A Proposal for a Video Content Generation Support System and its Application

Wenli Zhang  Xiaomeng Xu  Shunsuke Kamijo  Masao Sakauchi
Sakauchi Lab, the 3rd Department, Institute of Industrial Science, University of Tokyo

ABSTRACT

A video content generation support system, based on an interactive approach that maps low-level features to high-level concepts, is proposed. By consulting an ontological semantic object model database, the same semantic objects such as characters, backgrounds, and the main subjects in key frames of each video shot can be queried and automatically annotated based on the similarity of low-level features such as the color, area, and position of each region. Since image recognition techniques are limited in their ability to fully identify and compare images, an additional function is proposed, which uses a coarse model to recover a higher number of similar key frames to provide more relevant results. The content provider can then select relevant key frames interactively from the results to annotate matched objects in them according to the descriptions that are added into the model. Therefore, more complex content can be generated with a higher accuracy by using a combination of the application-oriented operations. The system has high potential for use in object-based interactive multimedia applications. One prototype application will also be presented.

1. INTRODUCTION

In recent years, there has been a dramatic increase in the number of videos produced, as a result of advances in digital broadcasting services such as satellite broadcasting, cable TV, and the Internet. Although many multimedia applications have been developed, such as videotape, video-on-demand, virtual reality, and multi-streaming via the Internet, in multimedia applications there is still a lack of a content management capability that allows potential users to handle the video content more effectively. An improved multimedia system is required that includes new technology and tools that can manage and manipulate video content, and even generate new videos. Currently available systems allow video content to be automatically extracted and represented, based on low-level feature-based models [1] [6] [8]. However, while their approach makes use of combinations of several features, it does not attempt at mapping them into high-level concepts, and queries are still based on "query images". Another problem with current systems is that query accuracy is unsatisfactory, owing to the limitations of the image-recognition techniques employed. Systems such as [2], [3], [4], [9], [10], and [11] usually add text annotation such as attributes or keywords either manually or semi-automatically, to represent the high-level concepts of a video’s content. Most of these systems try to annotate all of the concepts of the video’s content. As a result, annotation can become very tedious and time consuming. This paper proposes an integrated approach that interactively maps low-level features to high-level concepts. A video content generation support system is proposed that is based on this approach. By consulting an ontological semantic object model database, the same semantic objects such as characters, backgrounds, and the main subjects in key frames of each video shot can be queried and automatically annotated based on the similarity of low-level features such as the color, area, and position of each region. Since image recognition techniques are limited in their ability to fully identify and compare images, an additional function is proposed, which uses a coarser model to recover a higher number of similar key frames to provide more relevant results. The content provider can then select relevant key frames interactively from the results to annotate matched objects in them according to the descriptions that are added into the model. Therefore, more complex content can be generated with a higher accuracy by using a combination of the application-oriented operations. The system has high potential for use in object-based interactive multimedia applications. The reminder of this paper is organized as follows: Section 2 presents an overview of the proposed system; Section 3 introduces a interactive object annotation approach used for system. Section 4 discusses a prototype application, and Section 5 provides a summary and discusses future works.

2. AN OVERVIEW OF THE PROPOSED SYSTEM

The architecture of the proposed system is shown in Fig. 1.

Figure 1 The Architecture of the System

The system has three layers:

1. The content provider layer, the main functions of which are:
   (a) Video construction, including video segmentation and extraction of low-level features;
   (b) Multi-dimensional indexing for low-level features;
   (c) Semantic object modeling using ontology theory; and
   (d) Object extraction and annotation using interactive approach.

2. The application service provider layer. Application-oriented operations are a combination of various primitive operations. They are used for generating more flexible semantic content for various applications. The application service provider can establish application-dependent services that are based on the output from the content provider layer.
3. The end user layer includes an interface planner that connects the user side and the system side, allowing the user to send a request to the system and the results to be sent from the system to the user.

The main focus of this paper is a human-assisted semantic object extraction and its high-level concepts of description annotation approach, which is used for “content provider layer” Such description assists not only in creating application-oriented operations that enable efficient query and generating new content, but also managing and manipulating individual video clip and object in it. We will also discuss an object-based application called the “drama characters' popularity poll system”, as an example of an application that could be used in the application service provider layer.

3. INTERACTIVE OBJECT ANNOTATION APPROACH

Figure 2 shows the proposed interactive object annotation approach for the content provider layer.

3.1 Semantic Object Model

In general, an object model is a template describing a specific object. During the matching process, each template is inspected to find the “closet” match. In our system, a semantic object has a hierarchy structure shown as Fig.3 (a), from object, salient regions to low-level features. For example, a character model can be treated as being a combination of hair, face and body regions, as shown in Fig.3 (b). The low-level features, such as the color, area size, and position of each region, which are extracted automatically from segments obtained from a key frame using [7].

(a) Semantic Object Structure  
(b) An Example Object Model

Figure 3 Semantic Object Model and its Example

3.2 The Ontology Semantic Object Model

However, when we were establishing the models, we found that the same object was often described in different ways in different videos/shots of the same video. For example, in the case of a character model, the same person can appear in various movies or even in one movie as several different characters or as a character dressed in several different costumes. Therefore, it is necessary to make not only models based on each character, but also models that are based on the same character wearing different costumes. Therefore, our proposed system creates object models based on ontology theory. Ontology theory is a useful theory for intelligent information systems. It is used for conceptualization of an object and for the explicit description of a concept. The models are created based on a movie’s title, the date it was aired, and the characters. The various models are managed by an ontological video content model database, as shown in Figure 4. Tables of typical characters, backgrounds, and subjects are managed by the database.

Two or more models may correspond to the same abstract description. The model management table defines the relational correspondence between keywords and their corresponding models as follows:

\[
\text{keyword} = \text{Object}_{\text{keyword}} = \{\text{Model}_1, \text{Model}_2, ..., \text{Model}_N\}
\]

One model consists of several salient regions as follows:

\[
\text{Model}_i = \{r_1, r_2, ..., r_{N_i}\}, \in \text{templateframe} \in \text{keyframeDB}
\]

where templateframe indicates the various template frames that include the specified characters with different costumes; it can be chosen from key-frame DB. \(r_i\) indicates the selected salient regions of the template image frame that are used in the model.

Once an object has been created, by using the model-matching process, the same objects can be automatically extracted when they are included in other shots, such as a person, a background, or other subjects. Semantic descriptions that match the model can be added to the relevant retrieval results. This region-based model-matching method was proposed in our previous work [15]. Fig.6 (next page) shows the results that were matched with the model shown in Fig.3 (b). By using this method, images containing varying sizes and locations of the query object can be successfully retrieved.

3.3 Interactive Object Annotation Using Coarse Model

Although previous research has expended a significant amount of effort on image recognition, a certain level of fuzziness and imprecision in object recognition is inevitable and needs to be incorporated in the similarity measure in order to increase the success rate of queries and to not exclude good candidates. For this reason, human-assisted method is unavoidable. In this paper, we propose to use a coarse model that returns a higher recall rate to obtain more relevant results, the content provider is then allowed to manually choose the results that he determines relevant, and to annotate them based upon the high-level descriptions added into the model (Fig.2). The coarse model works well if any part of regions agrees with the model-matching conditions. For example, when only a face domain and a body domain agree with a character’s model that includes hair, face
and body domains will match as a possible result. On the other hand, in the strict model, all of the domains, their spatial relationship, the area rates, and the broadcast date, must match [15]. Fig. 5 shows an example of semantic object annotation. Since we retrieve data using regions, we know the coordinates of the minimal bounding rectangles of matched regions, and semantic objects that contain these regions can be determined.

![Figure 5 Example of Semantic Object Annotation](image)

The user can select a positive or negative description. For a positive description, a positive keyword is annotated, e.g., keyword = “Joe”. For a negative description, the positive keyword is negatively annotated, e.g., keyword = “not Joe”.

### 3.4 Experimental Results of Two Models

Table 1 shows an experimental evaluation that was obtained by using these two models. At present, the system stores ten TV dramas, with a total of about 10 hours, 12,998 key frames and 400,000 regions. Precision and recall are used and defined as follows:

\[
\text{Precision} = \frac{|A \cap B|}{|A|} \quad \text{Recall} = \frac{|A \cap B|}{|B|}
\]

where A is the set of all relevant images that contain the query image, and B is the set of retrieved images obtained by the query. The experimental data included nine main characters’ twelve models from six movies. C1, C2, C3, and C6-9 are seven characters from different movies. C4, C5 are the characters from the same movie wearing a different costume, respectively. In general, the characters (ex.C1, C3) who are wearing costumes that contrast with the background will be easier to segment than those (ex.C4-3, C2) wearing costumes of the same color as the background.

![Figure 6 An Interface of Annotation](image)

The annotated information is stored in an inverted file. The inverted file represents the relationship between the description or keyword of the high-level concept of an object and the set of frames that contain the object related to the description, and the object’s position in those frames. It is described as follows,

\[
\text{keyword} \rightarrow F = \{(\text{obj} \_\text{pos}, f_1), (\text{obj} \_\text{pos}, f_2), \ldots, (\text{obj} \_\text{pos}, f_N)\}
\]

where \( i \in [1, N] \). N is the total number of “The Inverted File Library”. In our system, inverted files are stored in an inverted file library as shown in Figure 2. The annotation cost would be prohibitive if we were to interactively describe all of the objects and not lower than 20 percent respectively, a user would be able to choose one result in five for annotation. In our opinion, this is acceptable. The system will be useful if we limit our annotations to the main characters, background, and subjects, using the interactive approach. If a query contains one annotated object and one non-annotated objects, the query accuracy will be higher than both is non-annotated objects. Therefore, we treat this approach as an efficient assist method for the system.

### 3.4 Inverted Files

The annotated information is stored in an inverted file. The inverted file represents the relationship between the description or keyword of the high-level concept of an object and the set of frames that contain the object related to the description, and the object’s position in those frames. It is described as follows,

\[
\text{keyword} \rightarrow F = \{(\text{obj} \_\text{pos}, f_1), (\text{obj} \_\text{pos}, f_2), \ldots, (\text{obj} \_\text{pos}, f_N)\}
\]

where \( i \in [1, N] \). N is the total number of “The Inverted File Library”. In our system, inverted files are stored in an inverted file library as shown in Figure 2. The annotation cost would be prohibitive if we were to interactively describe all of the objects in a video.

### 4. ONE APPLICATION

Since we can know the temporal position (key frame), physical position (minimal bounding rectangle) and high-level concepts description of objects in video content, it became possible to query semantic video content such as “get object name” and “relationship” between the

<table>
<thead>
<tr>
<th>Characters</th>
<th>Strict Model</th>
<th>Coarse Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision (%)</td>
<td>Recall (%)</td>
</tr>
<tr>
<td>C1:Sumie(green)</td>
<td>50.0</td>
<td>30.8</td>
</tr>
<tr>
<td>C2:Takagi(gray)</td>
<td>51.1</td>
<td>42.1</td>
</tr>
<tr>
<td>C3:Yumi(purple)</td>
<td>100.0</td>
<td>27.8</td>
</tr>
<tr>
<td>C4-1:Sutou(skinyellow)</td>
<td>22.1</td>
<td>62.5</td>
</tr>
<tr>
<td>C4-2:Sutou(lightgreen)</td>
<td>77.8</td>
<td>32.6</td>
</tr>
<tr>
<td>C4-3:Sutou(darkgreen)</td>
<td>13.0</td>
<td>16.3</td>
</tr>
<tr>
<td>C5-1:Kindaichi(withhat)</td>
<td>66.7</td>
<td>34.8</td>
</tr>
<tr>
<td>C5-2:Kindaichi_red</td>
<td>80</td>
<td>14.8</td>
</tr>
<tr>
<td>C6:Miyuki(Real)</td>
<td>83.3</td>
<td>11.9</td>
</tr>
<tr>
<td>C7:lijima(Orange)</td>
<td>41.7</td>
<td>89.3</td>
</tr>
<tr>
<td>C8:Ohara(flower)</td>
<td>23.1</td>
<td>50</td>
</tr>
<tr>
<td>C9:Takeda(yellow)</td>
<td>14</td>
<td>42.9</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>51.9</strong></td>
<td><strong>38.0</strong></td>
</tr>
</tbody>
</table>

For color distance =50, about 1/10 HSV space
objects”, such as isLeft(), isRight(), isOverlap(), and etc. As an example, we will now discuss an application-oriented operation to search a name for an object, which is described as follows:

<table>
<thead>
<tr>
<th>Function: getName()</th>
<th>get Name(float X, float Y, float time)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Parameters:</td>
<td>(X,Y) are the coordinates of the object and time is the time-axis value on the video when the user selects the object.</td>
</tr>
<tr>
<td>Output Result:</td>
<td>by searching the inverted file, return the corresponding name of the queried object.</td>
</tr>
</tbody>
</table>

An object-based video content generation application called as “Drama Characters' Popularity Poll System” is now under development using this operation. Java Media Framework kit is being used for operating the raw video. The system consists of a streaming server, which broadcasts the video data via the Internet or network, a video content server, which manages the content data provided by the content provider, and an additional information server, which collects and manages additional information such as vote information. Figure 7 shows the application architecture.

The vote information in the object representing each character is stored in the additional information server. Using the information client interface, the network users can watch the same video and vote for their favorite character. Figure 8 shows a vote interface of the client side. The above operation getName() is used here for specified the clicked female woman’s name when a user watch the video online and vote her.

**Figure 7 “Drama Characters’ Popularity Poll System”**

The vote information is then collected by the server to compute the popularity of the different characters. It could also calculate the popularity of a given character within a certain time zone. Figure 9 shows statistical results of vote-the popularity score of two characters in various shots in the same video.

**Figure 8 The Interface of Vote (End User Side)**

5. CONCLUSION

In this paper, we have proposed a general video content generation support system and introduced a prototype application based on this system. Our future work includes: (a) considering natural language processing for friendly user interface (b) implanting application-oriented methods such as define “relationship” between objects and (c) Implementing interactive object-based multimedia applications.

6. REFERENCES


Figure 9 An example of Vote Statistical Result