ABSTRACT

This paper presents a system architecture and the appropriate algorithms for confidential searching of multimedia digital libraries. The proposed scheme uses Middleware service layer that allows pre-processing of raw content with technology owned by the Search Engine, without compromising the security of the original architecture in any way. The specific search algorithms described are a hierarchical graph structure algorithm for preprocessing, and a backtracking search algorithm that achieves good real-time performance (speed, and precision-recall values) under the given security constraints.

1. INTRODUCTION

Multimedia digital libraries are emerging at an increasingly fast rate throughout the world. There now exist commercial digital archives containing several tens of millions of hours of films and video recordings. This vast amount of multimedia information requires new methods and tools that allow quick searching, indexing and retrieval of audio-visual information in a secure and trusted environment. Efficient and secure searching in multimedia digital libraries requires a) pattern recognition technologies for analyzing and describing content under a semantic framework, b) knowledge engineering algorithms for quick searching of annotated multimedia content, and c) computer science protocols and architectures for protecting intellectual property rights of both content and search technologies. In this paper, we address the latter two of these issues, assuming a known multimedia description scheme stemming from the MPEG-7 standard. Other description schemes can be also used.

The traditional way of representing a video, as a sequence of consecutive frames, (each of which corresponds to a constant time interval, e.g., 40 ms for the PAL system), while being adequate to “play” a video file in a movie mode, is not appropriate for new multimedia services, such as searching, retrieving and mining of video content over distributed multimedia platforms.

Sequential searching is a time consuming process since video archives contain enormous amounts of information. Thus, searching algorithms for multimedia content require nonlinear, normally hierarchical organisation schemes. Hierarchical video organisation is supported by the MPEG-7 standard [1] through the HierarchicalSummary Description scheme [2]. The MPEG-7 standard aims to provide a framework for multimedia description to support efficient searching, retrieval and mining of audio-visual content. The standard provides an XML schema for encoding hierarchical video organisation tools. It also suggests an algorithm for the implementation of a hierarchical video summarisation scheme, by extracting key-frames and then clustering the remaining video frames according to the key-frames’ visual content and the temporal distance between the key-frames and the remaining video frames [2].

In the VideoZoom [3] prototype, video frames are linearly decomposed in space and frequency domains to allow for fast and precise video browsing. VideoZoom is based on a linear spatiotemporal video decomposition. Content-based video organisation is achieved in [4] by hierarchically analyzing a video sequence into different content resolution levels, using a tree structure representation, the levels of which indicate the respective content resolution, while the nodes correspond to the segments into which the video is partitioned at this level. Based on this scheme, the user is able to select segments (tree-nodes) of interest and reject segments of non-interest, resulting in a multi-resolution interactive video browsing schema. In [5], a video abstract is proposed while a pictorial summary of the multimedia content is discussed in [6]. Finally, the cluster validity method is presented in [7] for hierarchical video content summarization.

While the aforementioned approaches are suitable for video browsing applications, they cannot be directly
applied for video searching, mining and retrieving scenarios since the algorithms are performed interactively, with a user guiding the process. In particular video content is traced by the user’s interaction and not in an automatic way, as it is required for the secure searching and mining schemas.

Figure 1: The proposed hierarchical graph structure for multimedia content representation.

In the context of this paper, we propose a platform, which can support multimedia searching where both preprocessing and real-time searching are performed in an automatic way, i.e., without any user interaction. In particular, we propose a preprocessing scheme which uses a hierarchical graph structure able to organize video data into different content resolution levels, resulting in a pyramidal hierarchy from the coarsest (lowest) to finest (highest) content resolution. Instead of the tree structure organization of [4], in the presented work the nodes are associated with each other, resulting in a pyramidal graph representation. This extension is the innovation that allows for automatic tracing of multimedia content, enabling offline, non-interactive execution. In contrast, in the tree representation of [4], video content is traced by the user’s interactions, making the scheme suitable only for video browsing applications.

We propose a new algorithm for real-time multimedia search, based on the preceding preprocessing scheme. The algorithm comprises two different phases: forward search and backtracking. In the forward search, the precomputed hierarchical graph is examined to find best matched nodes with respect to the user’s query. In the backtracking phase, different alternative paths of the hierarchical graph are returned in order to increase retrieval precision.

2. HIERARCHICAL GRAPH REPRESENTATION OF THE MULTIMEDIA CONTENT

To enable multimedia queries, we need to describe the rich media content through the use of appropriate multimedia metadata. Multimedia metadata are extracted by the application of audio-visual signal processing algorithms and stored in separate databases in Middleware layer. When a specific query is to be executed, these metadata are exploited to get the retrievals instead of the raw data. Multimedia metadata are offline estimated. In the proposed scheme the extracted multimedia metadata are organized in hierarchical graph structures to allow for the application of real-time search algorithms.

2.1. Hierarchical Multimedia Content Representation

In this section, we describe the algorithm that is used during the offline operation phase to generate the multimedia metadata. The proposed scheme used to non-linearly organize the multimedia content in a way that supports quick search. This hierarchy is encoded using the XML schema of the MPEG-7 standard to guarantee interoperability and universal accessibility. Note that this algorithm is executed entirely noninteractively instead of the previous approaches such as the work of [4].

In this paper, a hierarchically structured graph is adopted to nonlinearly organise the content of a video file. This is an important enhancement of the tree structure adopted in [4] since it allows automatic search without user’s interaction. The depth of the graph indicates
different content resolution levels starting from the coarsest (lowest) and ending with the finest (highest) resolution. Three different resolution levels are adopted in the presented approach; the shot representative level, the shot level and the frame representative level. At the shot representative level, the content of a video file is projected on the key-shots space. This means that at this level, the nodes of the graph correspond to the shot representatives of a video file. Similar content organisation is accomplished at the third resolution level (frame representative level) with the difference that video content is projected on the key-frames instead of the key-shots. Therefore, the graph nodes of the third resolution level are the frame representatives. Finally, at the second resolution level (shot level), video content is represented by the shot information. Shots are nonlinearly organised with respect to the shot representatives. At each resolution level, links are assigned from one node to another, which indicate the degree of association between the connected nodes. In this way, a node is related with other nodes of similar content characteristics but at the same content resolution level. Figure 1 presents the proposed three-level hierarchically structured graph adopted for representing video content in a nonlinear hierarchical way.

Shot and frame representatives are extracted by minimising a cross correlation criterion so that the ones that correspond to the most “uncorrelated” content are defined as shot/frame representatives. Instead, the shot level is constructed by applying a clustering algorithm.

A) Shot/Frame Representative Level Construction

In this section, we describe the way of extracting shot and frame representatives to construct the graph nodes of the first and third resolution levels.

Let us denote as $\mathbf{f}_i$ the vector, the elements of which correspond to the features exacted either for the $i^{th}$ frame or the $i^{th}$ shot of a video sequence. Let us also denote as $K$ the number of frame or shot representatives being adequate to describe the rich content fluctuation. The number $K$ can be estimated as in [4] so that the “difficulty”, measured as the number of frames that must be processed before finding content that fulfills the user requirements, is minimised.

Let us now denote as $\mathbf{x} = [x_1, x_2, \ldots, x_K]^T$ a vector, the elements of which refer to the indices of the $K$ representatives. These indices are found in this paper by minimising a cross correlation criterion, as

$$\hat{x} = \arg \min_{x} E(x)$$

$$E(x) = \frac{2}{K(K-1)} \sum_{i=1}^{K} \sum_{j=i+1}^{K} \rho(f_{x_i}, f_{x_j})^2$$  \hspace{1cm} (1)

where $\rho(f_{x_i}, f_{x_j})$ is the correlation coefficient of feature vectors $f_{x_i}$ and $f_{x_j}$ at the indices $x_i$ and $x_j$ respectively, while $\hat{x}$ gives the optimal index vector. Equation (1) indicates that the representatives should be as much as possible uncorrelated with each other.

The complexity of an exhaustive search for estimating the minimum value of (1) would be unreasonably great, since all possible combinations of frames would need to be examined. For this reason, minimisation of (1) can be applied by the use of a genetic algorithm as in [4]. This scheme is able to find a solution close to the optimal one within a small number of iterations.

B) Shot Level Construction

Having estimated the most representative shots and frames within a video sequence, the following step of the proposed approach is to construct the shot classes so that the second level of the proposed video hierarchy is developed.

Let us now assume that the $K$ most uncorrelated shots have been estimated using the aforementioned algorithm, that is the optimal index vector $\hat{x}$. Then, the $S_k$ with $k=1, 2, \ldots, K$ shot classes are constructed as

$$S_k = \{s_i : i \in Z(\hat{x}_k)\}, \text{ for all } i$$  \hspace{1cm} (2)

where $Z(\hat{x}_k)$ refers to the influence zone of the index $\hat{x}_k$ (i.e., the $k^{th}$ shot representative) and $s_i$ to a video shot. The influence zone is defined as a set which contains all shot indices, whose respective shot feature vector is closer to the feature vector of the representative shot defined by the index $\hat{x}_k$ than all the other representative shots, that is

$$Z(\hat{x}_k) = \{\forall i : \rho(f_{\hat{x}_k}, f_{x_i}) > \rho(f_{\hat{x}_k}, f_{x_m}) \} \forall m \in \{1, 2, \ldots, K\} \text{ and } m \neq k$$ \hspace{1cm} (3)

Equation (3) means that all shots that fall within the same influence zone of a shot representative are considered as members of the same class.

C) Node Association

In our scheme, instead of the approach of [4], each node of the graph is associated with the other nodes of the same level with similar visual content characteristics. Such an association yields a hierarchically structured graph representation scheme which in the sequel significantly increases the mining and searching efficiency. Modification of the algorithm presented in [4] to support association among nodes of the same content resolution level is performed to allow for automatic searching. Instead, the method of [4] is oriented only for interactive video navigation that demands user feedback. The
necessity of node association is described in Section 3, in which we present the searching algorithm.

Node association is performed using the correlation coefficient \( \rho() \) between nodes that are to be linked. More specifically, let us denote as \( n_i(k) \), and \( n_j(k) \) two nodes of the hierarchical graph at the \( k^{th} \) content resolution level. Then, between these two nodes a link is assigned with similarity degree the cross correlation of the feature vectors related with the content of these nodes, i.e.,

\[
d(n_i(k),n_j(k)) = \rho(f_{n_i(k)},f_{n_j(k)}) \tag{4}
\]

where \( d(n_i(k),n_j(k)) \) expresses the similarity degree among the nodes \( n_i(k) \), and \( n_j(k) \), while \( f_{n_i(k)},f_{n_j(k)} \) indicate the respective feature vector of the nodes \( n_i(k) \), and \( n_j(k) \).

The similarity degrees of a node are sorted in descending order so that the first links point out for nodes of the most similar content with respect to the reference node. As a result, in case that searching succeeds at some nodes at a given level, the searching can then follow nodes of similar content to get more relevant results to the user.

In particular, let us denote as \( \text{Assoc}(n_i(k)) \) a set which contains the sorted associate nodes of the node \( n_i(k) \). Thus, the \( j^{th} \) associate \( a_j(n_i(k)) \in \text{Assoc}(n_i(k)) \) refers to the \( j^{th} \) most relevant node with respect to \( n_i(k) \). Consequently, \( a_0(n_i(k)) \) is the best associate node of \( n_i(k) \) based on the similarity degree described in equation (4).

### 3. REAL TIME SEARCH

In this section, we describe a non-sequential content navigation algorithm that can be used for the online, real-time search phase of the system. The proposed algorithm exploits the hierarchically structured graph described in Section 2, which nonlinearly organises video content. The process is fully automatic and is characterised by small computational complexity compared to sequential (linear) searching, while simultaneously achieving high precision-recall performance.

The proposed automatic searching algorithm is divided into two main parts: forward searching and backtracking.

#### 3.1. Forward Searching

The purpose of the forward searching is to seek all nodes beneath a node given as input in the process and navigate towards relevant video content. The input node of the forward searching process is given from the backtracking process as described in Section 3.2 (but see Section 3.3 for initialization and termination of the algorithm). In our approach, we assume that the relevant content is located only at the third resolution level, that is at the key-frame representatives, so that only references to the most detailed content level are returned to the user as a final result (the other levels do serve for navigating the content efficiently, of course).

The forward searching process is divided into two main procedures: best node selection and node retrieval.

**A) Best Node Selection**

The best node selection procedure takes as input a node \( n \) and the respective \( k \) content resolution level at which node \( n \) is located. The purpose of this procedure is to seek all nodes beneath node \( n \) in order to return the node that best matches the user’s query and has not been previously selected. In particular, let us denote as \( \text{Child}(n) \) the set which contains all the children of \( n \). Variable \( c_i \in \text{Child}(n) \) expresses the \( i^{th} \) child of \( n \). Then, the best node at the following \( k+1 \) resolution is the child of \( n \) that best matches to the user’s query and has not been previously returned.

\[
\hat{c} = \arg \max_{c_i \in \text{Child}(n), c_i \in R} \{ \rho(f_q, f_{c_i}) \} \tag{5}
\]

where \( R \) is a set which contains all the processed nodes for a given user’s query. Variable \( \hat{c} \) refers to the best matched child, while \( f_q \) and \( f_{c_i} \) to the query feature vector and the feature vector of the child \( c_i \). We recall that \( \rho() \) is the correlation coefficient among the two feature vectors.

Having selected the best matched child, the process iteratively proceeds until a) either the last content resolution level is reached (i.e., \( k=3 \)) or b) all children have been previously returned or there are no children for the examined node. At the termination of the procedure the best matched node, say \( \hat{n} \) is returned.

The following table summarizes the main step of the best node selection at the next iteration.

**B) Node Retrieval**

This procedure is activated in case that \( k=3 \), i.e., the best node selection procedure reaches the key-frame representative level. The procedure takes as input the best matched node \( \hat{n} \), as provided from the best node selection procedure and returns as output the modified set \( R \) by adding in it a set of nodes that are considered relevant to the user’s query.

More specifically, the best matched node \( \hat{n} \) is included in the set \( R \). Apart from this node, set \( R \) is enhanced with
those associate nodes of \( \hat{n} \) which satisfy the following equation

\[
\gamma_r \cdot \rho(f_q, f_{a_r(\hat{n})}) > \rho(f_q, f_{\phi(\hat{n})}) \text{ and } a_r(\hat{n}) \notin R
\]  

where \( a_r(\hat{n}) \) is the \( r \)th best sorted associate of the node \( \hat{n} \) and \( \phi(\hat{n}) \) the respective father node. Vectors \( f_q, f_{a_r(\hat{n})} \) and \( f_{\phi(\hat{n})} \) indicate the feature vectors of the query, \( a_r(\hat{n}) \) node and the father node of \( \hat{n} \), i.e., the \( \phi(\hat{n}) \). Variable \( \gamma_r \) scales the correlation according to the order of the associate nodes. In our approach, \( \gamma_r \) increases exponentially with respect to the order \( r \)

\[
\gamma_r = \beta^r, \quad \beta > 1
\]  

Equation (7) means that all the associate nodes \( a_r(\hat{n}) \) of \( \hat{n} \), whose correlation coefficient between the query and the node \( a_r(\hat{n}) \) (scaled by a factor \( \gamma_r \) exponentially proportional to the rank \( r \) of the associate node) is greater than the respective correlation coefficient between the query and the father node \( \phi(\hat{n}) \) of \( \hat{n} \), are included in the set \( R \), containing the relevant nodes of the query. This is performed only for those nodes that have not been previously selected as relevant in set \( R \).

3.2. Backtracking

After the completion of the forward searching process, backtracking is activated to examine other possible paths (see Section 3.3 for initialization and termination of the algorithm). The backtracking procedure takes as input the father of the node \( \hat{n} \), i.e., the \( \phi(\hat{n}) \), as obtained from the termination of the BestNodeSelection procedure. The algorithm initially finds the first associate node of \( \phi(\hat{n}) \) that has not been previously selected (i.e., it does not belong to the set \( R \)), that is

\[
\alpha = a_{r_0}(\phi(\hat{n})) \quad (8)
\]

with

\[
r_0 = \arg \min_{r} a_r(\phi(\hat{n})) \notin R \quad (9)
\]

Let us denote as \( D \) the difference of the correlation coefficient between the query and node \( \alpha \) from the coefficient between the query and the father of \( \alpha \), i.e., \( \phi(\alpha) \),

\[
D = \rho(f_q, f_{\alpha}) - \rho(f_q, f_{\phi(\alpha)}) \quad (10)
\]

In case that \( D \geq 0 \) the backtracking process terminates and the forward searching is activated with input the node \( \alpha \) along with the respective content resolution level. On the contrary, in case of \( D < 0 \), indicating that the father of node \( \alpha \) is more relevant to the user’s query than node \( \alpha \), another backtracking is activating with input the \( \phi(\alpha) \).

The process is terminated whether either \( D \geq 0 \) or the root of the graph is reached.

Equation (10) can not be calculated in case that node \( \alpha \) is a root node, since the father \( \phi(\alpha) \) does not exist. This case means that another graph is examined, i.e., another video file. This file is the first associate of the current examined video file. In this scenario, the forward searching is directly activated with input the root of the new graph, that is the \( \alpha \).

3.3. Initialization and Termination

The search algorithm starts by activating the Best Node Selection procedure of the forward searching. In particular, all the graph roots of video files belonging to the same category of the query are examined and the one that best matches the query is initially selected for further searching. Let us denote as \( n_k(0), i=1,2,... \) the root nodes of graphs for video files belonging to the same category as the query one. Then, the best matched root node is found as the one which maximizes the

\[
n_k(0) = \arg \max_{n(0)} \rho(f_q, f_{n(0)}) \quad (11)
\]

The node \( n_k(0) \) along with the respective zero content resolution level are used as input in the BestNodeSelection(\( n_k(0),0 \)) for commencing the search algorithm.

The search algorithm terminates in case that the number of elements of the processed frames in \( R \) reaches a maximum number of data, that is \( |R| = N \), where \( N \) is the number of processed frames. The best K of these (\( K \leq N \)), either based on the correlation coefficient itself or on the results of the per-frame processing, are delivered as actual search hits.

4. EXPERIMENTAL RESULTS

In order to evaluate the efficiency of the proposed hierarchical tree-structure video representation for multimedia content searching, and to compare it with other approaches presented in the literature, objective quality criteria need to be introduced. A series of objective criteria are adopted in this paper for evaluating search performance. The criteria are the Precision-Recall curve, the Average Normalised Modified Retrieval Rank (ANMRR), and the Search Efficiency Ratio [8][9][10].
To evaluate the efficiency of the proposed scheme, we have used multimedia database consisting of 30 video files, each of average duration of 2 hours. The experiment is conducted by submitting 3,000 randomly selected queries and then computing the aforementioned described the objective criteria. The proposed scheme is initially compared with the traditional linear (sequential) search, which is currently the most popular multimedia search algorithm. We also present comparisons with other nonlinear organisation schemes, such as the works of [3][4][5][6] and [7]. It should be clarified, however, that the approach of [4] assumes a user’s interaction in decomposing the visual data and thus it can not be directly applied in the case where automatic video organisation is required. For this reason, we compare the proposed approach with these methods under the SER ratio, which does not require precision accuracy values.

Figure 2. Comparison of precision values of the proposed scheme with the linear (sequential) case, for varying proportions of examined data, expressed as percentage of the total data $U$. (a) Recall values of 5%, 10%, and 20%. (b) Recall values of 30%, 40%, and 50%. (c) Recall values of 60%, 70%, and 80%.

Figure 2 presents the precision values versus the number of examined frames (data) $U$ in the database. In this case, we have omitted the subscripts $s$ and $n$, since the experiments have been conducted for the same value of $U$ both for the sequential and the proposed search approach. In addition, in Figure 2 variable $U$ is expressed as percentage of the total number of data in the database for convenience. The experiment has been conducted for different recall values and for factor $\beta = 1.1$ [see equation (7)].

These results show that, in terms of precision, our system very slightly outperforms the sequential case as long as the number of examined frames $U$ is greater than the recall target. For values of $U$ lower than the targeted recall, an almost linear fall of the precision values for the linear approach occurs whereas our system suffers only small precision degradation. This is explained as follows. Let us assume that the content is uniformly distributed across all data of the database (an assumption that perfectly fits the random location of the data). Then, we can see that we are able to find an adequate number of relevant items, as long as $U$ is greater than the recall value, both for the proposed and for the linear case. This means that we examine a sufficient quantity of data in the database to reach the recall target. On the other hand, in case that the number of examined data becomes less than the recall target, we can see that the examined number of samples is not on average adequate to reach the recall value for the linear case leading to a significant deterioration of the precision performance. In contrast, the performance of the proposed method remains very robust since the algorithm navigates the database in regions that present high probability of locating relevant data. Thus, it can successfully discover content even if it can only examine a small proportion of the total data available. The rapid decrease for the linear case is more evident for high recall values. Note that the best performance achievable is in any case limited from above by the intrinsic level of difficulty at the individual frame level of extracting discriminating features from the content of interest; this means that the exact values shown in these figures are application-specific and thus, as far as evaluating the performance of our system is concerned, arbitrary – it is the comparison between the results of our method and those of the linear case that is important.
Recall Precision

U=90%, Linear Case
U=90%, The Proposed Case

U=80%, Linear Case
U=80%, The Proposed Case

U=70%, Linear Case
U=70%, The Proposed Case

U=60%, Linear Case
U=60%, The Proposed Case

U=50%, Linear Case
U=50%, The Proposed Case

U=40%, Linear Case
U=40%, The Proposed Case

U=30%, Linear Case
U=30%, The Proposed Case

U=20%, Linear Case
U=20%, The Proposed Case

U=10%, Linear Case
U=10%, The Proposed Case

(a) (b) (c)

Figure 3. Comparison of the precision-recall values of the proposed scheme and of the linear (sequential) case. (a) $U=90\%$, 80\%, and 70\%. (b) $U=60\%$, 50\%, and 40\%. (c) $U=30\%$, 20\%, and 10\%.

The conclusions drawn above are also supported by the results shown in Figure 3, where we have plotted the precision-recall curve for different values of $U$ both for the proposed method and for the linear approach. It was again clear that, in the linear case, the precision accuracy drops abruptly when $U$ exceeds the recall values. On the other hand, the precision accuracy in the proposed scheme remains robust regardless of the ARe and $U$ values. Once again, factor $\beta = 1.1$ is selected.

Figure 4. Performance comparison of the ANMRR values of the proposed scheme with the linear (sequential) case versus the proportion of examined data, expressed as percentage of the total data $U$.

The efficiency of the proposed scheme as far as the ANMRR values are concerned is depicted in Figure 4 versus the number of examined data $U$. This figure compares the results obtained from the proposed and the sequential case. The figure demonstrates that in the proposed scheme the ANMRR values remain relatively stable regardless of $U$. On the contrary, in the linear (sequential) case the ANMRR values drop sufficiently only as the proportion of examined data $U$ increases considerably.

Figure 5. The Search efficiency ratio (SER) versus the number of Relevant Retrievals. (a) A comparison with the methods of [2],[4],[7]. (b) A comparison with the methods of [3],[5],[6].

The results obtained using the Search Efficiency Ratio (SER), as defined in equation (20) for different non-linear approaches are presented in Figure 5. The SER criterion can be used to compare the performance of other non-linear search methods presented in the literature than the proposed one. In particular, Figure 5(a) compares the performance of the proposed approach versus the number of retrievals being considered as relevant for the methods of [2],[4],[7], while Figure 5(b) for the methods of [3],[5],[6]. We can conclude that the presented search methods reaches the performance of the [4] though the latter is an interactive method that a user guides the process, while the proposed one is an automatic approach. For all the other compared approaches, interactive or not, the presented one yields better performance, revealing its efficiency.

Finally, the results obtained at given values of the number of retrievals using the SER criterion for all the above mentioned compared approaches are shown in Table I. As is observed, the proposed video hierarchy provides a significant reduction of the difficulty in accessing frames of interest compared to sequential scanning (about 79 times for the first relevant retrieval). In this table, we have also compared the performance of the
proposed algorithm with other hierarchical approaches for video content decomposition and navigation, despite the fact that these approaches can not be used for multimedia data mining.

Table I: The SER ratio of the proposed scheme compared with other approaches.

<table>
<thead>
<tr>
<th>Nonlinear Video Representation Algorithms</th>
<th>SER (1&lt;sup&gt;st&lt;/sup&gt; Retrieval)</th>
<th>SER (5&lt;sup&gt;th&lt;/sup&gt; Retrievals)</th>
<th>SER (15&lt;sup&gt;th&lt;/sup&gt; Retrievals)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Proposed Scheme</td>
<td>79.60</td>
<td>100.32</td>
<td>110.56</td>
</tr>
<tr>
<td>The Method of [4]</td>
<td>87.20</td>
<td>108.84</td>
<td>121.35</td>
</tr>
<tr>
<td>(MPEG-7)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The Method of [2]</td>
<td>67.42</td>
<td>84.63</td>
<td>91.36</td>
</tr>
<tr>
<td>The Method of [5]</td>
<td>20.18</td>
<td>27.80</td>
<td>31.92</td>
</tr>
<tr>
<td>The Method of [3]</td>
<td>42.30</td>
<td>53.07</td>
<td>56.02</td>
</tr>
<tr>
<td>The Method of [6]</td>
<td>24.20</td>
<td>30.50</td>
<td>34.33</td>
</tr>
</tbody>
</table>

In Table I we also compare the proposed scheme to the semi-automatic method of [4], and the performance of the latter is in fact better. This is due to the fact that the work of [4] is semi-automatic, requiring user feedback for decomposing video content.

5. CONCLUSIONS

In the proposed architecture, a non-linear organization of multimedia data is presented that allows the direct implementation of an automatic non-linear search module. The algorithm, instead of the previous work of [4], which uses a tree-based decomposition of the data, exploits the application of a pyramidal graph that allows search to be applied directly on the content domain.

An automatic non-linear search algorithm is applied in the following to exploit the advantages of the proposed graph-structure hierarchical media content organization.

Experimental results using a series of objective criteria as well as comparisons with other approaches have been proposed to demonstrate the efficiency of the presented scheme than previously published works. The results have been obtained on a very large media database.

6. REFERENCES