

Collaborative Browsing and Search in Video Archives with Mobile Clients

Claudiu Cobârzan, Manfred Del Fabro, and Klaus Schoeffmann

Alpen-Adria-Universität Klagenfurt
9020 Klagenfurt, Austria
{claudiu, manfred, ks}@itec.aau.at

Abstract. A system for collaborative browsing and search within video archives is proposed. It comprises of mobile clients and a back-end server. The server is responsible for inter-client communication as well as for archive partitioning according to the active client’s population. The participating clients employ a GUI designed and optimized for Nexus 7 tablets.

1 Introduction

Exploring a video archive with either highly specialized tools or simple video players is a hard task for expert and non-expert users alike. Most tools, either high-end or simplistic, disregard any form of collaborative features or provide only limited ones. Building on our previous experience with collaborative video browsing and search on mobile clients [2], we propose a new system that tries to enhance the collaborative aspects of exploring a video archive, taking advantage of the high availability and popularity of tablet computers. While our previous tool focused on individual video exploration, the current system is aimed at archives and tries to exploit the power of collaboration in solving a specific search task. That translates in a divide and conquer approach when it comes to data exploration: a server system is responsible for the coordination of a collaborative search task and partitions the archive equally among all mobile clients participating in a search task at any moment in time. Primarily, each client only concentrates on assigned data, but can also explore the rest of the archive. Clients can reduce the search space by sending filtering requests based on color, motion, and face detection to the server. Filter settings as well as filtering results can be manually shared among clients (on request). State information, e.g. video segments currently being explored, the degree of exploration, etc., is automatically exchanged between clients in real time.

2 System Architecture

The proposed architecture employs mobile clients, a server as well as a database. The server is responsible for the coordination of search tasks as well as for the communication between the clients. The clients communicate only with the

server via Wi-Fi using a proprietary UDP based protocol. Pull and push techniques are combined in order to perform different tasks like selecting a video segment for search, iterating through key frames, setting filters and retrieving the corresponding results, sharing filters as well as key frames sets, etc. Details regarding the client’s as well as the server’s implementation are presented in the following subsections.

2.1 Server

The main tasks of the server are as follows: (a) assign and update the search space for each active client taking into consideration the client activity (new clients entering the system, clients leaving the system); (b) collect status information from the active clients (selected video, selected segment, selected keyframe) and share it among all participants; (c) retrieve query results according to client’s filtering requests; (c) ensure message passing among clients in order to support collaborative operations.

The search archive is held by the server which also holds the results of offline keyframe selection, content analysis and filtering within a proprietary database. The server assigns segments from the video archive to each client entering the system. The client is responsible for searching within those segments but it is able to perform specific operation in all available segments either at will or in response to another client’s specific request (share a search filter, respectively share a keyframe set). We consider segments of fixed lengths, namely 15 minutes of playback video.

Our keyframe selection approach operates on a sub-shot level and is able to create several keyframes for shots with varying content over time, such as camera pans, rotates, zooms etc. This will ensure that – in contrary to common shot detection algorithms [6] – we will not miss any important content, even for shots of longer duration.

The keyframe selection method consists of three stages. First, it inspects the optical flow [1] over successive frames. The algorithm starts with a densely sampled grid of keypoints in the first frame and tries to track these keypoints over consecutive frames. Keypoints that can’t be tracked anymore are removed and if the ratio of “still trackable” keypoints falls below a certain threshold, the current frame is selected as keyframe candidate and the optical flow estimation starts with a fresh grid of keypoints. In the second stage, the neighborhood of such candidate keyframes is further inspected and only the sharpest frame in this neighborhood is selected as keyframe. We use a Difference-of-Gaussian (DoG) approach to determine the blurriness (i.e., sharpness) of a frame. Finally, in the third stage consecutive keyframes are checked for their visual similarity by a simple histogram comparison. Such consecutive keyframes that are too similar are removed from the keyframe set.

For the content analysis part, we use a tool for determining the dominant color as well as the color layout for each previously detected keyframe. We also employ a face detector in order to detect keyframes containing 1, 2, 3 or more faces. Motion is detected at segment level. For color analysis we use

the MPEG-7 color layout descriptor [4]. When filtering for a certain dominant color, all segments containing keyframes that match the dominant color will be returned.

For face detection we use boosted cascades of Haar-like features [7]. We employ the classifiers that ship with the OpenCV library¹. Faces are detected and counted on all the previously obtained keyframes.

For motion analysis we use the motion histogram proposed in [5], which is applied on segment level. Both color-based and motion-based segment detection have been successfully applied to search tasks of the Video Browser Show-down 2013 [3].

2.2 Mobile Clients

The clients joining the system are automatically assigned a portion of the archive, which is searched in a way that is intended to ensure fairness (e.g. if there are only 2 active clients, each will have assigned half of the video collection - the first user will search within the first half of the collection, while the second will search within the second half). They can browse not only the assigned archive part but the entire archive as well. They can also issue search requests and then browse throughout the results. Each client will begin to search within the default allocated segments. If the allocated segments do not match the search request, a client can also collaborate with another client by searching within the segments assigned to that client.

The interface is divided into multiple task specific areas:

- *target segment selection, segment assignment and current activity overview area* (see Figure 1D): allows the users to have an overview of the whole archive as well as on the parts that are assigned to each individual user. The list of active users is presented on the left side. Each active user is assigned a number (U1 for user 1, U2 for user 2, etc.) and a color for identification. The same color is used to mark the segments that have been assigned to him by the server. At the finest granularity, each square represents a segment of 15 minutes of playback video. The arrow buttons on the right side of the area, as well as the plus and minus buttons are used to navigate within the archive and to modify the current viewpoint by increasing/decreasing the virtual segmentation granularity of the video archive. Decreasing the segment granularity level has a zoom-out effect: each two consecutive segments from the previous granularity level are merged into a new “virtual” segment (e.g. two 15 minutes segments will be combined to form a new 30 minutes segment).
- *segment exploration area*(see Figure 1B): when a user selects a segment from the target selection area (Figure 1D), the keyframes comprised within that particular segment start being displayed as thumbnail pictures. All the

¹ <http://opencv.org/> (last access 2014-09-25)

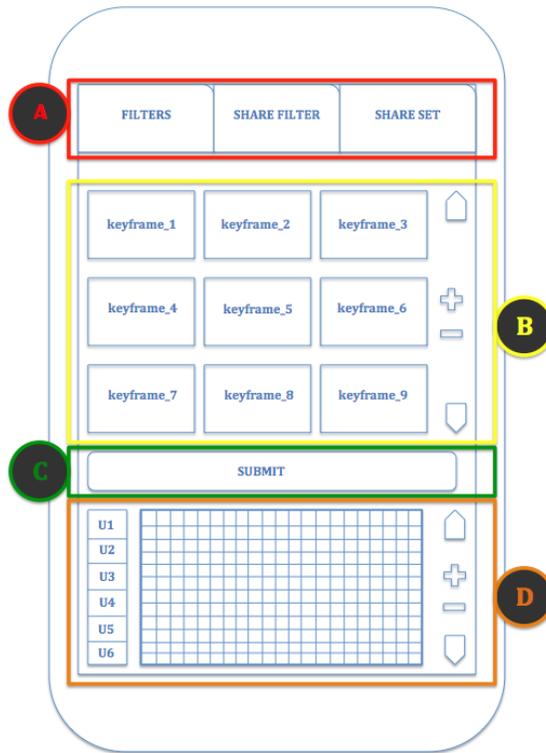


Fig. 1. Sketch of the client interface.

keyframes from one particular segment form what we call a *keyframe set*. The number of keyframes that are displayed at one particular moment can be modified by pressing the plus/minus button on the right hand side. The arrow buttons are used for navigation within the keyframe set in a page-by-page manner (e.g. the first 9 images in the set, the next 9 images when the down arrow is pressed and so on). The displayed keyframes act also as entry points within the corresponding video: double tapping a keyframe opens a basic video player with minimum controls (play/stop button and a seeker bar) whose current playback position is set to match the selected keyframe. Pressing the SUBMIT button (Figure 1C) sends the current frame to the server for checking whether it belongs to the target scene or not.

- *query construction area* (see Figure 2): by pressing the FILTERS button (Figure 1A), a user is able to build a filter that can include either single or multiple components that are grouped into 3 categories: (1) faces, (2) color and (3) motion (Figure 2A). A user can filter for keyframes containing 1, 2, 3 or more faces. One can also indicate the desired dominant color as well as the direction and speed of motion. All filters are applied

on the content-based analysis of keyframes and videos segments, which is performed on the server in advance. A user can combine filters from the 3 filter groups. Pressing the down arrow button (Figure 2B) adds the current selected filter to the filter composition bar (Figure 2C) and numbers it. Once added to the filter composition bar, a filter can be repositioned with respect to previously added filters or can be removed by pressing the corresponding buttons. A filter is applied by pressing the SUBMIT FOR PROCESSING button (Figure 2D). The filtering request is sent to the server and the corresponding segments/keyframes are returned. The results are highlighted in the overview area (Figure 1D) against the whole video archive. If segments with matching keyframes exist in the result set, the corresponding keyframes are automatically displayed in the segment exploration area (Figure 1B). If a filtering request returns no match within the assigned partition of the dataset, a segment assigned to another client can be chosen manually.

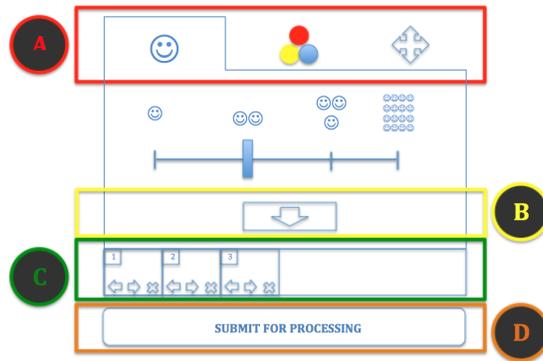


Fig. 2. Sketch of the *query construction area* (face filtering section).

The collaborative aspect of the proposed system includes sharing the most recent built filter with other users, sharing the current keyframes dataset as well as being able to view in real-time the segments that are being explored by the other participants. All clients can ask for help in exploring the results of one particular filter by pressing the SHARE FILTER button (Figure 1A). This translates in a notification being sent to all participants. The ones that accept the notification will have the segments returned as filter results highlighted in the corresponding area (Figure 1D). The server also shares among all participants the current segment being explored by each individual user. Those segments are correspondingly highlighted in the activity overview area (Figure 1D). Segments whose keyframes have been partially or full viewed in the segment exploration area are also highlighted. This allows collaborative clients to join a search task and focus mainly on unexplored or only partially explored segments as well as

monitor the current browse activity within the archive. Similarly, help can be requested when exploring a *keyframe set* that comprises all the frames from a selected segment by pressing the SHARE SET button (Figure 1A). Clients that accept the request will have the keyframes of the set displayed in the segment exploration area (Figure 1B).

3 Conclusion

We presented a new collaborative video browsing and search system targeted on archive exploration for the Video Search Showcase 2015 competition. Similar to our previous approach [2], we concentrate on mobile clients and try to explore and take advantage of the interaction capabilities of the most recent generation of tablets. The focus is on client, server and communication features as well as data organization that enhance the collaborative aspects of video exploration within large video archives.

Acknowledgments

This work was funded by the Federal Ministry for Transport, Innovation and Technology (bmvit) and the Austrian Science Fund (FWF): TRP 273-N15 and by Lakeside Labs GmbH, Klagenfurt, Austria, and funding from the European Regional Development Fund (ERDF) and the Carinthian Economic Promotion Fund (KWF) under grant KWF-20214/22573/33955.

References

1. Bouguet, J.-Y.: Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker Description of the Algorithm. Intel Corporation. 5, 1-10 (2001)
2. Cobârzan, C., Hudelist, M.A., Del Fabro, M.: Content-Based Video Browsing with Collaborating Mobile Clients. In: C. Gurrin et al. (Eds.) MultiMedia Modeling, 20th Anniversary International Conference, LNCS, vol. 8326, pp. 402–406 Springer (2014)
3. Del Fabro, M., Bernd, M., Böszörmenyi, L.: Smart Video Browsing with Augmented Navigation Bars. In: S. Li et al. (Eds.) Advances in Multimedia Modeling, LNCS, vol. 7733, pp. 88–98, Springer, Heidelberg (2013)
4. Kasutani, E., Yamada, A.: The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In: Proceedings. 2001 International Conference on Image Processing. IEEE, Thessaloniki (2001)
5. Schoeffmann, K., Lux, M., Taschwer, M., Böszörmenyi, L.: Visualization of Video Motion in Context of Video Browsing. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME). IEEE, New York (2009)
6. Smeaton, A. F., Over, P., Doherty, A. R.: Video shot boundary detection: Seven years of TRECVID activity. In: Computer Vision and Image Understanding, vol 114, (2010)
7. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Kauai (2001)