Semantic Video Model for Content-based Retrieval*

Jia-Ling Koh
Department of Information and Computer Education
National Taiwan Normal University
Taipei, Taiwan 106, R.O.C.
Email: jlkoh@ice.ntnu.edu.tw

Chin-Sung Lee, Arbee L. P. Chen
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
Email: alpchen@cs.nthu.edu.tw

Abstract

Traditional research on video data retrieval follows two general approaches. One is based on text annotation and the other on content-based comparison. However, these approaches do not fully make use of the meaning implied in a video stream. To improve these approaches, a semantic video model cooperated with a knowledge database is studied. In this paper, we propose a new semantic video model and focus on presenting the semantic meaning implied in a video. According to the granularity of the meaning implied in a video, a five-level layered structure to model a video stream is proposed. A mechanism is also provided to construct the five levels based on the knowledge categories defined in the knowledge database. The five-level layered structure consists of raw-data levels and semantic-data levels. A uniform semantics representation is proposed to represent the semantic-data levels. This uniform semantics representation allows measuring the similarity of two video streams with different duration. Then an interactive interface can provide browsing and querying video data efficiently through the uniform semantics representation.

1. Introduction

Video databases have received much interest in recent years. The modeling and indexing methods will influence the retrieving capability provided by a database. Traditional research on video data retrieval follows two general approaches. The first approach is based on text annotation. The querying power was usually limited by the fixed set of keywords and allows only exactly matching [3]. Another approach is based on visual content comparison. The QBIC system [6] and the VIMSYS system [13] considered the content of a video as a sequence of images. The extracted image features such as color and shape are used to index the key frames in a video. Ono et al. [8] combined image contents and descriptions to make retrieval flexible. The disadvantage of the second approach is that it can not support retrieval on general concepts.

In most situations, the duration of a video stream is long and the contained content is complex. To simplify the analysis of the visual content, a data model for representing a video stream is needed. [5] and [14] use the key frames derived from the shots to represent a video. Many approaches have been proposed for segmenting videos into shots [2, 12]. We also proposed a strategy to detect the shot boundaries in the compressed domain [7]. Moreover, building a video hierarchy to represent the detailed content contained in a video stream was provided in [4]. A general mechanism is to construct the video hierarchy with four levels: frame level, shot level, scene level, and video level [10]. Each level is constructed from its underlying level [13]. One problem of the hierarchy of four levels proposed before is that the unit of shot level is too rough to identify the duration of an event exactly. The duration of an event in a shot may be within or cross the boundaries of the shot. The defined units can not overlap is another problem. The video streams may contain special effects like *fade-in*, *fade-out*, and *dissolve* after post-production. It is not easy to decide the boundary between the two adjacent video clips.

The browsing and querying functions are usually separately provided in traditional approaches. However, the browsing function only focuses on a single video and users have to traverse the video database to find the required video clip. Suppose the querying function can be combined with the browsing function. When a video clip similar to the required video clip is found by browsing, the clip can be used to query the video database to find the other similar video clips. On the other hand, the returned results of a query are not guaranteed to satisfy the users' requirement exactly. The verifying process can be done through the browsing of each returned video clip. Therefore, it will improve the data retrieving power to integrate these two functions.

To summarize the discussion for the disadvantages and advantages of different approaches, a semantic video model cooperated with a knowledge database is studied. In this paper, a new semantic video model is proposed for representing the semantic meaning implied in a video. According to the granularity of the meaning implied in a video, a five-level layered structure to model a video stream is proposed. The five levels are frame, chunk, sequence, scene, and video, which can be used to precisely model the video data as layered meaningful pieces. A mechanism is also provided to construct the five levels based on the knowledge categories defined in the knowledge database. The five-level layered structure consists of raw-data levels and semantic-data levels. A uniform semantics representation is proposed to represent the semantic-data levels. This uniform semantics representation allows measuring the similarity of two video streams with different duration. Then an interactive interface can provide browsing and querying video data efficiently through the uniform representation.

For classifying and annotating videos, a concept knowledge database was proposed in [11]. In this approach, the descriptions

*This work was partially supported by the Republic of China National Science Council under Contract No. NSC 88-2213-E-007-052.

0-7695-0253-9/99 $10.00 © 1999 IEEE

for a video clip were simplified to the associated concepts. A video can be thus described by a sequence of concepts, which represent the meaning contained in a video with the easily understood form. The OVID system [9] organized the video as a hierarchy structure using an object-oriented system, which was the one closest to our semantic video model. The main difference is that our approach is cooperated with a concept knowledge database instead of using the descriptive attributes as the OVID system. A concept vector with a set of degrees for different concepts is designed to represent the semantics in a video clip. The uniform representation is proposed for units in semantic-data levels, which is also used to denote a query predicate. Therefore, it is easy to browse and query the similar video clips according to their semantics even they belong to different levels or have different duration. Moreover, through the uniform representation, the cooperation of browsing and querying functions provides more flexible and efficient power on both functions.

The paper is organized as follows. Section 2 describes the five-layered structure in our semantic video model. The representations of the units in the five-level layered structure are also defined in this section. Section 3 details the process to construct the units in semantic-data levels. The integrated video querying and browsing functions based on the uniform semantics representation are discussed in Section 4. Finally, Section 5 concludes this paper and discussed the future research issues.

2. Architecture of Semantic Video Model

2.1 Five-Level Layered Structure of Video Data

In our semantic video model, the consecutive and similar frames in a video are grouped as a unit and as a semantics representation for these frames. A video is organized as a layered structure consisting of five levels: *frame level*, *chunk level*, *sequence level*, *scene level*, and *video level*.

An event occurring in a video stream implies the concept in these frames. In addition, the set of objects which appear in a sequence of frames represent the context in these frames. Therefore, the object and event are defined to be two types of *semantic item* (SI), which will be considered in the process of structure construction.

The five levels in the layered structure are further divided into two categories. The first category is called *raw-data* levels including frame level and chunk level. The units defined in the raw-data levels have short duration and may not represent certain semantics completely. Another category is called *semantic-data* levels including the other three levels: sequence level, scene level, and video level. The duration of the units defined in semantic-data levels is long enough to represent certain semantics completely. As the discussion in Section 1, a unit in the shot level defined traditionally can not be used to describe the duration of an event accurately. Therefore, the shot level is divided into two levels: chunk level and sequence level in our layered structure. The whole layered structure is shown in Figure 1.

2.1.1 Raw-Data Levels

Frame level: Frame level is the bottom level in the five-level layered structure. The unit defined in the frame level is a *frame*, which corresponds to a video frame.

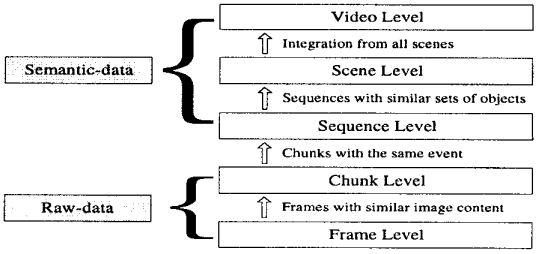
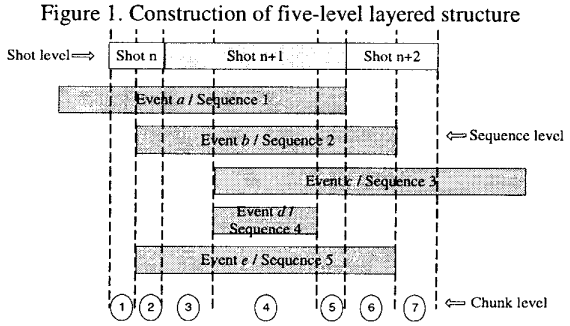


Figure 1. Construction of five-level layered structure



(a) An example about shot, chunk, and sequence

Figure 2. Example of chunks and sequences

Chunk level: The unit defined in the chunk level is named a *chunk*. A chunk consists of a sub-set of frames contained in a shot, in which the consecutive frames have the most similar visual contents and represent the same set of events. The boundaries of each chunk occur in the following four situations: the appearance of a new event, the finish of an old event, the starting of a shot, and the finish of a shot. The duration of different chunks is disjoint. Thus, a traditional shot will be divided into several chunks. For example, three shots are divided into 7 chunks as shown in Figure 2. For chunk 2 and chunk 3, they are separated by the boundary of shot n and shot n+1.

A chunk only represents part of the semantics implied by the events, which is the bridge that connects the raw-data levels and semantic-data levels. The representation and semantics of the three upper semantic-data levels is constructed from chunk level.

2.1.2 Semantic-Data Levels

Sequence level: The unit defined in the sequence level is named a *sequence* consisting of a collection of adjacent chunks to represent a specific event completely. A sequence can contain one or more chunks. It is possible that the different events happen within the same duration. Therefore, the duration of two sequences can overlap. Two sequences representing two different events also can have the same duration time. As the example in Figure 2, sequence 5 is integrated from 5 chunks which all contain event E.

Scene level: A *scene* is the unit defined in the scene level, which is a collection of adjacent sequences containing related context. The adjacent sequences containing similar set of objects are merged as a scene according to the related context. A scene can contain one or more sequences. In addition, the duration of two

scenes can overlap.

Video level: A *video* is the unit in the video level, which consists of all the underlying scenes. A video can contain one or more scenes.

In the following context, a period of video duration is named a *sub-video*. The sequence, scene and video are named *semantic sub-video* because the content in their duration represents certain semantics completely.

2.2 Semantics Representation of Video Data

After constructing a five-level layered structure for a video stream, a *uniform semantics representation* for the semantic-data levels is proposed. In order to represent the semantics represented in the semantic-data levels, a knowledge database providing a set of concepts is used to cooperate with the video database. The *concept vector* based on these concepts is defined to represent the semantics of a semantic item and a semantic sub-video. We assume the knowledge database contains N different concepts. In addition, there exists a set of identifiers (*IDs*) for denoting the concepts, objects, and events.

[Definition 2.1 Concept]: A concept [1] is "*A general idea derived or inferred from specific instances or occurrences*", e.g. working and fighting. A concept [1] is "*Something formed in the mind; a thought or notion*", e.g. scaring and romantic. Each concept is denoted by an identifier c_i.

[Definition 2.2 Concept Vector]: A concept vector is denoted by an N-tuple $[c_1, c_2, \ldots, c_N]$, where each entry in the N-tuple corresponds to a concept in the knowledge database. The value of c_I (i = 1,..., N) denotes the degree of the associated concept implied in the represented semantics, which is named *semantic degree* of the concept.

Example 1:

Assume the knowledge database provides four concepts: *working, fighting, scared* and *romantic*. The semantics implied in a given sub-video is represented by a four-tuple concept vector [39, 12, 3, 46]. The concept vector denotes that the semantic degree of working concept is 39, the semantic degree of fighting concept is 12, and so on.

To construct the three semantic-data levels, the semantics extraction begins from the chunk level. Each chunk is assigned a concept for denoting the semantics in its content. All SIs appearing in a chunk also need to be recorded. A chunk can contain a set of SIs. A SI also can appear in more than one chunk. The semantics of a SI is implied by the concepts of the chunks containing the SI, which is denoted by a concept vector as the following definitions.

[Definition 2.2 Semantic Item - Object]: An object, which appears in a single video frame and represents a real-world entity, is used to represent the spatial feature. An object is denoted as $SI_o = (oid, W)$, where *oid* is an object identifier and W is the associated concept vector.

[Definition 2.3 Semantic Item - Event]: An event, which appears in consecutive frames and represents an action, is used to represent the temporal feature. An event is denoted as $SI_e = (eid, W)$, where *eid* is an event identifier and W is the associated concept vector.

***[Definition 2.4 Representation of Chunk Level]*:** Given a chunk which is assigned a concept c_i, the chunk is represented as $RH = (T, E, O, c_i)$ where

1) $T = [t_1, t_2]$, where t_1 is the starting frame of the chunk and t_2 is the ending frame of the chunk;
2) E is a P-tuple $[e_1, e_2, \ldots, e_P]$, where P is the number of events contained in this chunk and each e_i (i = 1, ..., P) is an event identifier;
3) O is a Q-tuple $[o_1, o_2, \ldots, o_Q]$, where Q is the number of objects contained in this chunk and each o_i (i = 1, ..., Q) is an object identifier;
4) c_i is the assigned concept identifier.

Example 2: (continued from Example 1)

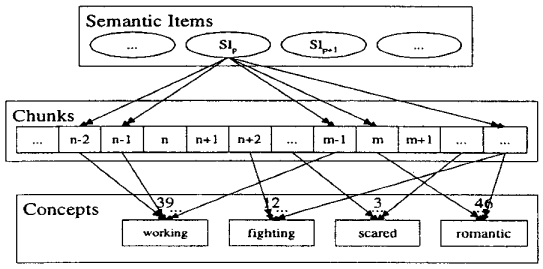


Figure 3. Example of concept vector construction for SI

In this example, the concept vector for semantic item SI_p is [39, 12, 3, 46]. The vector denotes that there are 39 chunks where SI_p appears. Besides, the semantic degree of working concept for each chunk is 1 and the summary semantic degree is 39. It is similar for describing the other three concepts.

The values in the concept vector of a SI are not fixed, which can be modified to reflect the extracted semantics in the video database until now. Each time a new video stream is inserted into the video database, the concept vectors of the contained SIs will be adjusted as Example 3.

Example 3: (continued from Example 1)

Suppose the content in the concept vector of SI_P is [39, 12, 3, 46]. When the SI_P is newly identified in a chunk with concept *romantic*, then the concept vector of SI_P will be modified into [39, 12, 3, *46+1=47*].

The three types of semantic sub-videos: sequence, scene, and video have a uniform representation, which is a 2-tuple value (R, W). R is used to denote the composed sub-videos in the underlying level. Each unit in the three semantic-data levels all represents certain semantics which is represented by a concept vector W as that used to describe the semantics of a SI.

[Definition 2.5 Uniform Semantics Representation of the Semantic Levels]: Each unit in the semantic levels is represented as $SU = (R, W)$, where

1) $R = [r_1, r_2]$, where
   (a) if SU is a sequence, r_1 is the ID of the starting chunk and r_2 is the ID of the ending chunk;
   (b) if SU is a scene, r_1 is the ID of the starting sequence and r_2 is the ID of the ending sequence;
   (c) if SU is a video, r_1=1 and r_2 is the ID of the last scene;
2) W is the associated concept vector.

In Section 4, we will show a new kind of querying method, where the querying predicate is also represented as the uniform semantics representation. Then the concept vector is used to match with those contained in the video database.

3. Semantic-Data Levels Construction

In this section, the process to construct the representation of a SI is introduced in detail. Also, the algorithms used to detect the units in the three semantic-data levels are described. Then the concept vector of each semantic sub-video is obtained according to the contained SIs. Furthermore, we will discuss the situations that the semantic degrees in the concept vector of a SI need to be adjusted.

3.1 Concept Vector Construction of SIs

When an event or an object is identified in a chunk, the concept vector of the SI needs to be adjusted. According to the concept that the chunk is assigned, the associated semantic degree in the concept vector is updated by adding 1. The pseudo codes are listed as below.

[Function: Concept Vector Construction Function of SI]
When a SI si_p is identified in chunk RH_q which is assigned a concept with identifier C_q.
if semantic item si_p has not yet existed in the database,
{get a new identifier id_p ,
$si_p = (id_p, [w_{p1}, w_{p2}, ..., w_{pN}])$, where w_{pi}=0 for i=1, …, N }

$$\begin{cases} w_{pn} = w_{pn} & ,n \neq C_q \quad n=1, \ldots, N \\ w_{pn} = w_{pn} + 1, & n = C_q \end{cases}$$

3.2 Sequence Construction from Chunks

According to the definition of a sequence, adjacent chunks containing a certain event are grouped to be a new sequence. For each event, every chunk has to be checked one by one to detect all possible sequences. The procedure *construct_ sequence* is used to construct sequences from the underlying chunks, where the pseudo codes are listed as below.

[Algorithm: Sequence Construction]
Let me_{ij} be the membership value of event i existing in chunk j.

$$\begin{cases} me_{ij} = 1, \text{if event i is identified in chunk j} \\ me_{ij} = 0, \text{otherwise} \end{cases}$$

```
Procedure construct_sequence(Video V)
{ I = number of events contained in video V
  J = number of chunks contained in video V
    for i = 1 to I
      { set j = 1
        while (j < J) do
              if (mei,j == 1)
                {create a new sequence, mark start of the sequence at chunk j
                 for k = (j+1) to J
                       if ( meik== 0)
                         {mark end of the sequence at chunk k
                         compute the semantic degrees in the semantic vector
                         of the new sequence (using equation 3.1)
                         set j = k+1
                         break for loop} /* end if */

                } /* end if */
      }/* end for */ }
```

Because a sequence is constructed according to a specific event, the semantic vector of the newly detected sequence is derived from the represented event. In addition, two properties are considered. The first property is that a sequence consists of more chunks represents the semantics of the event more accurately. Therefore, the semantic degree implied by a sequence is proportion to the number of contained chunks. This consideration is applied in equation 3.1(a). Moreover, it is possible that the various chunks in a sequence are assigned different concepts although they contain the same event. If more chunks are assigned a certain concept, more the concept is implied in the semantics of the sequence. This second consideration is applied in equation 3.1(b). Then each entry in the concept vector of a newly detected sequence is the sum of 3.1(a) and 3.1(b).

[Function: Concept Vector Construction Function of Sequence] Let C_m denote the identifier of the concept assigned to chunk m, and $E_{C_m C_i}$ present the equality value that C_m is equal to concept C_i.

$$\text{Then} \begin{cases} E_{C_m C_i} = 1, \text{if identifier } C_m \text{ is equal to } C_i \\ E_{C_m C_i} = 0, \text{otherwise} \end{cases}$$

Given a sequence SQ_x = ($[r_{x1}, r_{x2}]$, $[w_{x1}, w_{x2}, ..., w_{xN}]$) which is constructed by the event e_j = (eid_j, $[w_{j1}, w_{j2}, ..., w_{jN}]$), each entry in the concept vector of this sequence is calculated as the sum of equation 3.1(a) and equation 3.1(b), where n=1, .., N.

$$w^1_{xn} = eid_j.w_{jn} * (SQ_x.r_{x2} - SQ_x.r_{x1} + 1) \quad \text{... (3.1 a)}$$

$$w^2_{xn} = \sum_{m=1}^{m=N} E_{C_m C_n} \quad \text{... (3.1 b)}$$

$$w_{xn} = w^1_{xn} + w^2_{xn} \quad \text{... (3.1)}$$

Example 4 :

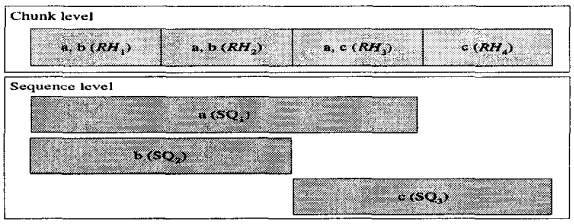


Figure 4. Example of sequence construction

Assume there exists a set of events E = {a, b, c, d}, where e_a = (eid_a, [39, 12, 3, 46]) and RH_1, RH_2, RH_3 are classified into concept c_1, c_1, and c_3, respectively. The sequence SQ_1 is constructed from RH_1 to RH_3 by event e_a. Then SQ_1 = ([1, 3], [w_1, w_2, w_3, w_4]), where the concept vector is computed as below.

According to 3.1a :	According to 3.1b
w^1_{11}= 39 * 3 = 117	w^2_{11} = 1 + 1+ 0 + 0 = 2
w^1_{12}= 12 * 3 = 36	w^2_{12} = 0 + 0 + 0 + 0 = 0
w^1_{13} = 3 * 3 = 9	w^2_{13} = 0 + 0 + 1 + 0 = 1
w^1_{14} = 46 * 3 = 138	w^2_{14} = 0 + 0 + 0 + 0 = 0

Finally SQ_1 = {[1, 3], [119, 36, 10, 138]}

3.3 Scene Construction from Sequences

According to the definition of a scene, the adjacent sequences containing similar sets of objects represent a scene with related context. In order to decide which sets of sequences should be grouped into a scene, the dissimilarity between two adjacent sequences needs to be evaluated. Here the dissimilarity between two sequences is defined as the difference of the two sets of objects contained in these two sequences.

[Function: Dissimilarity Function of Sequences]
Given two sequences SQ_P and SQ_Q, where SQ_P appears before SQ_Q. $SQ_P = ([r_{P1}, r_{P2}], [w_{P1}, w_{P2}, ..., w_{PN}])$, which contains a set of X objects $O_P=\{o_1, o_2, ..., o_X\}$. In addition, $SQ_Q = ([r_{Q1}, r_{Q2}], [w_{Q1}, w_{Q2}, ..., w_{QN}])$, which contains a set of Y objects $O_Q=\{o_1, o_2, ..., o_Y\}$. A set of R objects $O = \{ o_r \mid o_r \in O_Q \text{ and } o_r \notin O_P\}$ $(r = 1, ..., R)$ in SQ_Q are different from those contained in SQ_P. Then the dissimilarity value of sequence SQ_Q compared with sequence SQ_P is defined as below:

$$dissimilar(SQ_P, SQ_Q) = \frac{R}{Y} \quad \cdots (3.2)$$

The dissimilarity function is listed in equation 3.2. The result of the dissimilarity function depends on the order used to check these two adjacent sequences. The reason is that the ending of a scene is detected when the later sequence contains a large ratio of newly appearing objects compared with the objects in the previous sequence.

In order to detect the boundaries of scenes, a threshold value δ is required to set the maximum dissimilarity value between two adjacent sequences in the same scene. If the dissimilarity value between two adjacent sequences is lower than the threshold value δ, these two sequences are integrated to construct a newly scene as below. The procedure *construct_scene* is used to construct scenes from the underlying sequences, where the pseudo codes are listed below.

[Algorithm: Scene Construction]
Let integrate(SQ_P, SQ_Q)=SS_X=([r_{x1}, r_{x2}], [w_{x1}, w_{x2}, ..., w_{xN}])... (3.3)
where a) $r_{x1} = \min(r_{P1}, r_{Q1})$ and $r_{x2} = \min(r_{P2}, r_{Q2})$
b) w_{xn} is computed by equation 3.4, n=1,...,N
c) The scene SS_x contains all the objects appearing in SQ_P and SQ_Q.

```
Procedure construct_scene(Video V)
{ l = number of sequences contained in video V
  for i = 1 to l {
        create a new scene
        mark start of the scene at sequence i
        for j = (i+1) to l
          if (dissimilar(sequence i, sequence j) ≥ δ )
                {mark end of the scene at sequence j
                 set j = k
                 break;}
          else {integrate sequence j into this newly constructed
                scene using equation 3.3}
   } /*end for */ }
```

The related context in a scene is represented by the similar sets of objects. Therefore, the concept vectors of the contained objects in a scene are integrated to represent the implied semantics in the scene. A scene is composed of one or more sequences and an object contained in the scene may not appear in all of these sequences. For each contained object, the weight of its semantics which contributing to the scene is proportional to the number of sequences containing the object. An object appearing in more sequences will have higher weight. That is the consideration applied in equation 3.4.

[Concept Vector Construction Function of Sequence] Let NS_{ym} denote the number of sequences which are included in the same scene SS_y and contain the object O_m.
Given a scene SS_y = ([r_{y1}, r_{y2}],[w_{y1}, w_{y2},...,w_{yn}])) which contains a set of objects O=$\{o_1, o_2, ..., o_X\}$. The concept vector of this scene is calculated as below:

$$w_{yn} = \sum_{x=1}^{X} (o_x . w_{xn}) * (NS_{yx}) \quad n=1,..,N \quad \cdots (3.4)$$

Example 5:
Suppose there exists four sequences as Figure 5. The set of contained objects is {M, N, O, P, Q, S, T, U, V}, where FO_M = (oid_M, [w_{M1}, w_{M2}, w_{M3}, w_{M4}]), FO_N = (oid_N, [w_{N1}, w_{N2}, w_{N3}, w_{N4}]), FO_O = (oid_O, [w_{O1}, w_{O2}, w_{O3}, w_{O4}]), FO_P = (oid_P, [w_{P1}, w_{P2}, w_{P3}, w_{P4}]), FO_Q = (oid_Q, [w_{Q1}, w_{Q2}, w_{Q3}, w_{Q4}]), and FO_S = (oid_S, [w_{S1}, w_{S2}, w_{S3}, w_{S4}]).
To construct a new scene, the adjacent sequences are integrated one-by-one. The process is stopped when adding sequence-4 because the dissimilarity value 0.6 is larger than the threshold value 0.5. Therefore, the scene SS_1 is constructed from SQ_1 to SQ_3 with a set of objects {M, N, O, P, Q, S}.

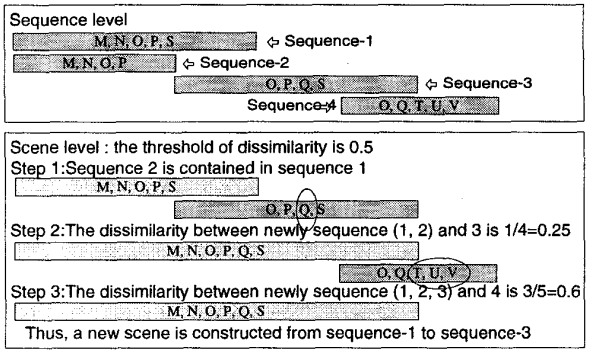


Figure 5. Example of scene construction

Then $SS_1 = \{[1, 3], [w_1, w_2, w_3, w_4]\}$, where
$SS_1.w_1 = w_{M1}*2+w_{N1}*2+w_{O1}*3+w_{P1}*3+w_{Q1}*1+w_{S1}*2$
$SS_1.w_2 = w_{M2}*2+w_{N2}*2+w_{O2}*3+w_{P2}*3+w_{Q2}*1+w_{S2}*2$
$SS_1.w_3 = w_{M3}*2+w_{N3}*2+w_{O3}*3+w_{P3}*3+w_{Q3}*1+w_{S3} * 2$
$SS_1.w_4 = w_{M4}*2+w_{N4}*2+w_{O4}*3+w_{P4}*3+w_{Q4}*1+w_{S4} * 2$

3.4 Video Construction from Scenes

A sequence or a scene is a sub-video which represents part of the semantics contained in the whole video, which gives a local view for the associated video time with different kind of viewpoint. Therefore, the complete semantics of a video is integrated from the semantics implied in all the sequences and scenes. The pseudo codes for constructing a video are listed below.

[Algorithm: Video Construction]
Given a video SV_x which contains P sequences and Q scenes, where
1) each sequence SQ_p=([r_{p1}, r_{p2}], [w_{p1}, w_{p2},...,w_{pN}]) p=1,...,P
2) each scene SS_q=([r_{q1}, r_{q2}], [w_{q1}, w_{q2},...,w_{qN}]) q=1,...,Q
Then the video SV_x=([r_{x1}, r_{x2}], [w_{x1}, w_{x2},...,w_{xN}]), where
a) $x1$=1 and $x2$=Q
b) $w_{xn} = \sum_{p=1}^{p=P} w_{pn} + \sum_{q=1}^{q=Q} w_{qn}$

Example 6 :
The whole layered structure of a video for example 4 and 5 is shown as Figure 6. The video contains a set of objects {M, N, O, P, Q, R, S, T, U, V} and a set of events {a, b, c, d}. There are 5 sequences and 2 scenes constructed.

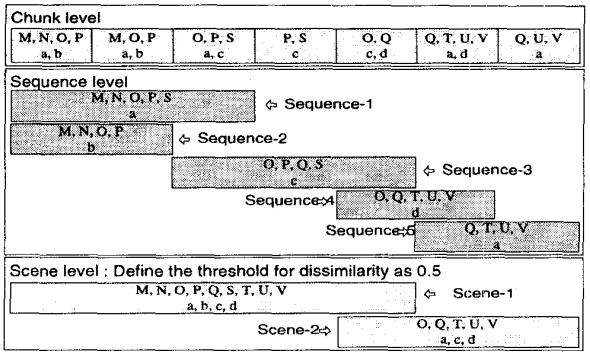


Figure 6. Example of video construction

3.5 Concept Vector Adjustment for SIs

The concept vector of a SI is constructed according to the concepts in the chunks which contain the SI. In addition, the semantics in the contained SI will be enhanced according to the semantics in the sub-video. A newly constructed semantic sub-video will influence the degree of each concept implied in the related SIs. Therefore, when a semantic sub-video is constructed, the concept vectors of the contained SIs need to be adjusted.

3.5.1 Concept Vector Adjustment for Events

The influence weight of the semantics in a sequence to an event e_p is proportional to the number of chunks in the sequence. Therefore, n times of the semantic degrees of various concepts implied by the sequence is added into the ones of the event e_p when a sequence containing n chunks is constructed.

[Function: Concept Vector Adjustment Function of Event]

When a new sequence $SQ_q = ([r_{q1}, r_{q2}][w_{q1}, w_{q2}, ..., w_{qN}])$ is detected using the event $e_p = (id_{ep}, [w_{p1}, w_{p2}, ..., w_{pN}])$, the new concept vector of e_p is calculated as below :

Assume the new value for w_{pn} in the concept vector is w'_{pn},

$w'_{pn} = w_{pn} + w_{qn}*(r_{q2}-r_{q1}+1)$ where n=1, ..., N.

3.5.2 Concept Vector Adjustment for Objects

The influence weight of the semantics in a scene to an object is proportional to the number of sequences containing the object. When a scene is identified, let n denote the number of sequences containing a specific object o_p. Then n times of the semantic degrees of various concepts implied by the scene are added into the ones of the object o_p.

[Function: Concept Vector Adjustment Function of Object]

When a new scene $SS_q = ([r_{q1}, r_{q2}] [w_{q1}, w_{q2}, ..., w_{qN}])$ is detected, which contains a set of P objects. For each object o_p in the set, $o_p = (id_{op}, [w_{p1}, w_{p2}, ..., w_{pN}])$ (p=1, ...,P). The new concept vector for each o_p is calculated as below with the weight value NS_{qp} defined in equation 3.4.

Assume the new value for w_{pn} in the concept vector is w'_{pn},

$w'_{pn} = w_{pn} + w_{qn} * NS_{qp}$, where p=1,..., P, n=1,..., N.

4. Semantic Video Browsing and Querying

An integrated method including browsing and querying functions is proposed based on the uniform semantics representation for all the semantic items and semantic sub-videos. The users can use currently browsing video clip, which is closing to the request, as a querying predicate to retrieve the sub-videos containing the similar semantics. Moreover, users can submit a query for finding the required sub-videos and verify the whole contents of the returned results by the browsing function.

4.1 Interactive Querying Functions

To retrieve a set of sub-videos containing the similar semantics with a given sub-video, a function is needed to evaluate the semantics similarity between two sub-videos. In our semantic video model, the semantics similarity between two sub-videos can be evaluated even their duration is different. The basic assumption is that the given sub-video must contain certain semantics completely and be a unit in one of the semantic-data levels.

4.1.1 Semantics Similarity Evaluation

The concept vectors of two semantic sub-videos are used to evaluate the semantics similarity between these two sub-videos. The semantics dissimilarity between two semantic sub-videos is calculated by using the Manhattan distance between their normalized concept vectors. The semantics similarity is obtained to subtract the semantics dissimilarity from 1.

[Function: Semantics Similarity Function]

Given two semantic sub-videos whose representation are

$V_x = \{[r_{x1}, r_{x2}], [w_{x1}, w_{x2}, ..., w_{xN}]\}$ and $V_y = \{[r_{y1}, r_{y2}], [w_{y1}, w_{y2}, ..., w_{yN}]\}$.

Let the normalized concept vectors of V_x and V_y be $[w'_{x1}, w'_{x2}, ..., w'_{xN}]$ and $[w'_{y1}, w'_{y2}, ..., w'_{yN}]$, respectively.

$$w'_{xn} = \frac{w_{xn}}{\sum_{z=1}^{z=N} w_{xz}}, (n = 1,..., N) \text{ and } w'_{yn} = \frac{w_{yn}}{\sum_{z=1}^{z=N} w_{yz}}, (n = 1,..., N)$$

The semantics dissimilarity function $DS(V_x, V_y) = \frac{\sum_{n=1}^{n=N} |w'_{xn} - w'_{yn}|}{N}$

Then the semantics similar function is defined as $S(V_x, V_y) = 1 - DS(V_x, V_y)$

4.1.2 Video Retrieval by Querying Functions

A new kind of querying method through the concept vectors is provided in our semantic video model. Then a user can give one concept vector **W** to be the querying predicate, where **W** can come from three ways.

<1> Use a browsing result: Users can stop browsing a video and use the concept vector associated with currently selected semantic sub-video to form a querying predicate. The returned results will be a list of semantic sub-videos which contain the similar semantics and located at the same semantic-data level with the given sub-video.

<2> Specify a concept vector: Users can describe the implied semantics in the required semantic sub-videos directly. The semantic degrees in the concept vector are specified to form a querying predicate. By using this way, all the semantic sub-videos in the video database will be checked.

<3> Specify Semantic Features: Events and objects are the semantic features in the sub-videos. Users can specify an event to retrieve related sequences. By using this kind of querying method, the concept vector of this event is used as the querying predicate. Users also can give a set of objects to retrieve the

related scenes. For processing this kind of query, the set of concept vectors of these objects are integrated and used as the querying predicate.

In order to bound the number of the returned results, a threshold value to set the maximum dissimilarity value or a maximum counts for candidates needs to be given. These two parameters can be pre-defined in the system or given by the users.

4.2 Interactive Browsing Functions

Browsing is the most intuitive way for searching a required video clip in a video sequence. To provide a efficient and flexible way to go through one video, three browsing functions are proposed.

<1> Layered Browsing: This approach provides the browsing function to view sub-videos from the topmost level to the lowest level. When a unit in the five-level layered structure is selected, its concept vector and the composed units in the underlying level will be shown. The browsing of video is in a linear order until the frame level. For example, when users select a video to browse, the concept vector of the video is shown. In addition, the first frames of the composed scenes are also shown to represent the content in these scenes. After selecting a specific scene to continuous browsing, the associated concept vector and the first frames in the composed sequences will be shown, and so on.

<2> Semantic Browsing: Users can select a semantic item to list the related sub-videos in the video being currently browsed. When a user selects an event as the filtering predicate, the first frames of the sequences containing this event will be listed. Similarly, when a user selects a set of objects as the filtering predicate, the first frames of the scenes containing this set of objects will be shown. This approach allows users to browse a video in non-linear order.

<3> Resultant Browsing: Since the semantics representations of sub-videos used in browsing function and querying function are the same, the returned results of a query can be used to do further browsing. The first frames and the concept vectors of the results will be shown. Then users can select one of the results and apply the above two browsing methods to do further browsing.

In order to realize the feasibility and flexibility of our data model, a prototype system has been implemented. There are two main components in the experimental semantic video system. The first one is an indexing tool used for constructing the index for all the units in the five-layered structure. The other one is an interactive retrieving tool. Both the browsing function and querying function are provided in this tool. Due to the page limit, the prototype system is not introduced in detail.

5. Conclusion

The traditional research on video data retrieval does not fully make use of the semantics implied in a video stream. A five-level layered structure is proposed in our semantic video model for providing the semantics implied in a video. The five-level layered structure newly proposed, which includes frame level, chunk level, sequence level, scene level, and video level, can segment a video into semantic units more accurately. For each unit in the semantic-data levels, the concept vector with a set of semantic degrees of concepts is used to present the implied semantics. This notation can represent different kind of semantics more exactly. The uniform semantics representation is also used to represent a querying predicate. Moreover, a mechanism is provided to integrate both browsing function and retrieval function based on the uniform semantics representation.

In current state, the semantic items need to be identified manually. Developing a tool by combining the knowledge databases and the techniques of pattern recognition to semi-automatically detect the objects is an important research issue, which is currently under our consideration. To identify the events automatically is a further advanced research.

References

[1] American Heritager Dictionary of the English Language, *Third Edition by Houghton Mifflin Company.*

[2] H. Aoki, S. Shimotsuji, and O. Hori, "A Shot Classification Method of Selecting Effective Key-Frames for Video Browsing", *ACM Multimedia Proceedings*, 1996.

[3] T. Chua and L. Ruan, "A Video Retrieval and Sequencing System," *ACM Transaction on Information Systems*, Vol. 13, No. 4, 1995.

[4] G. Davenport, T.A. Smith, and N. Pincever, "Cinematic Primitives for Multimedia", *IEEE Computer Graphics and Applications*, 1991.

[5] N. Dimitrova, T. McGee, H. Elenbaas, "Video Keyframe Extraction and Filtering: A Keyframe is not a Keyframe to Everyone", *International Conference on Information and Knowledge Management*, 1997.

[6] M. Fickner, H.Sawhney, W.Niblack, *et.al*, "Query by Image Content: The QBIC System", *IEEE Computer*, Vol. 28, No.9, September 1995.

[7] T.C.T. Kuo, Y.B. Lin and A.L.P. Chen, "Efficient Shot Change Detection on Compressed Video Data" *Proceedings of IEEE Workshop on Multimedia Database Management Systems*, 1996.

[8] A. Ono, M. Amano, M. Hakaridani, "A Flexible Content-Based Image Retrieval System with Combined Scene Description Keyword", *Proceedings of the International Conference on Multimedia Computing and Systems*, 1996.

[9] E. Oomoto and K. Tanaka, "OVID: Design and Implementation of a Video-Object Database System", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No.4, 1993.

[10] B. Rubin and G. Davenport, "Structured Content Modeling for Cinematic Information," *SIGCHI Bull.* Vol. 21, No. 2, 1989.

[11] K. Uehara, M. Oe and K. Maehara, "Knowledge Representation, Concept Acquisition and Retrieval of Video Data", *Cooperative Database and Applications*, 1998.

[12] V.V. Vinod and H. Murase, "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks", *Proceedings of the International Conference on Multimedia Computing and Systems*, 1997.

[13] M. Yeung, B.-L. Yeo and B. Liu, "Extracting Story Units from Long Programs for Video Browsing and Navigation", *Proceedings of the International Conference on Multimedia Computing and Systems*, 1996.

[14] H.J. Zhang, et. al, "Video Parsing, Retrieval and Browsing: An integrated and Content-Based Solution", *Proceedings of ACM Multimedia*, 1995.