© 2016 IJSRSET | Volume 2 | Issue 3 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Instant Video Search

Anas Aboobacker, Fayaz P A, Sarath Vinoy, Joby George
Department of Computer Science, M. G. University, Kerala, India

ABSTRACT

Mobile video is quickly becoming a mass consumer phenomenon. More and more people are using their smartphones to search and browse video content while on the move. In this paper, we have developed an innovative instant mobile video search system through which users can discover videos by simply pointing their phones at a screen to capture a very few seconds of what they are watching. The system is able to index large-scale video data using a new layered audio-video indexing approach in the cloud, as well as extract light-weight joint audio-video signatures in real time and perform progressive search on mobile devices. Un-like most existing mobile video search applications that simply send the original video query to the cloud, the proposed mobile system is one of the first attempts at instant and progressive video search leveraging the light-weight computing capacity of mobile devices. 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. These images 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.

Keywords: LAVE Indexing

I. 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 file. 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, it 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. 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, developed Instant Video Search, a desktop application with an interface designed for video files that have been loaded onto local disks from various sources. To solve the first problem, implemented video-indexing technology in Instant Video Search. 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, developed an automatic related-term collection framework for Instant Video Search. 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 Instant Video Search. User interfaces that have an entry of structured key-frame images are important for effective video searches of large video collections. Implemented an intuitive video search interface in Instant Video Search. 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. the search query. The system is almost entirely dependent on video frame comparison, and hence the primary step towards the development involves the creation of a system to extract frames from the video files and form a meta-database. We first extract the audio-video descriptors for each video from a large-scale source video dataset. Then, these massive descriptors are indexed by a novel LAVE indexing method. The online query stage consists of the following steps: 1) Capturing query video clips by the user. 2) Sending the query video to the server. 3) Extraction of light-weight audio-video descriptors from the query video. 4) The search process is carried out by comparing the extracted descriptors with the saved ones. 5) A fast and effective geometric verification-based visual re-ranking step is used to refine the search results.

II. METHODS AND MATERIAL

1. Implementation

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the

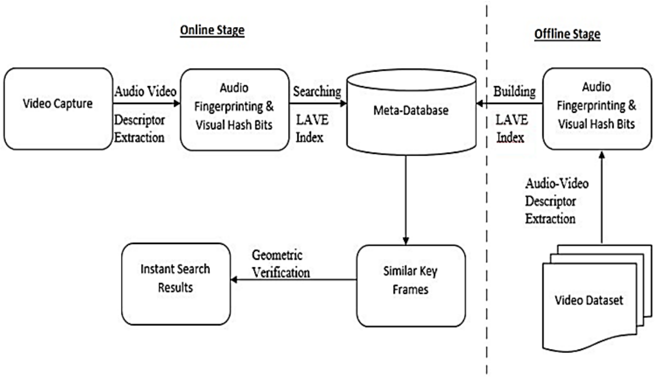

Figure 1: System Flow Chart

2. Proposed System

The system proposed here must consist a personal computer consisting a database of several video files and a mobile device with a camera to capture the video for the search query. The system is almost entirely dependent on video frame comparison, and hence the primary step towards the development involves the creation of a system to extract frames from the video files and form a meta-database. We first extract the audio-video descriptors for each video from a large-scale source video dataset. Then, these massive descriptors are indexed by a novel LAVE indexing method. The online query stage consists of the following steps: 1) Capturing query video clips by the user. 2) Sending the query video to the server. 3) Extraction of light-weight audio-video descriptors from the query video. 4) The search process is carried out by comparing the extracted descriptors with the saved ones. 5) A fast and effective geometric verification-based visual re-ranking step is used to refine the search results.

3. Implementation

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. The system is designed as an independent unit, along with a mobile device having a camera, a processor, and the trained data set. The camera should support high definition capture for optimum performance. Since the system has to be flexible enough to use any time of the day, lack of light must never be an issue. Therefore, a nonintrusive light, like the IR can be provided. The video thus captured, is given as the input to the processor. The processor must be capable of image processing, with at least 1 GB of memory, 300MHz speed and 2 GB RAM.

A. Sample Collection

The system is almost entirely dependent on video frame comparison, and hence the primary step towards the development involves the creation of a system to extract frames from the video files and form a meta-database. We first extract the audio-video descriptors for each video from a large-scale source video dataset. Then, these massive descriptors are indexed by a novel LAVE indexing method. This collection of descriptors is stored in the memory as a meta-database so that it can be used for searching process when a query is given.

B. Joint Audio-Video Descriptors

In contrast to other media, the main advantage of video is that it contains both abundant audio and visual information. As the complementary nature of the audio and video signals, the joint audio-video descriptors are more robust to the large variance of query videos, especially for complex mobile video capturing conditions (e.g., silent video or blurred video of low visual quality). Selecting effective audio-video descriptors is very important for Instant Video Search. There are three main principles for joint descriptor selection: 1) robust to the variance of the recorded query videos, 2) cheap to compute on mobile devices, and 3) easy to index for instant search.

C. LAVE Indexing

Even though the computing of the Hamming distance between binary audio and visual features is efficient, linear search for a big video dataset is still a bottleneck in real-time mobile video search. The most widely used binary feature indexing method is locality sensitive hashing (LSH). However, as it uses random selection, aim to achieve good search quality, it needs a large number of hash bits to build enough tables. In our system, we propose LAVE indexing. There are two layers in the LAVE. The first layer is the index entry, containing a multi-index: audio indexing and visual indexing. The second layer is the visual hash bits, which are used for accurate feature matching and combination. There are two advantages to these structures: 1) effectively employing the hierarchical decomposition strategy to improve the visual points search speed, and 2) holistically exploiting the complementary nature of audio and video signals. The different indexing entries in the first layer preserve the individual structure of audio and video signatures. In the second layer, the combination of audio and video can be weighted by the hamming distance of visual hash bits.

III. RESULTS AND DISCUSSION

1. Implementation Procedure

LAVE indexing is the main component of the Instant Video Search system. The whole search process depends on LAVE Indexing. There are two layers in the LAVE. The first layer is the index entry, containing a multi-index: audio indexing and visual indexing. The second layer is the visual hash bits, which are used for accurate feature matching and combination. There are two advantages to these structures: 1) effectively employing the hierarchical decomposition strategy to improve the visual points search speed, and 2) holistically exploiting the complementary nature of audio and video signals. The different indexing entries in the first layer preserve the individual structure of audio and video signatures. In the second layer, the combination of audio and video can be weighted by the hamming distance of visual hash bits.

A. **Building LAVE Index**

In contrast to the visual feature, the audio feature is highly compressed, only 25 bits for each point. Therefore, a linear search of the audio index can be quickly completed. We use the audio index as part of the first layer and each bucket in the audio index of the first layer is associated with the second layer by the video ID, audio time offset ta and key frame number t^v. Through the audio indexing, we can refine the number of visual points to be searched in the second layer, which obviously improves the search speed. However, if the audio information is changed significantly or missed, it will be difficult to find the closest neighbour in the second layer. The hash bits from the second layer visual index are indexed by m different hash tables, which construct the visual index of the first layer. The hash bits $h^{sub}{}_n$ of the visual index in the first layer are randomly selected from the hash bits in the second layer. For a received visual point, entries that fall close to the query in at least one such substring are considered neighbour candidates. The candidates are then checked for validity using the second layer index. We have m + 1 multi-indexes: visual indexing and audio indexing. Finally, all the results are fused and the top N results are returned. With the help of the audio index, we can greatly reduce the number m for the hash table. In our experiments, even one hash table can still work very well.

B. **Searching LAVE Index**

The search process in the LAVE indexing is presented as follows. Let $P_a = \{l_1, l_2, . . . , l_M\}$ be the received audio query points and $P_v = \{v_1, v_2, . . . , v_L\}$ be the received visual query points. Through the search process, the top K visual points will be returned for each query visual point.

1) For each audio point l_m in P_a, the nearest approximate neighbors will be acquired by a linear search in the audio index. Then the matching pairs are assigned to different candidate clusters $C = \{c_1, c_2, . . . , c_N\}$. Two pairs are assigned to the same cluster if their nearest approximate neighbors come from the same video.
2) All the clusters will be reordered by temporal verification. Here we define the temporal distance Δt to denote the time difference of the two LBAFs in the matching pairs. The histogram of Δt is computed for all pairs in c_n and the score of c_n equals h_n/M, where h_n is the maximum value of the histogram. Then the top K′ candidate clusters are chosen. All the buckets associated with the top K′ candidate clusters in the second layer are regarded as a subset
3) For each v_l in P_v, the K nearest approximate neighbors are obtained as follows:
   i) Top K approximate neighbors are determined by linear search in the subset of the second layer.
   ii) Use the multi-index indexing method to search other top K nearest neighbor points.
   iii) The 2K nearest neighbor points are reordered by similar distance. The top K nearest points are selected.
4) Finally, the top K nearest visual points are returned as the search results.

In summary, according to the process, we combine the audio and video in two stages. The first stage is Step 1 – Step3.a. In this stage, we use the higher compressed audio information as the crude filter and the more discriminative visual information as the fine filter, which obviously improves the final search speed. Furthermore, as the similarity is computed in separate layers, the combination stage also preserves the individual structure of each signature. The second stage is Step 3.b – Step 4. In contrast to the first combination stage, which heavily depends on audio search accuracy, the combination of audio and video can be weighted by the hamming distance of visual hash bits. The two combination stages holistically exploit the complementary nature of the audio and video signals for more robust mobile video search. Finally, as we have m + 1 multi-index, i.e., m visual indexes and one audio index, the computational complexity of searching the LAVE index mainly depends on the multi-index indexing method used to search the nearest visual neighbor points.

C. Geometric Verification

With the top N points, the Hough Transfer method is used to get similar source key frames of the query. Further-more, a subsequent geometric verification (GV) considering the spatial consistency of local features is needed to reject the false-positive matches. In order to reduce the time consumption of GV, we follow a fast and effective GV based re-ranking step to find the most similar image. As the method only utilizes the orientation of descriptors, there is no need to transmit the location information of the local features by the network. First, the method hypothesizes two matched descriptors of duplicate images should have the same orientation difference. So for two duplicate images, the orientation distance $\Delta\theta_d$ between each matched local feature pair is calculated. Then all $\Delta\theta_d$ are quantized into C bins (C = 10). Furthermore, the histogram is scanned for a peak and the global orientation difference is set as the peak value. Finally, the geometric verification score is given by the number of the pairs in the peak, which is normalized by the number of total pairs.

IV. CONCLUSION

Instant Video Search, a prototype system designed to search for video files in personal computers. Instant video search 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. In this, we have investigated the possibility of instant video search on mobile devices, where a very short phone-captured video clip is used as the query to identify the captured video from a large-scale video database. 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. 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 desk top video files on Instant Video Search.

V. REFERENCES

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

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

[3] 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.

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