CBVR for Face Based Digital Signatures

[1]M. Nagaraju, [2]M. Thanoj, [3]K. Lakshmi Tejaswini, [4]A. Koteswaramma
[1,3,4]Dept. of IT, Gudlavalleru Engineering College, AP, India
[2]Dept. of CSE, Andhra Loyola Institute of Engineering and Technology, Vijayawada, AP, India

Abstract

The characterization of a video segment by a digital signature is a fundamental task in video processing. It is necessary for video indexing and retrieval, copyright protection and other tasks. Semantic video signatures are those that are based on high-level content information rather than on low-level features of the video stream. The major advantage of such signatures is that they are highly invariant to nearly all types of distortion. A major semantic feature of a video is the appearance of specific persons in specific video frames. Because of the great amount of research that has been performed on the subject of face detection and recognition, the extraction of such information is generally tractable, or will be in the near future. We have developed a method that uses the pre-extracted output of face detection and recognition to perform fast semantic query-by-example retrieval of video segments. We also give the results of the experimental evaluation of our method on a database of real video. One advantage of our approach is that the evaluation of similarity is convolution-based, and is thus resistant to perturbations in the signature and independent of the exact boundaries of the query segment.

Keywords

CBVR, Video Retrieval, Face Recognition, Query-by-Example, Multimedia Databases

I. Introduction

One of the most fundamental technologies necessary for the management of digital video is the retrieval (from a video database) of one or more video segments that the user is interested in. The methods used for approaching video retrieval are similar to those used for the retrieval of other types of multimedia objects, such as images. In the case of both images and video, retrieval usually follows one of two paradigms:

A. 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.

B. Query-by-Example

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 selected image or video and used to find images or videos with similar features in the database, sometimes using advanced indexing techniques.

Methods that belong to both categories can be either semantic or non-semantic, based on whether the metadata or features they use have a semantic meaning or not. In general, query-by-keyword methods tend to be semantic, while query by- example methods tend not to be. However this distinction is not strict, since a keyword may refer to low level characteristics such as color and brightness, while an automatically extracted feature may be semantic, e.g. corresponding to an object in the image or video. The difference between image retrieval and video retrieval is that, since video has a temporal dimension, the video signature or its metadata must have a temporal component, i.e., it must either be continuous over time or be repeated at certain time intervals. Additionally, retrieval algorithms can be designed to return either one or many results. 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 this paper, we take advantage of a specific type of semantic information, namely information about the existence of faces of distinct individuals (e.g. actors), in order to characterize a video segment in a robust way. We do not concern ourselves with face detection and recognition, since ample work has been performed on both subjects [1]. This work tries to solve the problems of consistency and robustness with regards to face-based indexing, to represent face information with minimal redundancy and also to find a fast (near-logarithmic time) search method. Using face-related information for video indexing is not a new idea. However, most works until now [2-5] have focused on the extraction of the face-related information and not on its organization and efficient indexing. In effect, they are works on face recognition with a view to its application on indexing. As such, they actually present an excellent foundation for our work, in the form of detected and/or recognized faces. This is especially true for the works of Satoh [3] and of Eickeler et al [2], who perform identity recognition on the faces they detect. It would also be possible to extract face identity information not directly by face recognition but also through auxiliary clues, as in [6]. There has been some work on the characterization of video shots using face information [7], and on evaluating the similarity between different shots based on face information [8]. However, there has been no work on video retrieval with respect to large databases. In the present paper, we do not propose a new face detection and recognition method, but we investigate the performance of our retrieval system. The practicality of our system was verified by implementing a real system and testing its retrieval performance on a database of real video. The advantages of our algorithm are firstly that it is based on semantic information, and is thus robust with respect to video noise and manipulations, secondly that it is convolution-based and thus robust to change of query segment boundaries and to malfunctions of the face detector and recognizer, and thirdly that it is well suited to large video databases (up to thousands of hours of video).

The paper is organized as follows: In Section II, a description of the proposed algorithm is provided. Section III provides the experimental results. Conclusions are presented
in Section IV.

II. Description of Algorithm

The general idea of our approach is that the existence of faces of specific individuals can be used to characterize a video segment. Implicit in this approach is the use of a face detector module and a face recognizer module. The face detector module subsumes all other modules that are necessary for its function, such as a face tracker module. The output of these modules is assumed to be known. In the following, we first rigorously define the way we will use the output of the face detector and recognizer modules to construct our video signature. Then we give a measure for defining the similarity of two video segments based on their signatures. Finally, we present our algorithm for searching a database of video signatures in order to find the best fit for a given query signature.

A. Format of Signature

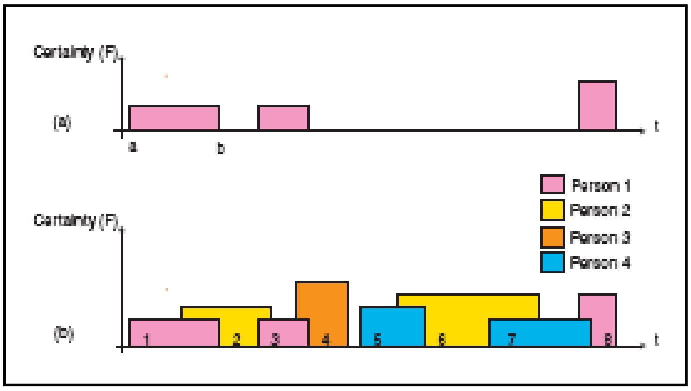


Fig. 1: Example of the Characterization of a Video Segment by Quartets. (a) Signature for a Single Person. (b) Signature of a Video Segment. Shades of Gray Correspond to Distinct Individuals. Signature Quartets are Represented by Rectangles

Let V = {f$_1$ f$_2$ . . . f$_N$} be a video consisting of a number of consecutive frames f$_n$, n = 1, . . . ,N that we wish to characterize through an appropriately constructed signature. Let S = {s$_1$ s$_2$. . . s$_M$} be the set of all the individuals s$_m$, m = 1 . . .M that have been imaged in the video. Optionally, with no loss of generality, we can assume S to contain only the individuals of interest. This can mean, for example, excluding the extras in a motion picture. Let us then assume a face detector and recognizer whose output is the certainty:

$$G(n,m) = \mathrm{Prob}\{s_m \text{ is imaged in } f_n\} \tag{1}$$

The face recognizer can either be of the hard (binary) decision type, in which case G(n,m) є {0, 1}or a soft one, in which case G(n,m)є [0, 1].

So we can approximate G(n,m) with

$$F(n,m) = \sum_i F_i^m \left[u(n - a_i^m) - u(n - b_i^m)\right] \tag{2}$$

where u(n) is the unit step function [ai^m, bi^m] is the i-th interval that contains the face of the m-th person.

For each person Sm, her signature triplets (F^m i , ai^m , bi^m), i = 1, . . . ,N form a pulse series in the video time domain, as can be seen in fig. 1(a). An example of such a signature signal is given in fig. 1(b). Each quartet corresponds to a unique face appearance. The number of quartets in a video is equal to $\Sigma^{M}{}_{m\text{-}1}$gm<< N×M, where gm is the number of appearances of person sm in the video, and N and M are the total numbers of frames and persons in the video.

The extraction of the signature from the video is straightforward if a face detection and recognition module is available. 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. This similar to applying a median filter to each person specific pulse series in the signature (Fig. (a)).

B. Signature Search

Let us assume two signatures F$_1$(n,m) and F$_2$(n,m), derived as per Equation (2), which are extracted from two video segments and refer to a common set of faces S. Let us assume that we move F$_2$ by a specific displacement d. We will define as co-occurrence C the evidence that the two signatures are the same. At a specific frame n for a specific person m and in the case of a binary decision recognizer, such evidence exists if and only if the person exists at both signatures, i.e. Chard(d, n,m) = F$_1$(n,m) · F$_2$(n + d,m). If the detector produces a detection certainty, the evidence that a specific person occurs in both signatures depends on the certainty of detection. Since the evidence of co-existence is only as good as the worst recognition certainty of the two signatures, in this case Csoft(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 over all frames and persons. If the lengths of the two video segments are N1 and N2, and assuming without loss of generality that N1≤ N2, C can be regularized by dividing by N1 and the number of possible persons, M. In the case of a hard detector (whose output is 0 or 1) this corresponds to:

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

In the case of a detector that produces detection certainties, we have:

$$C_{soft}(d) = \sum_{n=1}^{N_1}\sum_{m=1}^{M} \frac{\min(F_1(n,m), F_2(n+d,m))}{N_1 M} \tag{4}$$

Geometrically, C can be visualized as 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-occurrence Cmax = maxd C(d), obtained when sliding one signature with respect to the other. The computation of C(d) and Cmax is similar to the computation of a convolution between the two face signature signals. Thus Cmax tends to be insensitive to small changes in the signature, such as splits, shifts, changes in height or in width of the quartet rectangles — corresponding to errors in face detection and recognition. 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 infeasible. Thus we have developed an algorithm that does this in near-logarithmic time with respect to the size of the database [9]. This is achieved by indexing the database temporally and on person identity, and exploiting the properties of the signature in order to quickly compute the optimal matching location.

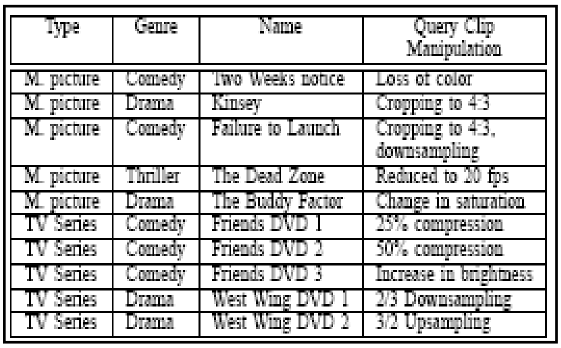
In essence, what our algorithm does is compute the above specific signature similarity metric in appropriate locations in the database. The novelty is not so much in the selection of appropriate locations, as in the efficient computation of signature similarity, specifically in finding the maxima of the similarity metric. There is ample work in the database literature regarding retrieval of intersecting intervals [10], and also for combining temporal coincidence with the matching of other attributes, e.g. [11]. These methods by themselves are neither substitutes for our algorithm nor comparable to it, since they do not address the specific problem of quickly computing our own specific similarity function.

C. Extension to Other Types of Semantic Signatures

The signature format and the matching algorithm that were described above, are not limited to the case of faces. It is possible to use them also for other types of features, related to object or person identity, as long as these features have the
following properties:

1. Refer to a type of object (e.g. person) that is common in the video
2. Correspond to a large variety of distinct identities of such objects or persons

Table 1 : Motion Pictures and Television Series Annotated with Face Appearances



Type	Genre	Name	Query Clip Manipulation
M. picture	Comedy	Two Weeks notice	Loss of color
M. picture	Drama	Kinsey	Cropping to 4:3
M. picture	Comedy	Failure to Launch	Cropping to 4:3, downsampling
M. picture	Thriller	The Dead Zone	Reduced to 20 fps
M. picture	Drama	The Buddy Factor	Change in saturation
TV Series	Comedy	Friends DVD 1	25% compression
TV Series	Comedy	Friends DVD 2	50% compression
TV Series	Comedy	Friends DVD 3	Increase in brightness
TV Series	Drama	West Wing DVD 1	2/3 Downsampling
TV Series	Drama	West Wing DVD 2	3/2 Upsampling

3. They can be formulated as a pulse series, with pulses which are a few seconds long
4. Cannot exist twice in the same frame for the same identity,
5. (Optionally) have a value that indicates certainty or significance
6. Can be extracted automatically. One such example would be speaker identities, as detected by a speaker recognizer.

III. Experimental Results

We implemented a real system to demonstrate the practicality of our method for the retrieval of video segments. This was a complete system, taking raw video as its input, with the only user interaction being the training of the face recognizer with a small sample of appropriately labeled faces.

A. Derivation of Experimental Data

A video corpus was first selected, consisting of approximately 8 hours of motion pictures, and approximately 8 hours of television programming. The motion pictures came from different genres (comedy, drama, thrillers), while the television programming consisted of several episodes of one drama and one comedy series, as described in Table I. Video resolution and aspect ratio varied, but was mostly in DVD format. Face detection and recognition was then performed on this corpus. The results of the face detection and recognition were then processed to create continuous tracks of person appearances, and inserted into a database as quartets. Motion picture 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, sanitized corpora such as news broadcasts mostly contain frontal, frontally illuminated and emotionally neutral faces, in specific formal attire and hairstyle. In order to perform face detection and recognition we utilized the FaceVACS toolkit, produced by Cognitec Systems GmbH . FaceVACS is considered to be very close to the state of the art in the field, as far as real-world systems are concerned. It was a participant in the Face Recognition Vendor Test 2002 , where it had ranked first in most tasks, and very close to the top in the rest. The choice of a commercial product was a conscious choice as it provided a standardized procedure for face detection and recognition, with little need for parameterization.

It should be noted that, due to the commercial nature of FaceVACS, there is a lack of technical detail regarding the algorithms that are incorporated in the toolkit. In short, face detection and recognition is performed by FaceVACS in the following 4 steps:

1. Face and Eye Localization

The image is taken in multiple scales and, for each scale, all locations in the image are checked for similarity to a face. If the similarity in a specific location is high enough, all appropriate locations within the face are checked for the existence of eyes. The most appropriate locations are considered to be the locations of the eyes. It is possible that no eyes are found, in which case this face is removed from further consideration.

2. Normalization and Preprocessing

The face region is checked for noise or blur, and if they are too high, the face is rejected. Then the face is geometrically normalized so that the eyes are in pre-specified positions. It is additionally normalized with respect to luminance and frequency content (i.e. ”edginess”).

3. Feature Extraction

A vector of visual features is extracted from the image, in a predefined manner optimal for distinguishing people. This vector is then subject to a subspace transform in order to maximize discriminance, producing a final feature vector.

4. Comparison of Features with Reference Set

The feature vector computed in the above steps is compared with all feature vectors in the reference set, i.e. the set of features corresponding to the list of persons that the module is trained to recognize. Those reference vectors exhibiting a similarity above a certain threshold are given as matches.

So, FaceVACS returns as its output the following four pieces of information (of which only the last three are utilized by our system):

- The location of the face in the image (given as a rotated rectangle).
- The location of the eyes in the image (given as two coordinate pairs).
- The identities of the top three matches that exceed the similarity threshold (given as up to three identities).
- The certainties of each match, which are essentially the similarity scores between the two vectors (given as numbers between 0 and 1).

The reference set for a specific person is constructed by performing face detection and recognition, as above, on a number of reference images. The set of features thus extracted is then clustered in order to extract up to 5 representative feature sets, which are considered to comprise the reference set for the person the module is trained to recognize. This procedure is called enrollment.

In our case the reference sets were constructed from a small number of images in the videos in question, approximately 25 per person. This number was necessary in order to compensate for changes in illumination, posture, facial expression and other factors. The persons that were chosen to be the targets of recognition, and consequently the basis of indexing, were the main actors in the motion picture or TV series. In general these were the actors that appeared in the starting credits, and ranged from 5 to 10 per motion picture or mini series. A total of 54 distinct persons were chosen, some appearing in more than one video.

Having obtained our video database and the gallery of persons that would be detected and recognized in it, we then proceeded to perform face detection and recognition using FaceVACS. For performance reasons only 5 frames per second were processed but this was adequate for granting the algorithm a good retrieval performance. For a subset of the video, the faces detected and identified by the algorithm were compared with an annotation that we have manually constructed for the faces in the video. In total, of all faces in all video frames, over 30% were correctly detected and identified. This includes faces that were very small, were looking at the opposite direction from the camera etc. In view of the great variety that was exhibited by the data, such a result is very good. In order to exploit the temporal continuity between frames, a procedure that greatly increases the performance by means of a voting scheme that rejects outliers and reinforces detections having a high certainty has been implemented. This procedure is described below.

As already mentioned, in order for the retrieval algorithm to function, it needs to have quartets as input. Thus the per frame face detection and recognition results given by FaceVACS have to be converted to quartets. This is done in three stages. First, the detected faces are unified into tracks using their spatio-temporal proximity. Then a single identity is determined for each track by a voting scheme that uses the recognition scores of the frames in each track. Finally the certainty of the recognition is computed. In more detail:

(i). If two detected eye pairs have the following properties, they are considered to be in the same track:

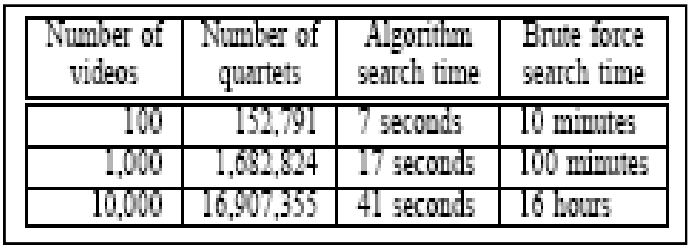
- They are less than one second apart.
- The ratio of inter-ocular distances of the two pairs is less than 2.
- The distance of the centers of the eye pairs is less that the average of their inter-ocular distances.

(ii). For each track, the detected faces corresponding to a specific identity are separated, and their certainties (i.e. recognition similarities) are summed. The track is then identified as belonging to the identity (person) with the greatest certainty sum. Obviously, each such track corresponds to a quartet related to a person. The quartets with a certainty sum below a threshold of 1 are deleted.

(iii). The certainty of the quartet is computed as the above certainty sum divided by the temporal duration of the track (in frames). The start and end frames of the quartet are obviously the start and end frames of the track.

Two sets of experiments were performed. One that involved the implementation on a real video database as described above, including face detection and recognition, and one for assessing computational performance using an artificial dataset of variable size.

Table 2: Average Search Time Result



Number of videos	Number of quartets	Algorithm search time	Brute force search 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

B. Retrieval Performance

The database with respect to which the retrieval would take place was constructed as described in section III-A. Then we selected 40 clips from the database, each having a duration of 2.5 minutes. These constituted approximately 10% of the total database size. An alteration, such as change of compression, change of resolution, cropping, change of frame rate, and conversion to grey scale was performed on each of them. The alterations that were performed on the query clips are described in Table I. Face detection and recognition was then performed on them, using the union of all reference sets used in the database as a basis for recognition. Then each query segment was searched in the database (i.e. retrieval was performed). The result for the whole database was a correct retrieval score of 90%, which verifies the effectiveness of our algorithm.

C. Computational Performance

In order to evaluate the computational performance of our algorithm, we created artificial video signature databases of different sizes using a model that simulates the quantitative characteristics of our experimental data. Each database consisted of 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 an 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. Using computer significantly behind the state of the art (Pentium 4 at 2.4 GHz), the average times of retrieval are given in Table II. As it can be seen, the performance of the algorithm is near-logarithmic with respect to the size of the database. 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 added 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.

IV. Conclusions

A method for performing fast retrieval 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. The retrieval performance of our algorithm has been verified by the implementation of a real system that uses face detection and recognition to index real videos. The results show that the proposed method performs very satisfactorily, both in terms of computational search efficiency (even in a database of 10000 hours of video), and in terms of retrieval errors. In general the method proves that face related information carries enough discriminant power to be used for video indexing and retrieval. The proposed face-based approach could be adapted, in order to index video using the appearances of persons derived from other modalities or even the appearances of other classes of objects that possess distinct identities.

References

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

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

[3] 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.

[4] M. Viswanathan, H. Beigi, A. Tritschler, 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.

[5] G. Wei, I. K. Sethi,"Omni-face detection for video/image content description", Proceedings of the 2000 ACM workshops on Multimedia, Nov 2000.

[6] Y. N. S. Satoh, T. Kanade, "Name-it: Naming and detecting faces in news videos", IEEE MultiMedia, Vol. 6, No. 1, pp. 22–35, 1999.

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

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

[9] C. Cotsaces, N. Nikolaidis, I. Pitas,"Video indexing by face occurrence-based signatures", in ICASSP, Vol. 2, 2006, pp. 137–140.

[10] B. Salzberg, V. J.Tsotras,"Comparison of access methods for timeevolving data", ACM Computing Survey, Vol. 31, No. 2, pp. 158–221, 1999.

[11] H.-P. Kriegel, M. Ptke, T. Seidl, Managing intervals efficiently in object-relational databases", in VLDB, 2000, pp. 407–418.

M. NAGARAJU received his B.Tech degree in Computer Science and Engg. from Malineni Lakshmaiah Engineering College, Prakasam, AP, India in 2006, the M.Tech degree in Computer Science and Engineering from JNTUK, Kakinada, AP, India, in 2011. He was a teaching assistant Professor with Department of Informatio Technology from 2011 to Till date. His research interests include digital signal processing.