映像検索におけるパッケージセグメントモデルと応用アプリケーション

Package-Segment Model for Movie Retrieval System and Adaptable Applications

國枝　孝之[*,**]　　脇田　由喜[*,**]
Takayuki KUNIEDA　　Yuki WAKITA

要　　旨

マルチメディアコンテンツの論理構造を表現するための方法としてパッケージセグメントモデルを提案し，これを利用して映像コンテンツの検索・実験システムを実装した．このデータモデルの備える枠組みと論理構造表現の柔軟性について説明する．更にこれをマルチメディアデータ管理に応用し，マルチメディアコンテンツの検索インデックスとしての有効性を示す．

ABSTRACT

The Package-Segment Model is proposed so as to verify a representation method for multimedia content logical structure. The Package-Segment Model consists of various objects that are defined as structure components and involved in the construction processes. Experimental retrieval system of the contents indexed by this Package-Segment Model is carried out, and the Model has representation flexibility for object framework and adaptable retrieval mechanism. It is applicable to integrate into various multimedia contents management system and automatic indexing and retrieval of the contents.

* (株)次世代情報放送システム研究所 Information Broadcasting Laboratories, Inc.[***]
(株)リコーより(株)次世代情報放送システム研究所へ兼任出向中．
** 画像システム事業本部　ソフトウェア技術開発センター Software Technology Development Center, Imaging System Business Group
*** (株)次世代情報放送システム研究所は，基盤技術研究促進センターの事業のひとつとして，基盤技術研究促進センター，ソニー(株)，松下電器産業(株)，(株)リコー，日本放送協会，日本テレビ放送網(株)，(株)フジテレビジョンの出資により平成9年2月に設立されたもので，次世代の放送「情報放送(IB: Information Broadcasting)」の研究開発を行っている．

1. Introduction

Multimedia data are flooding and there is a big demand for multimedia archiving systems. There are several activities to standardize content-based retrieval schemes [1]. Commencement of digital satellite broadcasting will bring about TV program explosion. Today's progress in storage technology and CPU power allow easy browsing of such multimedia contents. Moreover today's many database management systems are multimedia capable [2][3]. However there are only a few easy ways to search and select contents by user friendly interfaces. Most existing stream data (video data) have no structured information as plain text. Along with evolution of Internet web technology, many kinds of structure description languages have been proposed and have come into wide use recently (e.g. HTML, XML). On the other hand, video and audio stream data remain unchanged. We propose to treat those contents as semi-structured data. In this paper, we propose structural formula and methods for treating streaming data. The proposed scheme of the structure is based on a tree structure and represents the logical structure of multimedia content itself such as movie script. This structure is used as index of retrieval and is able to detect individual scenes. We call this structure Package-Segment Model (PS-Model). Discussions on the framework, components of the structure and verification of effectiveness are presented below. Lastly we introduce an experimental application developed.

2. The PS-Model and Structure Framework

Many problems associated with existing movie retrieval systems can be attributed to reliance on unstructured representations. Some construction schemes of multimedia contents have already been proposed [4][5]. To represent the logical structure of multimedia contents, we provide a flexible framework of structure description. In the PS-Model, a segment represents a cut, scene or logical duration in the content stream. Package plays a role in generalizing a segment set. The PS-Model structure is generally similar to a tree structure. In this tree structure, each node can have an attribute node that maintains some attributes of the parent node and a user-defined supplement. The procedure of structuring is as follows. In the first step, the root node is generated to represent the whole content stream. This node has one segment set which is unified by a package. In other words, the root node has a package node that has a segment representing the whole stream. In the second step, a dividing/distributing module creates some segments under the top segment (whole content) which are unified by a package describing the requirement and significance of the new segments. After constructing the skeleton of the content, in the third step, if necessary, an extracting module detects some key frames and/or key sounds (audio part) as nodes under a target segment.

2-1 Package-Segment Model Structure

Two sample streams are shown here to help understanding. Assume these two streams are news programs. In the sample strategy, those programs are categorized into following factors:

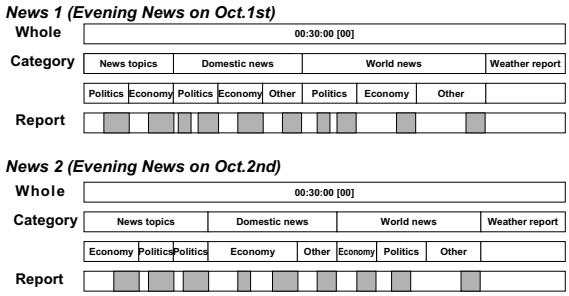

Fig.1 Two sample content structures which are news programs

1. 1st layer is rough classification by category (news topics, domestic news, world news, weather report, etc.).
2. 2nd layer is fine classification by segment subject under the above category (politics, economy, other events).
3. 3rd layer is to separate between the recorded news part and the live announcement part. Key frames are detected from the recorded part.

Fig.2 shows the PS-Model structure of an example stream

"News1". The intervention of MoviePackage object makes it possible to represent various scenarios and multiple scenarios can be contained in one structure. For example, content producers add two or more interpretations of their programs. Although logical structures are different for the same content, the intervention of MoviePackage object makes it possible to represent the content in one structure. The stream can be divided into segments in multiple ways depending on different points of view. The MoviePackage objects can represent each of the segmentation methods. In this example, the MoviePackage object has a MovieAttribute object that represents the classification category of each segment. MovieFrame objects, that are extracted as key frames from a "reported part" have some image features as a MovieAttribute object.

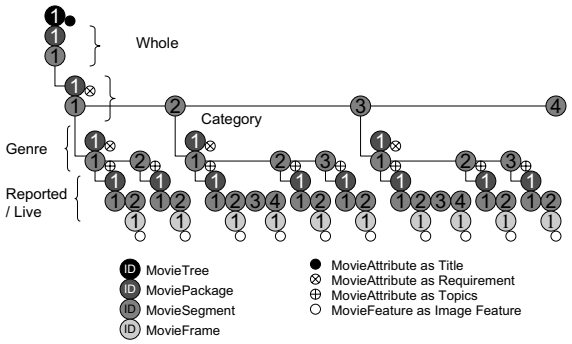

Fig.2 The PS-Model structure of a sample content "News1". The PS-Model is characterized by the intervention of the MoviePackage object.

2-2 Object Components and Behavior

The PS-Model is composed of various classes that are classified by the functions described below. Italic items are base classes.

A) Structure Definition Classes: MovieTree, MoviePackage, MovieSegment, MovieFrame, MovieSound, *MovieAttribute*, and *MovieFeature*

These classes represent the content structure as a node. The MovieSegment class represents a part of the logical structure in the content stream. The MoviePackage class plays an important role in representing the structure. The requirements of MoviePackage are aggregation of segments and representation of multiple scenarios, interpretation and so on.

B) Content Mediation Classes: MovieMapper, MovieMemory, *MovieStream*, MovieTimeCode, and *MovieCodec*

These classes mediate between the logical structure and physical structure. Video content is encoded in various formats (e.g. MPEG, AVI, etc.). The *MovieStream* class to be specialized, for example, as a content data format such as a MovieMpeg1SystemStream object, hides the MPEG-1 physical structure and protocol. These classes provide logical access methods for the PS-Model that do not depend on the stream encoding scheme.

C) View Definition Classes: MovieView and MovieLink

MovieView class represents the browsing scheme, and MovieLink class instructs an internal/external link point. Browsing is required to review a search result where several candidates have been retrieved. The MovieView class has MovieLink class which points to a node inside or outside and has the supplement that enables representation of the behavior of playing contents.

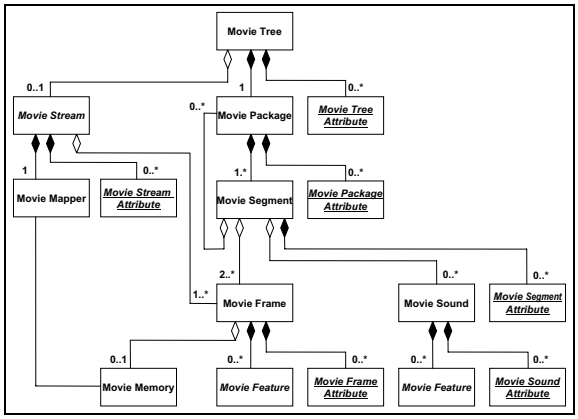

Fig.3 Class diagram for the PS-Model. Italic items are base classes.

D) Type Definition Classes: MovieTime, MovieRange, *MovieOption*, MovieImage, and MovieWave

E) Operation Classes: *MovieDivider*, *MovieDistributor*, *MovieDetector*, *MovieExtractor*, and *MovieArranger*

These operation classes are base classes and need to be

specialized as required in order to form the PS-Model structure and extract features. A specialized class can inherit some logical functions and create new tasks on those objects as completed ones. Typical functions and behaviors for each class are explained below.

・*MovieDivider* & *MovieDistributor*

These objects divide/distribute segments and generate new package-segment sets under the target segments. These functions can be overridden by segmentation and construction algorithms. In the most typical case, a *MovieDivider* object is defined as a cut divider by the requirement of cut detection. This object finds cut change points in the stream and creates new packages and segments under the target segment.

・*MovieDetector*

This object detects frames/sounds and generates new MovieFrame/Sound objects as key frames/sounds under the target segment. In the most typical case, a *MovieDetector* object is defined as a key frame detector. This object searches for important and distinctive frames in the target segment. Key frames/sounds detected by motion, still image or audio analysis are created and set under the segment as MovieFrame/Sound objects, which have a *MovieAttribute* object of image/wave features.

・*MovieExtractor*

The *MovieExtractor's* function is to extract features from frames/sound, segments as a *MovieFeature* object. This object is usually called from the *MovieDivider* or *MovieDetector* object as an analysis method and extracts some features from the target object.

・*MovieArranger*

The *MovieArranger* object arranges and creates MovieView objects. In the most typical case, a *MovieArranger* object is defined as a digest arranger. This object traces the PS-Model structure and retrieves segments the collection of which a user requested as a digest. Some MovieLink objects describe the location of the viewpoint by the package or segment node position. The *MovieArranger* object keeps those MovieLink objects in order and controls some display behaviors (e.g. slow, quick motion, etc.).

The relationship between the PS-Model structure and multimedia content is mediated by the *MovieStream* object, which is defined as a base class. In the fig.3, object relations are represented by a class diagram. The *MovieStream* object aggregates MovieFrame objects that belong to the MovieSegment. The MovieFrame objects in a MovieSegment represent the start and end frames of the segment. Each MovieFrame object can indicate the logical position in the PS-Model structure and access to the physical position of the content stream. However, the PS-Model structure is independent of the content. If it is necessary to refer to the content stream, some mediation objects are defined and are used to restore the physical relations.

3. Distributed Index

After generating the PS-Model structure, the structure is encoded as the retrieval index. The PS-Model structure is also used as index of retrieval, searching and browsing. We provide a separable index distribution scheme.

The PS-Model structure represents various types of information in one index for a particular content. It is easy to trace a structure and refer each node in one content. However, when some search engines refer to indexes in large quantities, it takes too much time to refer to all details and compare conditions. Because the complexity of the structure depends on the precision of the descriptions, there is no effective way to solve this problem. The Fig.4 shows the concept of separation of index. We separate some feature objects from the PS-Model structure and set into each feature space which manages a homogenous object. Each object keeps a link on both sides. This scheme improves the performance of search engines. We can provide the most effective search algorithms for each index space, and the processes of searching can run in parallel. This strategy is suitable for multi-thread process applications.

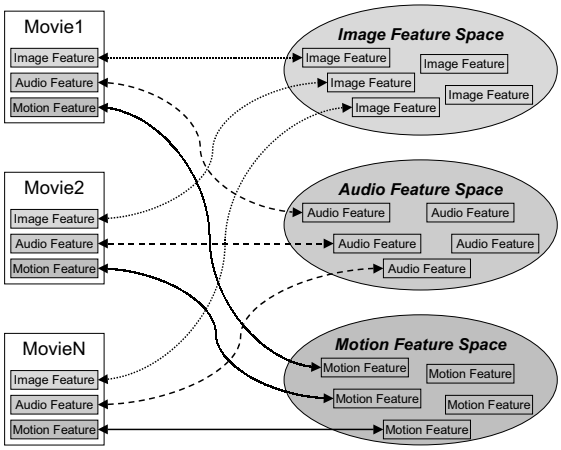

Fig.4 The concept of distributed index. Some feature objects are separated from the PS-Model structure.

4. Retrieval Mechanism

Some effective retrieval strategies have been proposed [4][6]. Here, to verify the adaptability of the PS-Model as a retrieval index, we provide indexing mechanisms for individual video content retrieval. In this index, there are many kinds of information that are used as attributes. When a user specifies a query for retrieval, some different category conditions are described. For representation of compound conditions, we provide a MovieRetrievalConditionBox object. This object can hold both heterogeneous and homogeneous conditions and can describe those logical relations. The following table is an example of compound conditions as queries.

Table 1 Sample compound conditions for retrieval

	Condition	Instances	Homogeneous logical relations	Type
A	Similar image	*P1, P2, P3*	*P1* and (*P2* or *P3*)	Image
B	Scene duration	*30 sec., 5 min.*	*30 sec.* < duration < *5 min.*	Time
C	Scene topics title	"*Olympic*", "*Ski*"	"*Olympic*" or "*Ski*"	String

- Relationship between heterogeneous conditions: C and (A or B)
- Target scope (Whole stream/Each scene): Each scene
- Evaluation point: 1 point

4-1 Attribute Comparison

There is a large amount of added or extracted information that represents the whole content and/or each scene. In the PS-Model, each node object such as a segment and a frame has an attribute object, which is defined as required. The searching method of the PS-Model is simple. An encoding index is loaded and reproduced as a tree structure. The detection of a node in the tree structure is processed in the descending order and if a condition is homogeneous, the searching module compares the attribute that the node has. If the comparison result is higher than the threshold or true, the MovieConditionBox records the node position in the structure and calculates a score, which is added for a ranking result.

4-2 Retrieval and Search Scope

If the user sends a query of compound conditions as a sample picture "P1" and scene topic "Olympic," how does the retrieval system reply? The problem is how to interpret the two conditions.

- Target scope: Whole stream

The user's requirement for retrieval is that the matching point of the two conditions is inside one content but not in the same scene. For example, a picture similar to "P1" appears in the head scene and the "Olympic" topic appears in the last scene. In this case, the retrieval result must be true.

- Target scope: Each scene

The user requires two conditions to coincide in the same scene. That is, the retrieval system finds the scene that satisfies the two conditions. The picture similar to "P1" and "Olympic" must appear in one scene in one content. If there is a scene which has a picture similar to "P1" and "Olympic" in one content. Only in this case the retrieval result must be true.

We distinguish between these retrieval scopes and provide two retrieval API's.

5. Adaptable Application Sample MovieTool

We implemented a tool named MovieTool to verify the effectiveness of the PS-Model and to experiment on the retrieval mechanism. This application provides three characteristic functions:

A) Automatic construction of the PS-Model structure by content-based analysis

Automatic indexing and indexing aid mechanisms are necessary to generate this type of structural information [7][8]. MovieTool provides the following analysis methods for the verifications.

・Segmentation methods:

Comparison of image features, constant time or time table, detection of the silent parts of audio, and event list marked by humans

・Key frame detection methods:

Constant selection of top, middle and end frames in a segment, camera parameters extracted by optical flow and rule-based key frame detection

B) Editing and modifying the PS-Model structure

The PS-Model structure is represented as a Tree View. A user can edit this structure by mouse operations.

C) Conversion to useful data

The PS-Model structure can be saved to various file formats. The file formats are as follows.

・Event List (input):

This is a text-based list of events in the content stream. It is provided for real-time event marking using an easy interface [9].

・PS XML (input/output):

This is eXtended Markup Language(XML) formatted text data. XML is suitable for representation of the PS-Model structure [10]. We provided the Document Type Definition for the PS-Model. Some feature descriptors are described as external links. This data can be also edited by hand and be reloaded to application.

・Retrieval Index (input/output):

This index is referred to by the retrieval engine mentioned above and specifies the content stream and scene position that a user requested.

・EDL (output):

This data format is in Editing Definition Language for stand-alone non-linear editing machines to create content. This format depends on the type of editing machine.

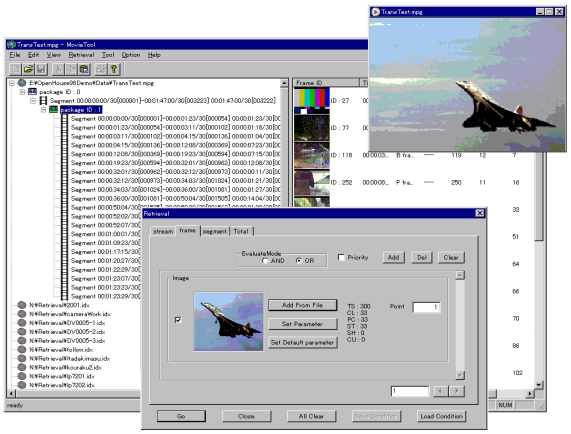

Fig.5 MovieTool main window which shows the PS-Model structure and retrieval condition dialog window which hold a sample image

・The program index for broadcasting (output):

The program index has been standardized in the Association of Radio Industries and Businesses in Japan [11]. It is a Meta-data of TV programs for new broadcasting services such as hierarchical program. It enables the specification of content inside an index for broadcasting.

D) Providing an experimental retrieval and search interface for the PS-Model index

This tool provides experimental retrieval interfaces. At the present time, conditions can apply for experimental retrieval and searching, such as duration of cut, similar still image, and stream duration. These conditions are extracted features only. Fig.5 shows the main window and the retrieval dialog window.

6. Conclusion and Future Work

This paper has examined the PS-Model. We have confirmed that the PS-Model has the ability to adapt to multimedia

content managing systems in terms of the representation flexibility, object framework and retrieval mechanism. Currently, some groups are using the PS-Model schema for applications such as real-time indexing systems for broadcasting programs, re-constructed retrieval of broadcasted contents from a Set Top Box (STB) [12][13][14]. Our final goal is a large-scale multimedia archive system.

Some problems, however, remain to be solved. Indexing work by hand takes a great deal of time, and content-based, automatic indexing techniques are still in their infancy. If we need to develop useful retrieval systems for multimedia contents, effective feature extraction and recognition techniques must be developed. The preparation of the infrastructure of indexing for the content producer is important. The provision of indexing aid system is an indispensable part of the authoring process. The spread of indexed contents will provide the impetus to the progress of technologies. In the short term, we will expand the target features to audio and motion. An improved index encoding scheme will also improve response time. In the long term, we will develop authoring systems that will provide indexing aid mechanisms for effective use of multimedia contents.

Reference

[1] MPEG Requirements group: MPEG-7 Requirements Document V.7, Doc. ISO/MPEG N2461, (1998).

[2] D. A. Adjeroh, K. C. Nwosu: Multimedia Database Management Requirements and Issues, IEEE Multimedia, 4, 3 (1997), pp.24-33.

[3] Jan Gecsei: Adaptation in Distributed Multimedia Systems, IEEE Multimedia, 4, 2 (1997), pp.58-66.

[4] H. J. Zhang et al.: Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution, Proc. ACM Multimedia (1995), pp.15-24.

[5] H. Ueda, T. Miyatake: Automatic Scene Separation and Tree Structure GUI for Video Editing, Proc. ACM Multimedia (1996), pp.405-406.

[6] M. G. Brown et al.: Automatic Content-Based Retrieval of Broadcast News, Proc. ACM Multimedia (1996), pp.35-43.

[7] Y. Taniguchi et al.: An Intuitive and Efficient Access Interface to Real-Time Incoming Video Based on Automatic Indexing, Proc. ACM Multimedia (1995), pp.25-33.

[8] H. Aoki, S. Shimotsuji, O. Hori: A Shot Classification Method of Selecting Effective Key-Frame for Video Browsing, Proc. ACM Multimedia (1996), pp.1-10.

[9] Service Information for Digital Broadcasting System", ARIB STD-B10 V1.2, Part.3, Association of Radio industries and Businesses Japan (1999)

[10] B. Z. Gottesman: Why XML Matters, PC Magazine, October 6 (1998), pp.215-238.

[11] J. Kuboki, T. Hashimoto, T. Kimura: Method of Creation of Meta-Data for TV Production Using General Event List (GEL), The Institute of Image Information and Television Engineers Technical Report (1999), pp.1-6.

[12] T. Hashimoto, Y. Shirota, and T. Kimura: Digested TV Program Viewing Application Using Program Index, The Institute of Image Information and Television Engineers Technical Report (1999), pp.7-12.

[13] T. Hashimoto, Y. Shirota, J. Kuboki, T. Kunieda, A. Iizawa: A Prototype of Digest Making Method Using the Program Index, Proc. of IEICE Data Engineering Workshop CD-ROM (1999).

[14] T. Hashimoto, Y. Shirota, H. Mano, and A. Iizawa: Prototype of Digest Making and Viewing System for Television, IPSJ-SIGDBS Database Workshop (1999), (to appear).