VIDEOPOT: INDEXING-BASED DESKTOP VIDEO SEARCH SYSTEM

Hidenobu Nagata, Dan Mikami, Shozo Azuma, and Masashi Morimoto

NTT Cyber-Solutions Laboratories, NTT Corporation
1-1 Hikarino-Oka, Yokosuka-Shi
Kanagawa 239-0847, Japan
+81-46-859-3338
{nagata.hidenobu, mikami.dan, azuma.shozo, morimoto.masashi}@lab.ntt.co.jp

ABSTRACT

We present VideoPot, a desktop video search system designed for digital video files stored on personal computers. The core metric of this system is based on video-indexing technology and an automatic related-term collection framework. Digital files on local disks are crawled, and images of representative scenes are extracted from video files using video-indexing technology. Related terms, used as target information in the video search, are collected from both local files and Web pages using an automatic related-term collection algorithm. These images and terms are then stored in the metadata database. We describe a unique search result interface that helps users distinguish the target video. In the search results interface, the results of the video search are listed at two levels to let users search videos faster. The effectiveness of images on our interface derived from video-indexing is evaluated through a comparison with a conventional file search application.

1. INTRODUCTION

High-capacity HDDs enable us to store a large number of digital video files on our computers. However, we must spend a lot of time searching through videos on our local disks using conventional desktop file search applications. There are two problems when searching videos this way.

1) Search results are obtained using only a list of filenames.

> Conventional desktop file search applications provide search results from a list of filenames, or thumbnails extracted from the first frame of video files [1]. These filenames and thumbnails are seldom useful for identifying the target video, unless the filename accurately describes the video's content, or the user can recall the content from the image in the first frame. Consequently, we often need to replay them to find the scenes that we remember and locate the target video files. Thus, it would be helpful to have a more effective interface for selecting targets from search results [10].

2) The target information is only a filename.

> Users often remember only some of the related terms and not the exact filenames of the target videos. Because filenames are the only target information in conventional desktop file search applications, users cannot obtain any target videos unless they use the exact filenames as queries.

To overcome these problems, we developed VideoPot, an desktop application with an interface designed for video files that have been loaded onto local disks from various sources.

To solve the first problem, we implemented video-indexing technology in VideoPot [7]. Segment images, including audio and visual features in each video file, are extracted with this technology and are then put in the search results. These images are more helpful for figuring out the video's content than images extracted from regularly spaced time series of images or first frame images as is done in conventional applications.

To solve the second problem, we developed an automatic related-term collection framework for VideoPot. Because videos stored on local disks come from various sources (e.g. TV recording, downloading, copying), the ability to handle various metadata is important. Our framework collects related terms of video files from local files and Web pages, and uses them as the target information of video searches on VideoPot.

User interfaces that have an entry of structured key-frame images are important for effective video searches of large video collections [10, 5, 4]. We implemented an intuitive video search interface in VideoPot. The videos in the search results are shown as a list of extracted images, and they appear on two different levels so that users can easily figure out the whole scene and locate the target video in fewer replays.

Section 2 of this paper discusses related work. Section 3 is a system overview of VideoPot. VideoPot's user interface and its evaluation are described in sections 4 and 5. Section 6 summarizes this paper and describes our future work.

1-4244-0367-7/06/$20.00 ©2006 IEEE

2. RELATED WORK

Several papers have addressed the issues related to video indexing based on multiple audio/video features [7]. Relevance feedback technologies and interfaces for effective video searches are recent topics in video retrieval [2, 6, 5, 4]. However, considering the consistency with the user interfaces of popular desktop searches [1], we decided not to use relevance feedbacks in our system. Desktop search systems using metadata based on history of users' actions have been proposed [8, 3]. These studies focus on using the history of accessed Web pages and the context of a file's location in a file-systems for searching files. In our study, we focused on user actions of saving video files on a PC by using local Web cache files to create metadata for the downloaded videos.

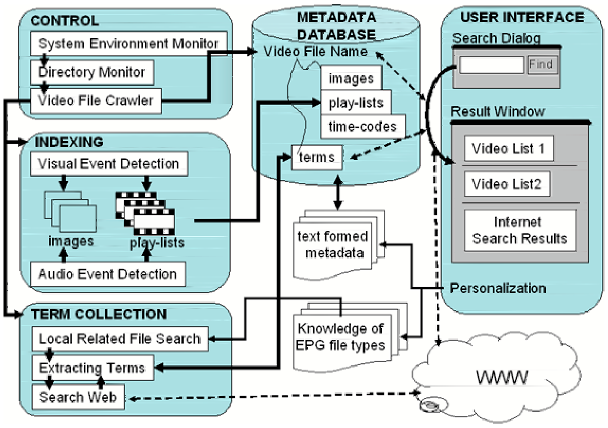


Fig. 1. Program modules in VideoPot.

3. SYSTEM OVERVIEW

Figure 1 shows the structure of the program modules, metadata database, and interface of VideoPot. The following is an overview of the module process.

3.1. Control

Video files in a PC are crawled and the paths to them are stored in the database. New video files are monitored. After a file has been found, it is sent to the video-indexing module and term collection module.

3.2. Video Indexing

Representative scenes are extracted from the videos based on the audio-visual features [7]. Segments of cuts, camera-work, and telop entries are extracted as visual events. Segments of speech and music are extracted as audio events. These events and their time-codes are used to label the images.

3.3. Term Collection

This module collects the related terms to be used as the target information for searching.

For downloaded video files, the HTML files in the Web browser's cache are parsed and the texts in "alt=" attributes of the links of the video files are extracted. For video files recorded using TV recording software, the registered EPG files used to record the program are parsed. A synopsis, channel number, and category names are extracted from the EPGs. File properties to be used in the operating systems are also extracted.

Content-related terms can be also found on Web pages. The terms collected from the local files are used for collecting related terms from the Web. The method of collecting terms from the Web is based on an algorithm of Sato and Sasaki [9]. Our system generates a set of seed terms, derived from the local files by using a tagger program. The Web pages are obtained by using combinations of seed terms as queries, and the terms on the pages are filtered using the following indicator,

$$R_{\wedge/\vee}(s,x) = \frac{Hits(s \wedge x)}{Hits(s \vee x)}$$

$where\ Hits(s)$ = number of Web page hits by term *"s"*

In the above equation, if x is a general term, the indicator R will be reduced. If the relationship between s and x is not strong, R will be small. The threshold value can be set according to the number of filtered terms.

3.4. Metadata Database and User Interface

All images with time-codes and collected terms are associated with the path of the original video files in the metadata database. Users or developers can customize or superimpose the access interface to the database.

4. USER INTERFACE

This section describes VideoPot's video search interface.

One of the most time-consuming tasks for users searching videos with conventional desktop applications is selecting targets while playing video.

We designed the interface of VideoPot to enable more effective video searches. The videos in our search results are listed on two different levels, i.e, in the main results window and in the sub-window shown in Figure 2.

Main results window

The main results window consists of three different sections (a, b, and c in Figure 2). There are two sections for the searched local video files (a and b) and one for the Internet search results (c). The first section lists local video files whose

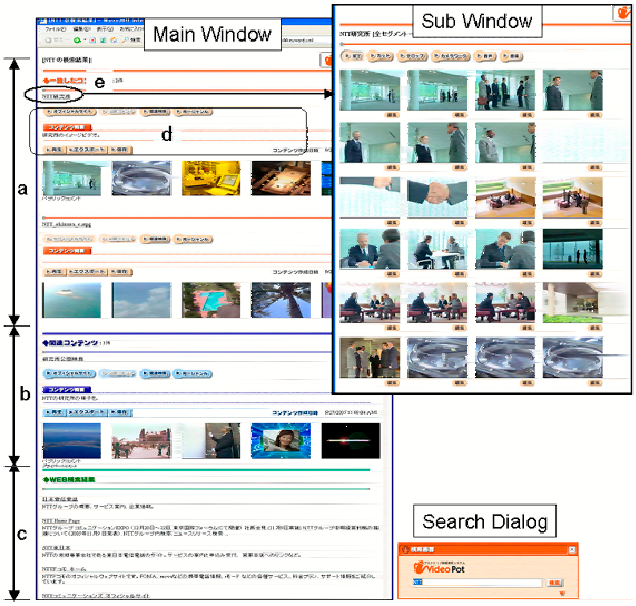


Fig. 2. VideoPot user interface.

file names match the query. The second section lists video files whose database terms match the query. The first (a) and second (b) sections show five images selected according to the difference in the time-codes of the neighboring images together with each filename.

If the registered EPG files are detected, their abstracts appear over the images (d). The user can play short video clips by clicking on the images. Figure 2 shows an example of search results when two files in the first area and one file in the second area appear. The third section displays the Internet search results for the query word. Seamless access to the Internet provides helpful information to the user, even while he or she is searching the local video files.

Sub-Window

Each filename in the main results window has a link to the sub-window (e). The sub-window helps users to distinguish the target video in the search results with fewer playback controls. The sub-window lists all the images extracted from the video.

The images in the sub-window are time aligned and can be filtered based on events by video-indexing using the event selection buttons (see Fig. 2).

Users can easily find specific scenes in a video stream by glancing over the images in the sub-window. By clicking on one of the images, they can play a short clip of the original video file that starts with the segment represented on each image.

5. INTERFACE EVALUATION

We set up VideoPot on a trial PC to evaluate the effectiveness of its video search interface. We focused on the effectiveness of the main window and sub-window when selecting targets. Thus, collected terms were not used in this evaluation.

5.1. Trial Environment

We used a Dimension-1100 (DELL, Pentium-4 2.80 Ghz, 1 GB RAM, 160 GB HDD). The trial video files comprised a total 100 hours of 126 digital video files in various categories (news, dramas, sports, talk-shows, etc.) that were derived from broadcasting in Japan. A running number was used for the video filenames to prevent the subjects from remembering or guessing the video content by filename. After the video-indexing process had finished for the trial video files, VideoPot generated a page listing all the trial videos, and we used it as the start page for our video searches.

5.2. Participants

There were ten subjects (three male; seven female) in their 20s to 30s. All had experience in searching for files on a PC but did not have any experience in using VideoPot. A 60-minute of training lecture on using the VideoPot interface was given to them before the test.

5.3. Video Search Tasks and conditions

We composed 59 different tasks of searching for specific video files or scenes (e.g, home-runs in baseball, stock prices, or the prime minister's speech) in the trial video files. For each task, representative keywords or abstracts of the targets were given to the subjects before they began searching. To scale the apparent workload, we administered a questionnaire on the subjects' feelings about their difficulties after they finished each task. For comparison, the subjects completed the same task set on Microsoft Explorer on WindowsXP. All tasks were randomly assigned to the subjects to avoid task duplication in VideoPot and Explorer. We asked the subjects to keep searching for at least 10 minutes in each task. The task completion time was measured, and the subjects' activities were monitored.

5.4. Results and Discussion

Table 1 shows the average elapsed time per task and incomplete tasks. The averaged elapsed time per task with VideoPot was approximately 88% of the elapsed time with Explorer. The subjects had fewer incomplete tasks using VideoPot.

The subjects seemed to focus on their targets more effectively with VideoPot. Our observations revealed that the

Table 1. Average result per task

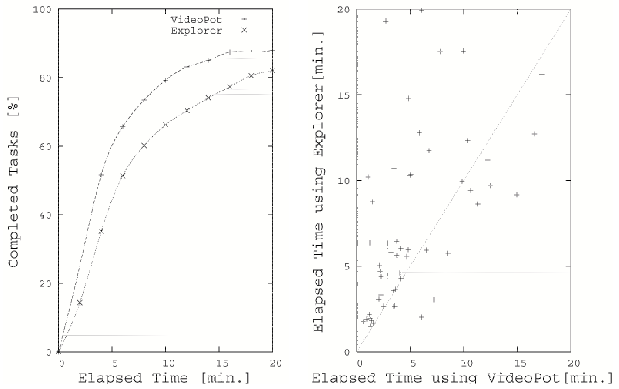
	Elapsed time[sec.]	Incompleted tasks[%]
VideoPot	463	2.90
Explorer	525	12.1

Table 2. Subjects' activities per task

	unique files	play duration[sec.]
VideoPot	4.7	50
Explorer	7.6	154

Table 3. Subjects' feelings about task difficulty

	easy	intermediate	difficult
VideoPot	61%	34%	5%
Explorer	43%	31%	25%



(a) Task completion ratio

(b) Comparison of elapsed times

Fig. 3. Results of video search tasks.

average number of checked unique files and duration of playing videos per task were significantly reduced when they used VideoPot (see Table 2).

Figure 3(a) shows the ratio of completed tasks along with the elapsed time. For example, it took the subjects 10 minutes and 18 minutes to complete 80% of the tasks with VideoPot and Explorer, respectively. This result shows that VideoPot had a better overall task completion ratio.

Figure 3(b) compares the task elapsed times of VideoPot and Explorer for the same tasks. Each sample in the figure indicates a different task, and samples over the dashed line indicate that the task took less time on VideoPot. In over 70% of the tasks, VideoPot was faster than Explorer.

Table 3 shows the breakdown of all questionnaire answers about task difficulties. Note that the same task set was prepared for VideoPot and Explorer. This result indicates that subjects felt they bore less workload when searching with VideoPot.

According to these results, it can be concluded that subjects searched the target videos faster and felt less fatigued when they used VideoPot's interface.

6. CONCLUSION

We presented VideoPot, a prototype system designed to search for video files in personal computers. VideoPot crawls digital video files on the local disk, and then it generates a metadata database that includes images and target information by using video-indexing technology and a related-term collection framework. We designed a video search interface for VideoPot that consists of a main and sub-window of search results. We tested the effectiveness of the interface by having subjects conduct video searching tasks, and found that it helped users find video files faster and with less workload than Explorer did. In our next study, we will evaluate the accuracy of the collected terms and their effectiveness for video searches. We will also look at implementing a new event detection method for video-indexing and self-organizing desktop video files on VideoPot.

7. REFERENCES

[1] Google desktop search, http://desktop.google.com/.

[2] Trec video retrieval evaluation http://www-nlpir.nist.gov/projects/trecvid/.

[3] P. Chirita, R. Gavriloaie, S. Ghita, W. Nejdl, and R. Paiu. Activity based metadata for semantic desktop search. In *Proc. ESWC*, 2005.

[4] M. Christel and N. Moraveji. Finding the right shots: Assessing usability and performance of a digital video library interface. In *Proc. ACM Multimedia*, pages 732–739, 2004.

[5] A. Girgenson, J. Adcock, M. Cooper, and L. Wilcox. Interactive search in large video collections. In *Proc. CHI 2005*, pages 1395–1398, April 2005.

[6] Y. Kinoshita, N. Nitta, and N. Babaguchi. Interactive clustering of video segments for media structuring. In *Proc. ICME2005.*

[7] H. Kuwano, Y. Kon'ya, T. Yamada, and K. Kawazoe. Scenecabinet/live!: Real-time generation of semantic metadata combining media analysis and user interface technologies. In *Proc. IBC2005*, pages 253–260, 2005.

[8] T. Morita, T. Hidaka, T. Kura, K. Ooura, and Y. Kato. Desktop search system based on the action-oriented algorism. In *Proc. APSITT2005*, pages 204–207, 2005.

[9] S. Sato and Y. Sasaki. Automatic collection of related terms from the web. In *Proc. ACL 03*, pages 121–124, 2003.

[10] M. Yang and G. Marchionini. Deciphering visual gist and its implications for video retrieval and interface design. In *Proc. CHI2005*, pages 1877–1880, 2005.