社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－DBS－134　(I)　(7)
2004／7／13

動画MPEG-7に対する索引アルゴリズム

朱 成敏 †　遠山 元道 ‡

† 慶應義塾大学大学院 理工学研究科 開放環境科学専攻
‡ 慶應義塾大学 理工学部 情報工学科
E-mail: † joo@db.ics.keio.ac.jp, ‡ toyama@ics.keio.ac.jp

ISO/MPEG グループが提案した XML に基づく MPEG-7 はマルチメディアコンテンツを記述するメタデータの標準になった。MPEG-7 はユーザが必要とするマルチメディアデータに付随する多種多様なメタデータを記述しているため、データの量が膨大になる。MPEG-7 は XML に基づいた文書なので、データの量が増加するほど、XML パーサによる処理が増え、検索に多大な時間を要する結果となってしまう。そこで、本研究では MPEG-7 文書の検索を高速化るため MPEG-7 が持っている情報の特徴を利用し、距離計算を通じて効率的な索引アルゴリズムを提案する。

キーワード：MPEG-7 、 XML 、 retrieval

An Efficient Indexing Algorithm for MPEG-7 Documents

Sungmin JOO †　Motomichi TOYAMA ‡

†School of Science for OPEN and Environmental Systems,
Faculty of Science and Technology, Keio University.
‡Department of Information and Computer Science, Faculty of Science and Technology,
Keio University.
E-mail : †joo@db.ics.keio.ac.jp ‡toyama@ics.keio.ac.jp

MPEG-7 is an ISO/IEC standard being developed by MPEG, and aims to create a standard for describing the multimedia content data that will support various multimedia search and retrieval applications. However, since it is described with text-based XML, it is very hard to efficiently search the documents on multimedia contents describe in MPEG-7 if the size of the documents an efficient indexing algorithm on MPEG-7 documents on video data. It first analyzes the some requirements on MPEG-7 documents indexing mechanism, and secondly proposes an indexing algorithm that clusters the Visual Descriptor values in MPEG-7 documents into several levels according to their distributions. This clustering helps to reduce the search range by just comparing the Visual Descriptor values of query video with the pre-computed similar Visual Descriptor values of target videos. This thesis also experimentally shows that the performance of the indexing algorithm is dependent on the number and granularity of the levels, and their values could be computed easily. The proposed indexing algorithm could be used to develop a large video archive system in which the contents on multimedia are described in MPEG-7.

keyword : MPEG-7 、 XML 、 retrieval

1 Introduction

Technologies related to the Internet and computers enabled the development of various multimedia, which in turn enabled to service and use multimedia contents in various ways. As audio and video information are being produced in large quantities, however, there are growing demands for meta-data media for users to search, exchange and share information more easily and accurately.

In response to such demands, XML-based MPEG-7[1] was proposed, which is a standard for the structure and meta-data of audio/video data. MPEG-7 describes various pieces of information for content-based retrieval of multi-media contents. Because descriptors in MPEG-7 includes information about the production of multimedia contents such as title and genre, high-level information based on 5W/1H and low-level information such as color, movement, the volume of sound.

Thus it can contain most information for the media contents. As the quantity of information increases, however, the volume of description also expands, which makes search difficult. Like XML-based search, MPEG-7-based search has the same problems as those in text-based data. That is, when a query was made in an information storage space composed of a lot of segments, it takes a long time to search information because all documents had to be referred to.

This study proposes an indexing algorithm for efficient search of MPEG-7 stored in the form of XML and analyzes its usability through an experiment. For the purpose, this study used a method of creating an index by clustering the similarities of the values of MPEG-7 Visual Descriptor in a storage space. This method uses each descriptor as a value representing a document that describes a segment, so it is advantageous in that it shortens search time by reducing the range of search from the entire documents to part of them through an index. When the method was used in actual search, it significantly reduced the range of search and was efficient in terms of time because it search the reduced range. In addition, the result of search using an index was the same as that of ordinary search. Thus the use of an index did not increase the error rate.

The structure of this paper is as follows. Section 2 reviews previous researches on MPEG-7, and Section 3 explains an algorithm for efficient MPEG-7 search using an index. Section 4 explains a search system based on the algorithm proposed in Section 3 and presents the results of an experiment. Lastly, Section 5 describes the strong and weak points of the algorithm proposed in this study and suggests the direction of future research.

2 Related Works

The scope of researches on the search of MPEG-7 documents is not clear. Researches on efficient access to documents are being made on the assumption that researches on XML search will be successful. On the other hand, researches on XML are mainly focused on the use of XML as a means of information exchange such as the application of XML documents to existing RDBMS systems and the introduction of RDBMS for effective inquiry process on XML documents. After all, researches on access method for MPEG-7 documents are not active. Thus it is necessary to study the characteristics which are different from those of general XML documents.

This section examines situations to be considered for efficient indexing for MPEG-7 documents and looks for methods for resolving these situations.

2.1 Requirement Analysis

As in Figure 1, there are many kind of descriptors in MPEG-7 and each descriptor stores various values. It will be difficult to search for a desired document in a storage space where a large number of MPEG-7 documents exist in order to

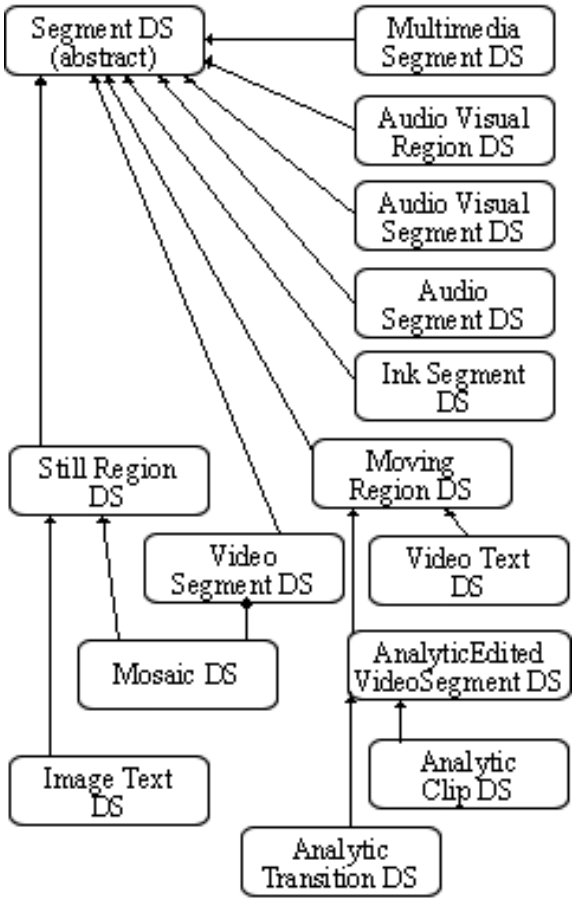

図 1: MPEG-7 Segment DS

process users' queries quickly. Thus this section looks into requirements and design conditions for an indexing technique for the efficient search of a large number of MPEG-7 documents. First, when a query is made, similar results should be produced through an algorithm of comparing all documents in the storage space with The MPEG-7 eXperimentation Model[2] and, for this, a great number of calculations have to be made. This study tries to solve the problem of the large number of calculations using an index. In the situation that we do not know what queries will be made, it is necessary to consider how to prepare the index for efficient search when actual queries are made. he second problem is how to create a value representing a segment composed of several descriptors, which are obtained by calculation using different comparison algorithms for each type of descriptor. Because MPEG-7 designates a segment as the basic unit of a medium, if a segment is described with 8 descriptors, the similarity values of the descriptors are calculated using 8 comparison algorithms. When a MPEG-7 document is compared with other documents using the 8 similarity value, however, it is a delicate problem what criteria should be used in determining the similarity between them.

Another consideration is that, in using MPEG-7 descriptors, the indexing technique must be independent from each descriptor. MPEG-7 has various descriptors and they are used differently according to users' purpose. Thus if an index is dependent on the characteristic of a specific descriptor, each descriptor has to have its corresponding index and algorithm and the system is affected by whether a descriptor is used or not. That is, when an index is applicable to all types of descriptors regardless of their nature, namely, be flexible, it can be used consistently whether a new descriptor is added or an existing one is deleted.

If the three problems mentioned above are solved, the indexing technique will be definitely efficient for MPEG-7. The next section examines an indexing technique to solve these problems.

2.2 Index Structure and Design

This section examines solutions for problems listed in Section 3.1 and look for a method of designing an index that reflects all the solutions.

2.2.1 Index Structure

The MPEG-7 eXperimentation Model presents the method of calculating similarity calculation for comparing each descriptor. However, the similarity is for each descriptor, not for a segment containing several descriptors. Thus it is necessary to consider a method of comparing segments using the similarity of each descriptor. First, let's dis-

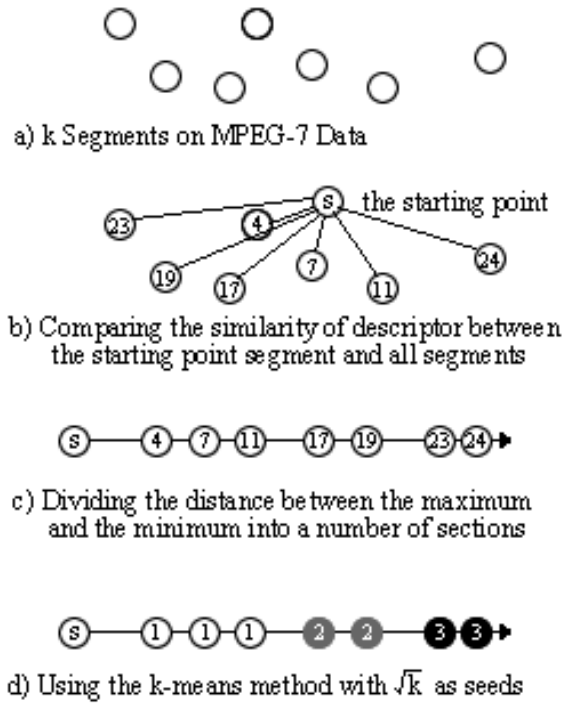

図 2: Clustering Segments by descriptor according to similarity

cuss how to build an index through pre-processing. A segment entered as a query should be compared with segments in the storage using the comparison algorithm of The MPEG-7 eXperimentation Model, but it takes a long time to compare the query segment with a large number of segments in the storage one by one using the comparison algorithm. In addition, it is impossible to calculate in advance because we do not know what segment users make queries on.

Then the following method may be suggested. Calculate the similarity between segment A and B. Again, calculate the similarity between segment C and A. Compare the similarity between segment A and B and that between C and A to obtain distance between B and A as well as between C and A. Based on these, the distance between segment B and C can be obtained without separate calculation through the relative distances of them from the starting point segment A. That is, relative distances between segments can be converted into absolute distances. In this way, an arbitrary segment is taken as the starting point and the distances between the segment and all the other segments within the group are built into an index in advance. When a segment is given as a query, the distance between the segment and the starting point segment is calculated, and all segments with a similar distance are those similar to the query segment. In this way, when a query is made, it is not necessary to calculate similarities to all segments. Instead, the similarity between the query segment and an arbitrary segment is calculated, and it is compared with pre-processed similarities to find corresponding segments to search. With this method, the problem of pro-processing can be solved when we do not know what segment will be queried. What should be considered here, however, is the location of the starting point segment. If the distance between the starting point segment A and segment B is -2, but the result of the *the distance* $|SegmentA - SegmentB|_d$ is 2, and *the distance* $|SegmentA - SegmentC|_d$ is 3. Using the equation above, *the distance* $|SegmentB - SegmentC|_d$ is $|-3| - |-2|$, namely, 1, but their actual distance is $|(-3) - (-2)|$, namely, 5. This problem can be solved by taking an absolute starting point as the starting point segment. That is, if descriptors in a segment have their minimum value and the segment is not larger than any other segment, it can be used as an absolute starting point. Accordingly, this study suggests defining a base document with descriptors of the minimum value in order to calculate distances for building an index. The second problem to consider is how to compare segments using individual descriptors. Each segment has one or more descriptors. In addition, the similarity of a segment may be determined by all of its descriptors of some of them as demanded by users. Thus, for a segment, the similarity of each descriptor has to be stored. For this, this study suggests the following method. In order to build an index, cluster by descriptor according to sim-

ilarity first. Organize groups based on the result of clustering. Criteria for organizing groups are as follows. First, for each descriptor, find the maximum and the minimum distance from the base document with descriptors of the least value. Divide the distance between the maximum and the minimum into a number of sections and use the k-means method with the boundary values of the sections as seeds. Use the square root of the number of the whole segments as the value for division. When k segments are clustered based on the similarity of a certain descriptor, $\sqrt{k}$ become the seed value. As the segments are clustered through k-means method using the seed, $\sqrt{k}$ groups are organized. Each segment group clustered in this way has an identifier. That is, if a segment has n descriptors, it has n identifiers. As a result, the index stores information on each segment and its identifiers. Identifiers are from 0 to $\sqrt{k}$ according to the closeness of similarity.

2.2.2 Index Design

Queries in this study are on segments. Thus the representative frame of a segment produces results for each MPEG-7 descriptor through the standard experimentation model suggested for MPEG-7, and creates a document describing the segment by combining outputs for each descriptor. An index is built using these documents, and queries are made through MPEG-7 documents describing query segments. The output of a query is the corresponding document. In order for MPEG-7 search to be implemented through indexing, the location of the document in the storage is calculated and the identifier of the clustered group corresponding to the location is fetched. In addition to information on each segment, it is necessary to record the size of the storage in the index to determine the size of the storage. The index file is created as a XML-based document. If an index file is represented in a hierarchical structure, <Index> has child tags <IndexInfo> and as in Figure 3.

```
<complexType name=Index final=#all>
 <complexContent>
  <element name=IndexInfo>
   <attribute name=DistanceOfDescriptor1 type=floa
   <attribute name=MaxFlagofDescriptor1 type=inte
   <! - - for each Descriptors - ->
  </element>
  <element name=Segment>
   <attribute name=flag type=integer>
   <! - - for each segments - ->
  </element>
 </complexContent>
</element>
```

図 3: An example of subtree update

<IndexInfo> has as its attributes the file name of the base segment, the similarity distance of each descriptor and the number of groups clustered. When a query document is put it, similarity to the base segment specified in <IndexInfo> is calculated through the comparison algorithm of the MPEG-7 eXperimentation Model. When the distance calculated, it is determined first which group the descriptor of the query segment belongs to, and the identifier of the group is obtained. In this way, each descriptor finds its corresponding identifier and the index is searched with the identifiers. includes the name of the comparison file and, as its attribute, indicates the distance of descriptors contained in the file.

2.3 MPEG-7 search through index

This section discusses search using the index suggested in the previous section. Figure 4 shows the process of MPEG-7 search through index. A query segment can calculate the space of segments for each descriptor in the storage through <IndexInfo> of the index. The locations of the query segment are calculated and identifiers corresponding to the locations are obtained. Then segments with the

same identifier as that of the query segment are retrieved in through Xpath and the segments are used as the range of search. Of course, each descriptor is independent and users can choose a descriptor to use for search.

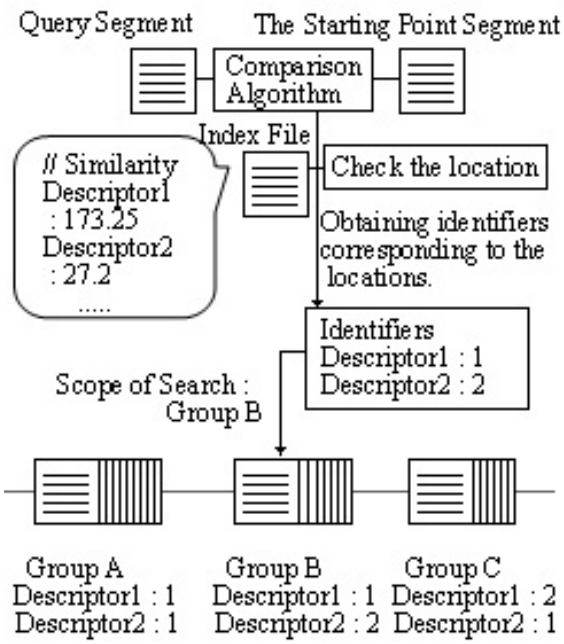

図 4: MPEG-7 search through Index

3 Experimentation

Based on the algorithm proposed in Section 3, this study implemented indexing for efficient MPEG-7 search as well as search through indexing as the final results of this study.

3.1 Experiments

In order to implement the algorithm of this study, the author used IIS5.0-based ASP on Windows 2000 and MSXML as a XML parser. Thus, this study implemented a system that can perform not only search but also indexing on the Web. Interface developed for the experiment is divided into the part of search and the part of indexing. Also, users can enter their queries by specifying values for one or more descriptors using OR and AND operands, and entered queries are converted into Xpath queries for process. Query documents are parsed and information in the index file is calculated. Using the calculated results, similar segments are searched from segment information within the index file. The number of segments used in the experiment was 1863, which are sampled arbitrarily from movies, dramas and sports.

3.2 Performance Results

The verification of the efficiency of the index-based search algorithm proposed was focused on accuracy and the reduction of search time. Here, accuracy means that the result of search using the index should be identical to that of sequential search without using an index. In addition, the reduction of search time means that search through the index should be faster than ordinary sequential search. In this experiment, an arbitrary query segment is provided and the result of sequential search based on similarity was compared with that of index-based search. Table 1 shows the result

表 1: Performance Results

	range	sequential(sec)	index-based(sec)
1	167	16.6402	0.3437
2	190	15.8414	0.2656
3	173	16.3593	0.2656
4	312	15.8434	0.3593
5	418	15.7343	0.4843
6	286	15	0.3283
7	341	15.8437	0.3906
8	221	15.8125	0.2968
9	192	15.5625	0.2812
10	153	15.3125	0.2656

of ind ex-based search and that of ordinary se-

quential search for the same query segment. In order to enhance reliability, search was made 12 times and results excluding the best one and the worst one were used. The average range of index-based search was 245.3 segments, which is only 8% of the entire 1863 segments. The average time taken for search was 15.7950 seconds for sequential search, while 0.3281 second for index-based search, which is only 2% of the former. This is because index-based search does not parse all segment documents but parse only a necessary range of documents, which is determined by parsing the index document. Thus if efficiency is evaluated based on the range and time of search, the results above show that the index-based search is definitely more efficient than ordinary sequential search.

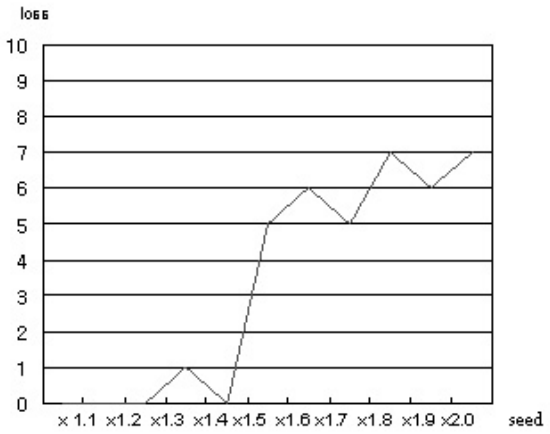

図 5: The Average Loss

Next, let's discuss accuracy. In case of the k-means method, which was used as a method of clustering, the result of search proposed in this study is different according to the initial seed value. The initial seed value was obtained using the square root of the number of the whole segments. It is because, when searching a total of n elements, it is most effective to divide the elements into $\sqrt{n}$ groups with $\sqrt{n}$ elements. For example, segment A is close to the query segment but they may be classified into different groups and use different identifiers as a result of the k-means method, causing loss in search. Thus this study measured the loss by adjusting the seed value at random. Figure 6 is a graph representing the average loss resulting from the increase of the seed value. As the graph shows, the seed value $\sqrt{n}$ as proposed worked as significantly convincing seed without loss.

4 Conclusions

Because MPEG-7 proposed as a standard for multimedia meta-data is XML-based, namely, is described in text, it takes a long time to search a desired document through sequential comparison if the number of documents is large. It is because MPEG-7 is a XML-based document. This problem has not been studied much because researches on XML query process have been expected to bring improvement in the matter. On the contrary, there have been active researches on fast XML document search using RDBMS. Consequently there have been few researches on fast MPEG-7 search. To solve this problem, this study described the characteristics of a MPEG-7 document using the similarity of its descriptors and recorded them into the index. By analyzing a query document and mapping the result to documents in storage through the index, this study shortened search time. According to the result of an actual experiment, the range of search was reduced to about 16The method suggested in this study used the characteristics of MPEG-7 documents, which are different from general XML documents. A query suggested in this paper is a segment and the result of search is also a segment. Thus the algorithm proposed in this study may be a useful algorithm for query-by-example, which is being actively used for content-based retrieval. Documents utilized in this study used low-level descriptors such as color and texture in search. If it becomes possible to search using high-level information combining scene description based on a performer's name or 5W1H, the system may be more accurate and satisfacto-

rily meet users' demands.

参考文献

[1] M.Martinez, *Overview of the MPEG-7 Standard,* ISO/IEC JTC1/SC29/N4032, March 2001.

[2] Akio Yamada, Mark Pickering, Sylvie Jeannin, Leszek Cieplinski, and Jens, *MPEG-7 Visual part of eXperimentation Model Version 9.0,* ISO/IEC JTC1/SC29/WG11/N3914, January 2001.

[3] L. Cieplinski, W. Kim, J.-R. Ohm, M. Pickering, and A. Yamada, *MPEG-7 Visual part of eXperimentation Model Version 12.0,* ISO/IEC JTC1/SC29/WG11/N4548, December 2001.

[4] Quanzhong Li, and Bongki Moon, "Indexing and Querying XML Data for Regular Path Expressions," *Proc. of the 27th International Conference on Very Large Data Base,* pp. 361-370, 2001.

[5]] Sophie Cluet, Pierangelo Veltri, and Dan Vodislav, "Views in a Large Scale XML Repository," *Proc. of the 27th International Conference on Very Large Data Base,* , pp. 271-280, 2001.

[6] Naoko Kosugi, Yuichi Nishimura, Seiichi Kon'ya, Masashi Yamamuro, and Kazuhiro Kushima, "Let's Search for Songs by Humming!," *Proc. of ACM Multimedia 99,*pp.194, 1999.

[7] P. Beek, B. Ana, *Text of 15938-5 FCD Information Technology - Multimedia Content Description Interface - Pard 5 Multimedia Description Schemes,*, ISO/IEC JTC1/SC29/WG11/N3966, March 2001.

[8] Jerome
H. Friedman, *Data Mining and Statistics: What's the Connection?,* http://www-stat.stanford.edu/ jhf/, November 1997.

[9] Naoko Kosugi, Yuichi Nishimura, Seiichi Kon'ya, Masashi Yamamuro, and Kazuhiro Kushima, "Let's Search for Songs by Humming!," *Proc. of ACM Multimedia 99,* pp.194, 1999.

[10] A.B. Benitez, M. Beigi, and S.F. Chang, " Using relevance feedback in content based Image Meta-search Engine," *IEEE Internet computing v2 n4 ,* July-August 1998.

[11] Jongho Nang, Jinguk Jeong, Myung-hwan Ha, Byunghee Jung, Kyeongsoo Kim , "An Authoring Tool Generating Various Video Abstraction Semi-Automatically," *Lecture Notes in Computer Science,* pp.311-318 Vol. 2532 Spinger-Verlag, December 2002 .

[12] Jorma Laaksonen, Markus Koskelam, and Erkki Oja, "PicSOM-Self-Organizing Image Retireval With MPEG-7 Content Descriptors," *IEEE TRANSACTIONS ON NEURAL NETWORKS,* pp.841-853 Vol. 13, JULY 2002.

[13] J.A. Lay and L.Guan, "Processing Models for Distributed Content based Multimedia Retrieval Systems," *Proc. of the 5th International/National Conference on digital image computing techniques and applications,* 1999 .

[14] Myunghwan Ha, Byunghee Jung, Kyeongsoo Kim, and Jongho Nang , "On-Demand Video Summary System ," *Proc. of International Workshop on Advanced Image Technology,* pp.223-228, January 2002.