ABSTRACT

Content-based video retrieval technology holds the key to the efficient management and sharing of video content from different sources, in different scales, across different platforms, and shared over different communication channels. In this work we present a fast retrieval algorithm based on matching the geometry of video sequence traces in the principal component space. Techniques to address scale (spatial and temporal) issues, as well as, noise and other possible distortions, such as frame dropping, are discussed. Experimental results demonstrate the effectiveness of the proposed approach.

1. INTRODUCTION

With the proliferation of digital video capturing, storage and communication devices, the amount of information in video form is growing rapidly in personal entertainment, security, and military applications. To effectively share and manage video content presents a technical challenge to the existing information management systems. Semantic features based management systems require substantial amount of manual labeling of the content and are therefore in general not practical.

Consider the following example representing an application addressed by this work. A mobile phone user has just watched a low visual quality (e.g., QCIF size, 10fps), short (e.g., 5 sec) segment of a soccer game from some unknown source. S/he wants to now watch the complete game in SDTV format from her personal soccer game video collection, or some content provider’s collections. The system will therefore need to search a database based on this 5-sec segment and return the locations of the full size program, if it exists. The semantic information is clearly not present in the querying segment. The matching has thus to be “content-based”. In addition, the variance in temporal and spatial scale, as well as, the noise and distortion incurred during the communication must also be addressed.

Content-based retrieval approaches have been investigated extensively by many researchers [1]-[4], [7], [9]-[12]. Such approaches are typically based on the visual features of video frames and a similarity metric defined on these features. Visual features are typically high dimensional and the commonly used are color, shape, texture, and motion. Drawbacks of such approaches are the computational expense associated with the extraction and matching of visual features, and the fact that the video sequence is treated as a collection of images and the collective temporal behavior of the sequence is typically not addressed. The retrieval performance can also be negatively affected by the scale variance, noise, and quantization distortion of the video content.

In the proposed approach video sequences are viewed as temporal traces in some high dimensional space. Each video frame is reduced to a point in its Principal Component (PC) space [5][8] of much lower dimension. The trace over time of a video sequence in this space should provide sufficient information to differentiate it from other sequences. In the PC space, the matching of sequences becomes a problem of matching the geometry of the traces; when the dimensionality of the PC space is small, this matching as well as indexing can be done efficiently. Implementation details are described to further speed up the retrieval and also address spatial and temporal resolution differences between the query and the database video.

The paper is organized into the following sections. In section 2 we present the method for computing the trace of a video sequence in its PC space and the matching method. In section 3 we discuss implementation issues, in section 4 we present simulation results, and in section 5 we draw conclusions and outline our future work.

2. PRINCIPAL COMPONENT SPACE TRACE AND MATCHING METRICS

2.1. Scaling and Principal Component Space Projection

Let \( n \) denote the dimensionality of a frame, i.e., a video frame \( f_j \) belongs to \( \mathbb{R}^n \). Principal Component Analysis (PCA) [8] finds an \( nxd \) transformation \( V_{ds} \) with \( d \)
orthogonal unit n×1 vectors, that maps the frames of the video sequence \( f_j \) to a low \( d \)-dimensional (\( d<n \)) Principal Component space, that is,
\[
x_j^d = V_d^T f_j
\]
(1)
where \( x_j^d \) is an \( d \times 1 \) vector. \( V_d \) is found according to,
\[
V_d = \min_V \sum_{j=1}^{N} \left\| (f_j - f_0) - VV^T (f_j - f_0) \right\|^2,
\]
(2)
where \( f_0 \) is the average of the frames observed. Notice that \( V \) is sample dependent and its accurate computation requires correct modeling of the covariance of the sample frames with a large number of samples.

For a sequences with frame width \( W \) and height \( H \), the original data dimension \( n=W\times H \), can be quite high, and the amount of available data is typically not adequate in accurately performing PCA. Therefore we would like to reduce the frame size to a desirable scale \( w \) first, with minimum information loss possible, before the PCA. This is typically done via the wavelet scaling.

For \( w=8 \), \( h=6 \) and \( d=2 \), the PCA basis vectors and eigen-values are shown in Fig. 1. Notice that the first 4 components captured most energy of the sample frames.

![Figure 1. Principal component basis vectors and eigen-values](image)

With \( d=2 \), frames are reduced to feature points in the 2D PC space and the trace of video sequences can be visualized. Examples of such trace for the “foreman” and a certain “mixed” sequences are illustrated in Fig. 2. The “mixed” sequence consists of 50 various video clips of 60 frames each. This is a very compact representation compared with other image features like color, shape and texture features.

2.2. Matching Metrics

The traces of different video clips occupy different areas in the 2D space, and have different trace geometry, as shown in Fig.2.

Let \( l_j \) denote the differential trace of an \( m \)-frame sequence. As an example, the differential trace for the “foreman” sequence is shown in Fig.3, for \( d=2 \) and \( d=4 \) cases. Based on Fig. 3 and additional data, it appears that the differential trace is relatively invariant with respect to the dimensionality of PCA for \( d > 2 \). This is because most energy is captured by the first 2 dimensions.

Let us denote by \( l_j \) the distance between two differential traces \( l^a \) and \( l^b \) of length \( m \). For example, if the \( L_2 \) norm is used, is computed as,
\[
d(l^a, l^b) = \sqrt{\sum_{j=1}^{m} (l^a_j - l^b_j)^2}
\]
(6)

![Figure 2. Sequence traces in the 1st-2nd principal component space](image)

For a \( q \)-frame querying video clip with PC feature representation \( Q={x_1, x_2,..., x_q} \), and a \( p \)-frame video collection with feature representation \( D={y_1, y_2,..., y_p} \), with \( p>q \), let the distance between \( Q \) and sub segment of \( D \) starting at \( k \) be,
\[
d^k_0 = \left( \sum_{j=1}^{q} (x_j - y_{k+j-1})^2 \right)^{1/2}
\]
(3)
Let the minimum distance between \( Q \) and \( D \) be \( d^k_0 = \min_{k \in [1, p-q+1]} d^k_0 \). If the video collection contains the query clip, then \( d^0 = 0 \). However due to different frame sizes, quantization scales when dealing with compression, and errors/noise in a communication application, we declare the query clip exists if the minimum distance is below certain threshold \( d_0 \), and the location is determined by,
\[
k^* = \arg \min_{k \in [1, p-q+1]} d^k_0
\]
(4)
Alternatively, a scalar feature of a trace, the differential trace step, can be used in matching. It is defined by
\[
l_j = \begin{cases} 0, & \text{if } j = 1 \\ \left| x_j - x_{j-1} \right|, & \text{if } j > 1 \end{cases}
\]
(5)
Let \( L = \{ l_1, l_2,..., l_w \} \) denote the differential trace of an \( m \)-frame sequence. As an example, the differential trace for the “foreman” sequence is shown in Fig.3, for \( d=2 \) and \( d=4 \) cases. Based on Fig. 3 and additional data, it appears that the differential trace is relatively invariant with respect to the dimensionality of PCA for \( d > 2 \). This is because most energy is captured by the first 2 dimensions.

Let us denote by \( d(L^a, L^b) \) the distance between two differential traces \( L^a \) and \( L^b \) of length \( m \). For example, if the \( L_2 \) norm is used, is computed as,
Figure 3. “foreman” seq trace step length plots
The differential trace can now be used as a matching metric for retrieval. That is, for an m-frame querying video clip, with differential trace \( L^q\), its best match is found in a database according to

\[
k^* = \arg \min_k d(L^q - L_k^b) \tag{7}
\]

where \( L_k^b = [t_k^0, t_{k+1}^0, \cdots, t_{k+m-1}^0] \) is the partial differential trace of a database sequence of length \( m \) starting at time instance \( k \). Compared with the retrieval method in (3), this differential trace based method places more emphasis on the temporal behavior of the sequence, than the location and geometry. It seems to be more appropriate to use this method when the query clips are long.

The search performance can be further improved by efficient indexing. Because of the relatively low dimensionality of the feature space, an \( R^* \) tree [13] like indexing structure can be employed to store the features for each video shot.

3. IMPLEMENTATION CONSIDERATIONS

As mentioned in section 1, the querying video clip very often is of different spatial and temporal resolution than the clips in the database. To address the spatial resolution incompatibility (plus additive noise and quantization error issues) we scale the frames to a common spatial resolution by low-pass filtering and down-sampling both the querying and the database sequences. Typical common resolutions used are 8x6, 12x9 and 16x12. This scaling process can also improve the accuracy of the PCA process (2) with limited samples, since it reduced the data space dimension.

The differences in temporal resolution between querying and database sequences can be addressed by pre-computing or computing on the fly the traces and differential traces of the sequences in the database at different frame rates, like for example, 10fps, 15fps, 20fps, 25fps and 30fps. For a given frame rate in the query clip, we just match the database features that have the same timestamp offset with the querying clips.

A related issue to the differences in temporal resolution is when there are random frame drops, due, for example, to transmission errors. Assuming the frame timestamp is present, then for the matching method in (3), we simply ignore the frames that are missing. The distance is now defined as

\[
d^k_q = (\sum_{j=1} h_j (x_j - y_{k+j-1})^2)^{1/2} \tag{8}
\]

where \( h_j \) is zero if frame \( j \) is missing in the query clip, otherwise \( h_j=1 \).

For the differential trace based matching, the missing frames need to be interpolated. We perform linear interpolation in the PC space. Although more sophisticated interpolation methods could be employed, we experimentally found out that linear interpolation is adequate for retrieval purposes.

4. SIMULATION RESULTS

In our simulations the PCA is done based on 3200 randomly selected video clips from various collections. We experimented with scales 4x3, 8x6, 16x12, 32x24, and found out that the scale of 8x6 offers the best compromise between energy compaction and amount of information loss.

To demonstrate the effectiveness of the proposed method, we set up a video collection of 3000 frames from 50 different clips. We then set up 2500 positive queries with different clip length and locations from the collections. We also set up 500 negative queries from clips that do not exist in collections.

When there is no noise and no random frame drops, the retrieval achieves 100% accuracy for all 2500 positive queries and 500 negative queries, with query lengths of 8, 12, 16 and 20 frames for both geometry matching and differential trace matching methods.

Selected noise free retrieval results are illustrated in Fig. 5a. We have four 20-frame queries created from “tennis”, “flower”, “susie” and “foreman” sequences. Among them only the “foreman” query does not exist in the collections.
5. CONCLUSION AND FUTURE WORKS

In this paper we presented a new content-based video shot retrieval solution. The video frames are reduced to points in low (2~4) dimensional space and the retrieval is based on matching the location and geometry of the sequence trace. Our solution is fast and robust to noise, distortions and differences in spatial and temporal resolutions between querying and database sequences. The low dimensionality of the proposed feature also makes efficient indexing possible. The proposed solution can be useful in a wide range of practical applications that require real time response to video queries.

6. REFERENCES