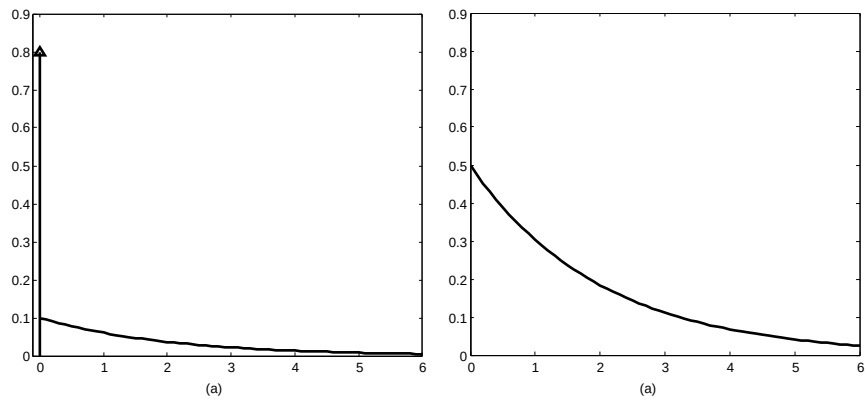

Fig. 5. The random distributions for approximating the start and end frames of quartets. (a) the distance from the start or end of a shot. (b) the distance from the previous quartet corresponding to the same person.

ability of each person. In practice the details of this distribution are not very significant, as we are concerned mostly with ratios between certainties in quartets.

(c) *Frame of appearance a*. The video frame where a person appears for the first time in a shot is specified through the distance of this frame from the first frame in the shot. For subsequent quartets of the same person the frame of appearance is computed through the distance from the last frame of the previous quartet of the same person in the shot. The probability distribution of the distance from the first frame in the shot can be approximated as the sum of an impulse at 0 with a height of 0.8 (i.e. it has a 0.8 probability of being zero) and an exponential with a μ of 2 seconds, scaled by a factor of 0.2 so that the sum is a valid probability distribution, as shown in Figure 5(a). The probability distribution of the distance from the end of the previous quartet is approximated by an exponential distribution with a μ of 2 seconds, as shown in Figure 5(b).

(d) *Frame of disappearance b*. This was found to be best expressed with

respect to its distance from the end of the shot. The probability distribution of the distance from the end of the shot was approximated as the sum of an impulse at 0 with a height of 0.8 (i.e. has a 0.8 probability of being zero) and an exponential with a μ of 2 seconds, again multiplied by 0.2. This is the same distribution as the one used in the previous paragraph for the distance of the first quartet of a person in the shot from the beginning of the shot.

4.2 Statistical Modeling of the Output of Face Detection and Recognition

For modeling the behavior of a face detector and recognizer, we assume that they introduce inaccuracies, which act as noise on the ground truth described in Section 4.1. We also assume that the work of the recognizer is aided by a tracker that follows faces from one frame to the next. The following errors can be introduced by these modules:

(1) A split in the middle of a quartet. This error simulates the case when the face tracker stops to track a face, and then the face detector finds it again, starting a new quartet. The probability of such an error depends on the length and recognizability of the person's appearance.

(2) Face detection and recognition with a recognizability different than the ground truth.

(3) Detection of a non-existent face. In effect a new quartet is inserted into the signature.

(4) Wrong estimation of the start and end frames of a quartet. The wrong estimation of the start frame can be due to periodicity in the initialization of the tracker (e.g when a face detector is applied every 10 frames in order

to initialize the tracker). The wrong estimation of the end frame can be due to a premature end of tracking or false continuation of tracking. In addition, both can happen due to the failure of face detection.

(5) Failure to detect a face appearance. This is simulated by random deletions of quartets.

(6) Face misclassification. It occurs when a face is misclassified as belonging to a different person.

(7) Merging of two quartets of the same person. It can occur when the face tracker ignores a break between two spatially and temporally proximate appearances of the same person. This is a rare error.

For testing the performance of our algorithm, both in terms of precision and of computational complexity, data created according to the above models were used. These data consist of ground truth sequences that are modified by the face detector and recognizer model. We should note that no explicit modelling of phenomena such as compression, cropping, video noise etc are required. Such manipulations ultimately only affect the output of the face detection and recognition modules and, thus, their effect can be included in the model of the employed face detection/tracking/recognition module.

4.3 Computational Performance

In order to evaluate the computational performance of our algorithm, we created artificial video signature databases of different sizes. Each database consisted of signatures from a number of videos, each having a duration of 90 minutes and containing between 1000 and 2000 quartets. The number of different persons for each database was chosen to be 10 times the number of

videos. We then selected query segments with average lengths ranging from 2.5 to 10 minutes and ran our search algorithm on these segments, using a commercial RDBMS system for the implementation. We observed that the length of the query segments did not influence the search time, which is consistent with the theoretical computational analysis in Section 3.2. Using a computer significantly behind the state of the art (Pentium 4 at 2.4 GHz), the average times for the retrieval of a single segment are given in Table 1. As it can be seen, the performance of the algorithm is near-logarithmic with respect to the size of the database, which again conforms with the theoretical analysis described in Section 3.2. This is in contrast to the cost of exhaustive frame-by-frame computation of the signature similarity, which is constant at about 6 seconds per video (i.e. approximately 4 seconds per hour of video). In addition to the search times, face detection and recognition would add another 1 to 5 seconds per second of video with the above hardware, depending on the sampling rate. With better hardware, it would be possible to achieve real time performance for the system. We should note that the cost of the insertion of a video to the database is only the cost of face detection and recognition, i.e. 1 to 5 times the duration of the video with the present hardware, or much less with better hardware.

4.4 Retrieval Performance

Given that neither the face detection algorithms, nor the face recognition algorithms are perfect, we performed a series of experiments to test the retrieval performance of our algorithm in the presence of noise introduced during face detection and recognition. In order to be able to extract meaningful conclu-

Table 1

Average search time results

Number of videos	Number of quartets	Algorithm search time	Brute force time
100	152,791	7 seconds	10 minutes
1,000	1,682,824	17 seconds	100 minutes
10,000	16,907,355	41 seconds	16 hours

sions from our experiments, we did not explore all the 7 possible categories of noise that were described in Section 4.2. Doing this would have created an unnecessary amount of data which would be largely redundant, since many of the noise types have similar effects. Specifically, noise types 3 (detection of a non-existent face) and 5 (non detection of an existing face) can be approximated as a special case of noise type 6 (change of the person that is recognized) because of the way that signature similarity works. Additionally, the effect of noise types 1 (split of the quartet in two), 2 (change of the recognizability) and 7 (merging of two quartets of the same person) on our algorithm is similar to the effect of type 4 (change of start and end frames of quartet), since all three only affect the amount of overlap between the rectangles that correspond to the quartets. As a result, noise types 3 and 5 can be subsumed by type 6, and types 1, 2 and 7 can be subsumed by type 4. Thus the following types of noise have been considered:

(1) Change of quartet start and end frames (hereafter called *face detector noise*). This is one of the most typical errors made by face detectors and trackers. Exponential noise has been added to the start time of a quartet

(to simulate a delayed detection), and zero mean Gaussian noise to the end time of the quartet (to simulate either early loss or false continuation of tracking). It should be noted that, when the noise level is high, it can result in the complete elimination of a quartet or the merging of quartets. The standard deviation of the noise varied from 1 to 5 seconds in steps of 1 second, for both the start and end frames of the quartets. We varied the noise concurrently for both the starts and ends of the quartets, in order to reduce the amount of experiments that need to be done, as varying the start and end frame noise independently would require 25 experiments per set (instead of 5). The mean of the noise was equal to the standard deviation in the case of a quartet's start frame (since the distribution is exponential), and zero in the case of a quartet's end frame (as mentioned above).

(2) Change of the person's identity in a quartet (face recognizer noise). This is a typical error made by face recognizers. Here we assumed a probability (between 2.5% and 40%) that a person's identity would be randomly changed to another one.

This set of experiments was run using an artificial signature database of 1000 videos, each 60 minutes long. From this database we randomly extracted 4 sets of 100 segments each. The segments in the first set was chosen to contain 16 quartets, and had an average duration of 2.5 minutes, those in the second 32 quartets and 5 minutes, those in the third 48 quartets and 7.5 minutes, and those in the fourth 64 quartets and 10 minutes. On each set we added noise representing face detector and recognizer errors, as described above, and then proceeded to seek them in the database. If the best match that was retrieved temporally overlapped the original segment by at least 50%,

a correct retrieval was marked. A misretrieval was declared when the best match did not correspond to the original segment, and a non-retrieval when no appropriate matching segment was found. If the n ($n > 1$) best matches were taken into account, the original segment would have a higher probability to be included therein, and would therefore improve retrieval performance. Thus, in our experiments we have chosen to only retrieve a single match, in order to explore the worst-case scenario. The algorithm used was as described in Section 3, with T_{reject} equal to half the number of quartets in the query segment. Since, in this case, it is desirable for the algorithm to always return a matching segment, T_v was not used. As we have noted, modeling of the effects of noise on the video itself is unnecessary.

The retrieval performance of our algorithm with respect to query segment size, detector noise and recognizer noise is shown in Figure 6. We can see that for low noise levels the performance is always satisfactory. For high noise levels, especially for face detection noise, the performance drops significantly when few signature quartets (e.g. 16) are used. However we should note that the average quartet length (both in real and artificial videos) is less than 4 seconds, while the added noise (change in the start and end frames) had a mean that was as much as 5 seconds. This means that, in effect, the amount of noise added in the most extreme experiments was comparable to the signal being observed. Nevertheless, the search algorithm proved to be very robust, since it is able to utilize the temporal redundancy of the signature to resolve ambiguities. In addition, increasing the length of the query segments greatly diminished the effect of the misbehavior of the face detection and recognition.

Figure 7(a) shows the average temporal accuracy of the signature matching in the database. In other words, this figure depicts the average difference be-

tween the true displacement d and the estimated displacement produced by the matching algorithm. One can see that the displacement error is approximately equal to half the detector noise mean, a fact that proves that the accuracy of the method is very satisfactory.

4.5 Fingerprinting performance

As was mentioned already, video fingerprinting i.e. verifying whether a certain video segment is a (possibly modified) copy of any part of the videos in a large database, has significantly different requirements than simple retrieval. The main difference is that a fingerprinting algorithm should, when queried with a certain video segment, return a video from the database only if the query segment is a (possibly modified) copy of this video, and return an empty set when this does not hold. On the other hand a retrieval algorithm needs only to return the most similar match. Therefore, unlike in the case of simple retrieval, in the case of fingerprinting the threshold T_v is used to control the acceptance of matches by rejecting those having a similarity ratio less than T_v. As was explained in Section 3, the similarity ratio is the ratio of the total area of overlap between two segments, divided by the total area of the first segment. In our experiments we have varied the value of T_v between 20% and 40% of the total area of the quartets of the query segment. We use a threshold that depends on the area of the quartets of the query segment, because the signature similarity we have defined in Section 2.2 is generally proportional to this area.

In order to test our algorithm in the fingerprinting context, we have run a series of experiments by varying the threshold T_v and seeing its effect in the following

Table 2

Performance of Algorithm Adapted for Fingerprinting

Threshold T_v		20%				30%				40%			
Query Length		16	32	48	64	16	32	48	64	16	32	48	64
False Acc. (%)		20,2	3	1.8	1.4	30.3	6.7	3	2.1	38	11.4	4.5	3
Recogn. Noise [b]	Detect. Noise [a]	**False Rejection (%)**											
5%	1sec	14	8	7	6	5	2	4	3	1	0	2	3
10%	1 sec	32	22	23	22	17	7	4	4	11	4	4	4
5%	2 sec	23	22	23	28	7	4	3	4	2	2	2	3
10%	2 sec	34	43	52	53	14	11	7	12	3	7	7	6
Recog. Noise	Detect. Noise	**Misretrieval (%)**											
5%	1 sec	3	0	0	0	4	0	0	0	5	0	0	0
10%	1 sec	2	0	0	0	5	0	0	0	6	0	0	0
5%	2 sec	13	4	1	1	16	4	1	1	18	4	1	1
10%	2 sec	8	1	0	0	11	2	0	0	12	3	0	0

[b] Probability of false recognition.

[a] Mean deviation of the noise added to start and end quartets.

cases: (1) when using query video segments that existed in the database, and (2) when using query video segments that did not exist in the database. In both cases, all experiments were performed by varying the size of the query segment between 16 quartets (i.e. 2.5 min) and 64 quartets (i.e. 10 min), in steps of 16 quartets. In the first case we used a procedure similar to the one described in the previous section, except that we also varied T_v as well as the query segment size and the characteristics of the two types of noise, i.e. change of quartet bounds (detector noise) and person identity (recognizer noise). Two sorts of errors can occur in this case: falsely marking a query video segment as having no matching segment in the database (*false rejection*), and retrieving an incorrect segment from the database (*misretrieval*). In the second case, we created a set of 1000 videos that were different from the ones in the database. Since the content of the new videos was completely unrelated to those in the database, there was no need to alter them in order to represent failures in face detection and recognition. Using different values of T_v, we then run the algorithm on the same artificially created face signature database as in the previous section, using parts of the new videos as query segments. Erroneously accepting a query segment as being identical to one of the segments in the database (*false acceptance*) is the only possible type of error in this case.

The results are given in Table 2, where the strength of the detector noise is given as the mean deviation of the change in the start and end frames of the quartets (in seconds), and the strength of the recognizer noise is given as the percentile probability of false recognition. Additionally a ROC curve showing the effect of T_v is shown in Figure 8. As expected, the results are worse than in the case of simple retrieval. False acceptance and false rejection rates are unacceptable for query segments of 16 quartets (2.5 minutes of video), but they

improve greatly when the size of the query segment increases. For example, for a query segment of 64 quartets, and with moderate noise (2 seconds detector noise and 10% recognizer noise) the false acceptance rate is 3% and the false rejection rate is 6%. Again, as in retrieval, Figure 7(b) depicts the average difference between the true displacement d and the estimated displacement. In this case we have chosen a face detector noise standard deviation equal to 1 second, a face recognizer failure rate equal to 10% and varied T_v to examine its influence on the displacement.

It should be noted that, if the query segment is large enough, it is possible to continue verification beyond 64 quartets and thus have an even smaller number of false acceptances. Moreover, since only one segment is retrieved (out of the thousand hours in the database) it is also possible to use other, more costly methods (e.g. frame-by-frame comparison) to verify that this segment is the correct one, and thus further reduce false acceptance. By reducing false acceptances in this way, it is then possible to choose a less strict T_v to also reduce false rejections.

5 Tests on Real Video Data

In order to demonstrate the feasibility of the proposed indexing and fingerprinting method in real-world video databases, we had initially implemented and tested a complete system for video indexing and fingerprinting. Five complete motion pictures of various genres and 15 episodes of one drama and one comedy series provided a sufficient data corpus for the evaluation. In total the corpus comprises over 16 hours of video, with various resolutions and aspect ratios. Motion pictures and TV series were chosen as a test corpus because the

human faces in them exhibit a full spectrum of pose, lighting and scale, and also different emotions, hairstyles and apparel (sunglasses etc). In contrast corpora such as news broadcasts mostly contain frontal, frontally illuminated and emotionally neutral faces, in specific attire and hairstyles. Moreover, from the standpoint of intellectual rights protection, this type of video is obviously the most interesting.

The face detection and recognition required for constructing the signatures was performed using the FaceVACS toolkit, produced by Cognitec Systems GmbH [27], which very close to the state of the art in the field [28]. We chose a commercial product in order to enhance the robustness and verifiability of our retrieval and fingerprinting system. In order to reduce processing time, only 5 frames per second were processed but this was found to be adequate for the operation of the algorithm.

FaceVACS functions by localizing faces and eyes in each frame of the video, and doing appearance-based feature extraction on each such face. The features thus extracted are then compared with a reference set, which is constructed by performing face detection and recognition, as above, on a number of reference images. Our reference sets were constructed from approximately 25 images for each significant person in the videos of our database. The significant persons that were chosen as targets of recognition were the main actors in the motion picture or TV series. These were mostly those that appeared in the starting credits, ranging from 5 to 10 per motion picture or series.

The output of FaceVACS consists of the location of a face in a frame, the identities of the top three matches for this specific face, and the certainties of each match. In order to exploit the temporal continuity between frames, a proce-

dure that greatly increases the performance by means of a voting scheme that rejects outliers and reinforces detections having a high recognizability was implemented. Initially, the faces detected by FaceVACS were unified into tracks using their spatio-temporal proximity. Then a single identity was determined for each track by a voting scheme that uses the recognition scores of the frames in each track. Finally the corresponding recognizability was computed.

5.1 Results

Having created a database of real video signatures using FaceVACS, we proceeded to select 40 clips from the database, each having a duration of 2.5 minutes and constituting in total about 10% of the database. Additionally, we performed on them some operations that simulate the changes and/or attacks that videos may be subject to in real world situations. Such changes included change of compression, change of resolution, cropping, change of frame rate, and conversion to greyscale. The clips were then processed by FaceVACS, using the union of all reference sets used in the database as a basis for recognition. We then applied our retrieval algorithm on these clips with reference to the whole database. The result was a correct retrieval score of 90%. Fingerprinting experiments were then performed, with leave-one-out methodology for finding false acceptance. A T_v equal to 40% resulted in a false acceptance rate of 12.5% and a false rejection rate of 15%. When T_v falls to 30% and 20%, false acceptance rises to 17.5% and 25% respectively, while false rejection falls to 0% for the specific dataset. These results have verified that our algorithm can function efficiently in a real-world situation.

6 Conclusions

A method for performing fast retrieval and fingerprinting in video based on the output of face detectors and recognizers has been presented. The proposed method is both robust because it is based on a convolution-like video content similarity computation, and fast because it makes extensive use of database indexing. Experimental results were computed on artificial data based on realistic models of the appearances of faces in videos and of face detector/recognizer behavior that have been devised for this purpose. The results verified that the proposed method performs very satisfactorily, both in terms of computational search efficiency (even in a database of 10000 hours of video), in terms of retrieval errors, and in terms of fingerprinting performance. The effect of the malfunction of face detectors, trackers and recognizers on the performance of retrieval and fingerprinting was quantified through a large set of experiments. The results we have obtained for various levels of face detector and recognizer performance can help a potential user of the proposed system to determine its performance, when using a specific face detector and recognizer. The experimental results on artificial data have additionally been verified by the implementation of a real system that uses face detection and recognition to index real videos. In general the method proves that face related information carries enough discriminant power to be used for video indexing, retrieval and fingerprinting. The proposed face-based approach could also be used as it is using person identities computed by other means, e.g. through voice identification. Alternatively, it could be adapted in order to index video using the appearances of other classes of objects that possess distinct identities. Finally, it could also be used as a first stage in a retrieval or fingerprinting pipeline,

quickly retrieving candidates that would then be validated by other, slower but more accurate algorithms (e.g. using frame-by-frame information).

7 Acknowledgements

This work was developed within ECRYPT IST-2002-507932, European Network of Excellence in Cryptology (http://www.ecrypt.eu.org/), funded under the European Commission IST FP6 programme.

A Proof of the location of local maxima

Assume two pulse series $f(t)$ and $g(t)$ with $f(t) = \sum_{i=0}^{n} f_i(t)$ where $f_i(t) = c_i^f(u(t - a_i^f) - u(t - b_i^f))$ and $a_i^f < b_i^f, \forall i$ and $b_i^f < a_j^f \forall i < j$. In the same way $g(t) = \sum_{i=0}^{m} g_i(t)$ where $g_i(t) = c_i^g(u(t - a_i^g) - u(t - b_i^g))$ and $a_i^g < b_i^g, \forall i$ and $b_i^g < a_j^g \forall i < j$. The area of the overlap of f and g when displaced by t' with respect to each other is expressed by:

$$L(t') = \int min(f(t), g(t + t'))dt \tag{A.1}$$

We are going to prove that the all the values of local maxima of $L(t')$ can only occur for values of t' such that $\exists i, j : t' = a_i^f - a_j^g$ or $t' = b_i^f - b_j^g$, that is when the beginning of a pulse in $f(t)$ coincides with the beginning of a pulse in $g(t)$, or when the same thing happens with their ends.

Proof:

We have that $L(t') = \sum_{i=0}^{n} \sum_{j=0}^{m} L_{ij}(t')$ where:

$$L_{ij}(t') = \int min(f_i(t), g_j(t+t'))dt \tag{A.2}$$

We will first compute $L_ij(t')$ for two arbitrary pulses $f_i(t)$ and $g_j(t)$. For simplicity we change the notation to $f_i(t) = c_f(u(t-a_f) - u(t-b_f))$, $g_j(t) = c_g(u(t-a_g) - u(t-b_g))$, and we can assume without loss of generality that $b_f - a_f \geq b_g - a_g$ and $c_f \geq c_g$. Then:

$$L_{ij}(t') = \begin{cases} 0 & t' < a_f - b_g \text{ or } t' > b_f - a_g \\ (t' - a_f + b_g)c_g & a_f - b_g \leq t' < a_f - a_g \\ c_g(a_f - a_g) & a_f - a_g \leq t \leq b_f - b_g \\ (b_f - a_g - t')c_g & b_f - b_g < t \leq b_f - a_g \end{cases} \tag{A.3}$$

$$L'_{ij}(t') = \begin{cases} 0 & t' < a_f - b_g \text{ or } t' > b_f - a_g \text{ or } a_f - a_g \leq t' \leq b_f - b_g \\ c_g & a_f - b_g \leq t' < a_f - a_g \\ -t'c_g & b_f - b_g < t' \leq b_f - a_g \end{cases} \tag{A.4}$$

$$L''_{ij}(t') = c_g(\delta(t' - a_f + b_g) - \delta(t' - a_f + a_g) - \delta(t' - b_f + b_g) + \delta(t' - b_f + a_g)) \tag{A.5}$$

It is known that the condition for a local maximum in a function $\phi(t)$ is that $\phi'(t) = 0$ and $\phi''(t) < 0$. Another type of maximum is the regional maximum, where $\phi'(t) = 0, \forall t \in [a, b]$ and $\phi''(a) < 0$ and $\phi''(b) < 0$. However even in the case of regional maxima the value of the maxima can be found at the values of t such that $\phi''(t) < 0$. From Equation (A.5) we can see that since $L''(t') = \sum_{i=0}^{n} \sum_{j=0}^{m} L''_i j(t')$, $L''(t')$ is composed of pulses, and the negative ones

only happen when $t = a_i^f - a_j^g$ or $t = b_i^f - b_j^g$. Therefore all the values of local maxima occur at these values.

References

[1] N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor, "Applications of video-content analysis and retrieval," *IEEE Multimedia Magazine*, vol. 9, no. 3, pp. 42–55, July 2002.

[2] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *Proc. 5th International Conference on Recent Advances in Visual Information Systems (VISUAL 2002)*, Mar. 2002, pp. 117–128.

[3] J.Law-To, L. Chen, A. Joly, Y. Laptev, O. Buisson, V. Gouet, N. Boujemaa, and F. Stentiford, "Video copy detection: a comparative study," in *ACM International Conference on Image and Video Retrieval*, 2007.

[4] S. Pradhan and K. Tanaka, "A query model to synthesize answer intervals from indexed video units," *IEEE Transactions Knowledge and Data Engineering*, vol. 13, no. 5, pp. 824–836, 2001.

[5] S. Park and K.-H. Hyun, "Trie for similarity matching in large video databases," *Information Systems*, vol. 29, no. 8, pp. 641–652, July 2004.

[6] C. Boehm and S. Berchtold, "Searching in high-dimensional spacesindex structures for improving the performance of multimedia databases," *ACM Computing Surveys*, vol. 33, no. 4, Sept. 2001.

[7] J. Lee and B. Dickinson, "Hierarchical video indexing and retrieval for subband-coded video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 824–829, Aug. 2000.

[8] E. K. Kang, S. J. Kim, and J. S. Choi, "Video retrieval based on scene change detection in compressed streams," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 3, pp. 932–936, Aug. 1999.

[9] S. H. Kim and R.-H. Park, "An efficient algorithm for video sequence matching using the modified hausdorff distance and the directed divergence," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 592–596, July 2002.

[10] A. Hampapur and R. Bolle, "Comparison of sequence matching techniques for video copy detection," in *Proc. Conf. Storage and Retrieval for Media Databases*, 2002, pp. 194–201.

[11] K.-W. Sze, K.-M. Lam, and G. Qiu, "A new key frame representation for video segment retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 9, pp. 1148–1155, Sept. 2005.

[12] R. Fablet, P. Bouthemy, and P. Perez, "Nonparametric motion characterization using causal probabilistic models for video indexing and retrieval," *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 393–407, Apr. 2002.

[13] H. Yi, D. Rajan, and L.-T. Chia, "A new motion histogram to index motion content in video segments," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1221–1231, July 2005.

[14] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang, "On clustering and retrieval of video shots through temporal slices analysis," *IEEE Transactions on Multimedia*, vol. 4, no. 4, pp. 446–458, 2002.

[15] S. Dagtas, W. Al-Khatib, A. Ghafoor, and R. Kashyap, "Models for motion-based video indexing and retrieval," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 88–101, Jan. 2000.

[16] W. Zhao, R. Chellappa, P.-J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Survey*, vol. 35, pp. 399–458, 2003.

[17] C. Cotsaces, N. Nikolaidis, and I.Pitas, "Face-based digital signatures for video retrieval," *IEEE Transactions on ircuits and Systems for Video Technology*, vol. 18, no. 4, pp. 549–553, Apr. 2008.

[18] S. Eickeler, F. Wallhoff, U. Iurgel, and G. Rigoll, "Content-based indexing of images and video using face detection and recognition methods," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, May 2001, pp. 1505–1508.

[19] S. Satoh, "Comparative evaluation of face sequence matching for content-based video access," in *Proc. 4th International Conference on Automatic Face and Gesture Recognition(FG2000)*, 2000, pp. 163 – 168.

[20] M. Viswanathan, H. Beigi, A. Tritschler, and F. Maali, "Information access using speech, speaker and face recognition," in *Proc. IEEE International Conference on Multimedia and Expo (ICME 2000)*, July-August 2000, pp. 493–496.

[21] G. Wei and I. K. Sethi, "Omni-face detection for video/image content description," in *Proceedings of the 2000 ACM workshops on Multimedia*, Nov 2000, pp. 185–189.

[22] Y. Chan, S.-H. Lin, Y.-P. Tan, and S. Kung, "Video shot classification using human faces," in *Proc. IEEE International Conference on Image Processing (ICIP 1996)*, vol. 3, 1996, pp. 843–846.

[23] J. Viallet and O. Bernier, "Face detection for video summaries," in *International Conference on Image and Video Retrieval (CIVR 2002)*, 2002, pp. 348–355.

[24] G. W. Snedecor and W. G. Cochran, *Statistical Methods*, eighth edition ed. Iowa State University Press, 1989.

[25] N. Vasconcelos and A. Lippman, "Statistical models of video structure for content analysis and characterization," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 3–14, 2000.

[26] B. T. Truong and S. Venkatesh, "Finding the optimal temporal partitioning of video sequences," in *Proc. International Conference on Multimedia and Expo (ICME 2005)*, July 2005.

[27] "FaceVACS-SDK 5.0." [Online]. Available: http://www.cognitec-systems.de/products-sdk.htm

[28] "Face Recognition Vendor Test 2002." [Online]. Available: http://www.frvt.org/FRVT2002/default.htm

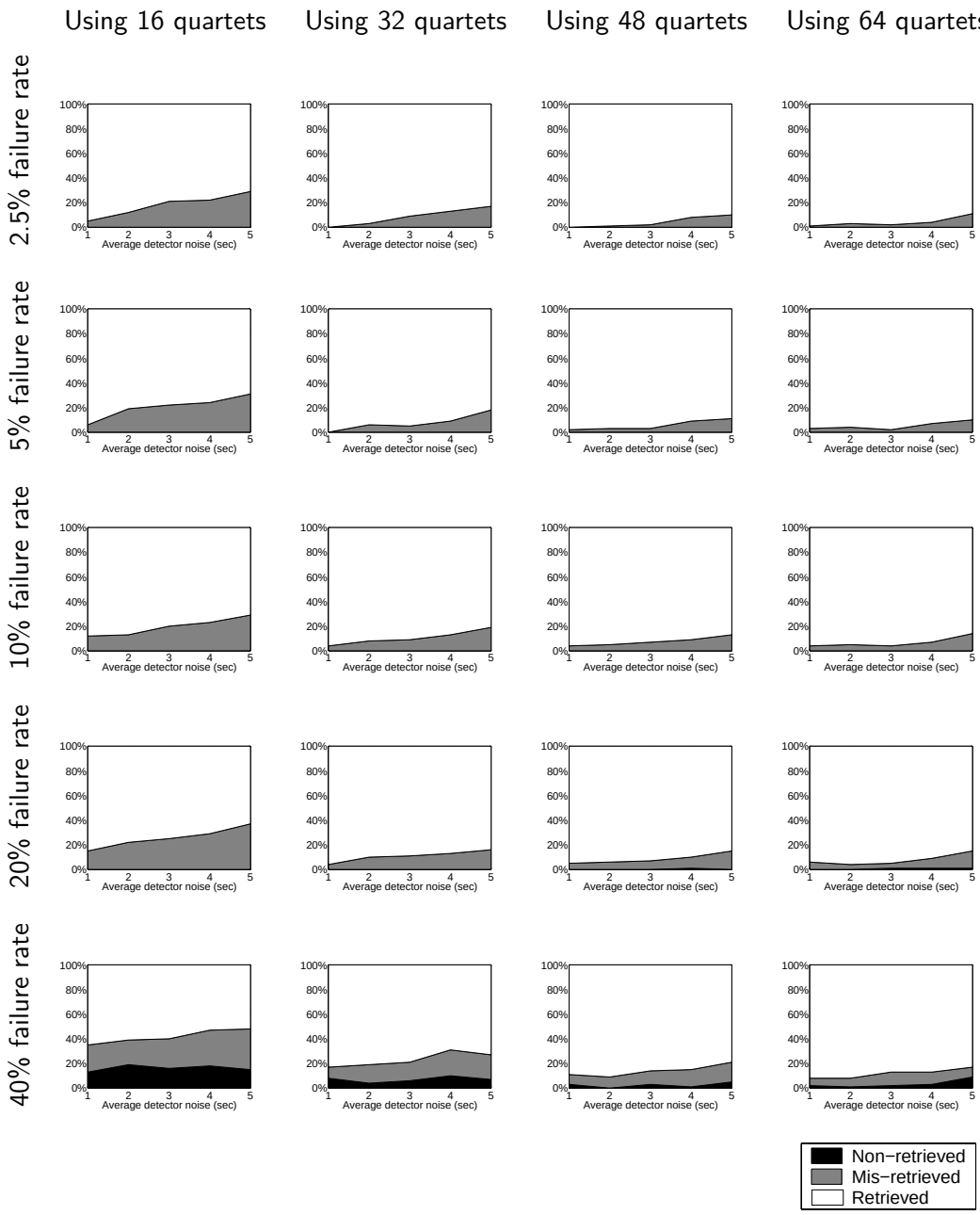

Fig. 6. Retrieval performance of the algorithm with respect to face detector and recognition performance and size of the query segment. Rows of graphs correspond to different failure rates of the face recognizer, columns refer to different lengths of the query segment, while the x axis of the graphs represent the σ of the noise added both to the start and end of the quartets of the query segment to represent the failure of the detector and tracker.

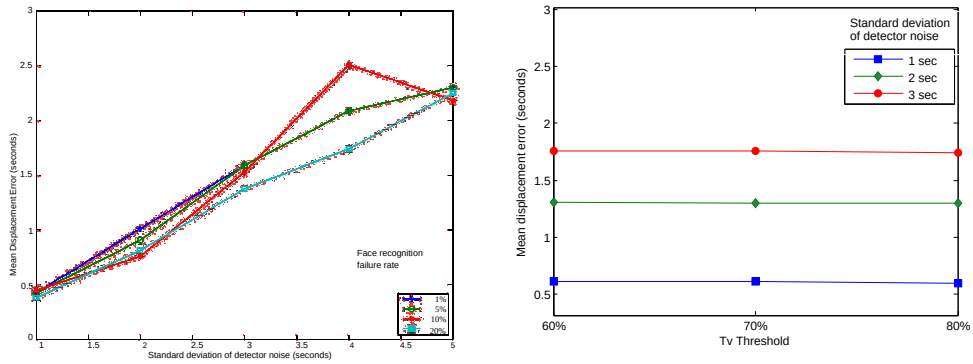

Fig. 7. Average temporal matching accuracy as a function of noise, (a) for retrieval and (b) for fingerprinting. The different lines in the retrieval figure correspond to different failure rates of the recognizer, while in the fingerprinting figure we keep the face recognition failure rates constant at 10% and the vary standard deviation of the face detector noise between 1 to 3 seconds and the threshold T_v from 20% to 40$.

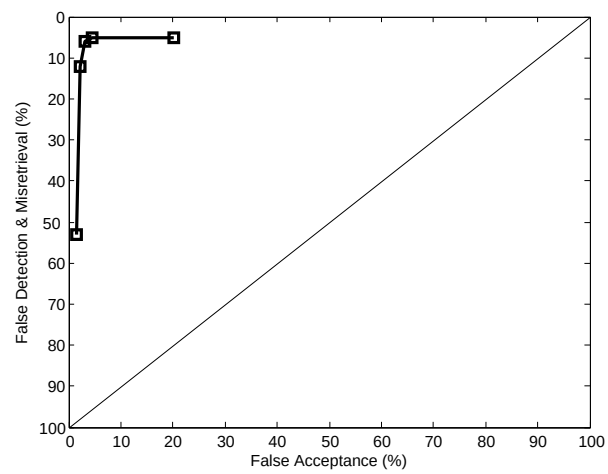

Fig. 8. ROC curve obtained by varying threshold T_v, for face detector noise equal to 10%, face recognizer noise equal to 2 seconds, and query segment length equal to 64 quartets. In this case we plot false acceptance against the sum of false detection and misdetection