An End-to-End Framework for Multi-view Video Content: Creating Multiple-Perspective Hypervideo to View on Mobile Platforms

Gregor Miller, Sidney Fels, Michael Ilich, Martin M. Finke,
Thomas Bauer, Kelvie Wong, and Stefanie Mueller

Human Communication Technologies Laboratory,
Department of Electrical and Computer Engineering,
University of British Columbia
{gregor,ssfels,michaeli,martinf}@ece.ubc.ca,
privat@boweh.de, kelvie@ieee.org,
stefanie.mueller@student.hpi.uni-potsdam.de
http://hct.ece.ubc.ca

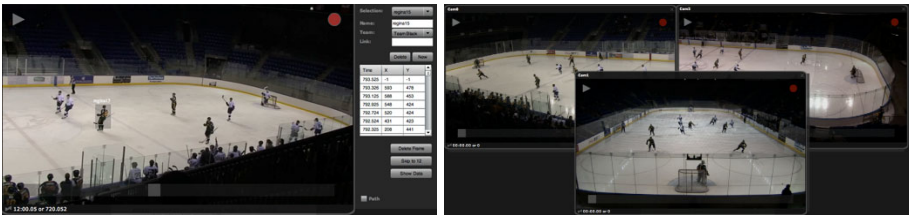
Abstract. We present a work-in-progress novel framework for the creation, delivery and viewing of multi-view hypermedia intended for mobile platforms. We utilize abstractions over creation and delivery of content and a unified language scheme (through XML) for communication between components. The delivery mechanism incorporates server-side processing to allow inclusion of additional features such as computer vision-based analysis or visual effects. Multi-view video is streamed live to mobile devices which offer several mechanisms for viewing hypermedia and perspective selection.

1 Introduction

This paper presents a novel framework for production, delivery and consumption of multi-view hypervideo targeted towards mobile devices. Each of these three components are separate projects within our framework and each represents a current work-in-progress infrastructure with the aim of providing modular and simply connected systems. The novelty of our work comes from the combination of these components targeted towards multi-view and mobile platforms, with the addition of computer vision infrastructure to enhance the user experience.

The framework we present here includes many components which are tied together. We introduce a mobile networked camera system which can address cameras uniquely and stream data from them to any platform. To organize and annotate the generated footage we present our authoring tool for multi-view hypermedia along with our semantic description we use as the base language of the video metadata. The image and semantic data is processed on servers and then delivered to mobile devices over a transport middleware developed specifically for image processing and transfer. Finally, we describe our mobile platform multi-view hypermedia viewers which receive the data and present it in an intuitive manner to the user.

J. Anacleto et al. (Eds.): ICEC 2011, LNCS 6972, pp. 337–342, 2011.



(a) Object annotation editor (b) Multi-view video annotation display

Fig. 1. Within-video objects annotated with position and identity information (a), supporting multi-view annotation (b)

2 Related Work

Most related work to this area is in video authoring, annotation tools and hypermedia viewers, since complete frameworks are rare. Annotations in video visually emphasize objects in a video stream (shown as a bounding box in Figure 1) and by serving as an anchor between objects and additional information. Annotated videos are also often referred to as hypervideo [9], which resolves the linearity of video structures and so creates a non-linear information space. Among others, HyperCafe [8], HyperSoap [3], and HyperFilm [7] are considered to be the first research projects that integrate video annotation as a core concept. All of these projects focus on a separation of the presentation and authoring component and so omit functionalities to support collaboration needed in a multi-user environment. Since then various commercial products have surfaced such as VideoClix [10] or ADIVI [1] presenting annotated video content over the Internet for single use only. VideoClix is unique in that it does not visualize the annotations in the video players. Users have to actively search for annotations by moving the mouse pointer over possible video objects of interest. The ADIVI system, however, visualizes all annotations in the video player by means of a half transparent shape and a solid frame. Both applications allow users only to interact with the annotations and so to perceive additional information but not to create their own annotation in the Web based video player. Support for multi-view video content is not integrated in both user interface concepts.

3 Mobile Multi-view Video

The work presented in the following sections describes the formation of a framework for creating, annotating, processing, delivering and viewing of multi-view video data designed for consumption on mobile platforms. We have defined a camera abstraction framework for access to video streams from many imaging devices; our hypermedia creation tool can annotate multi-view video with spatio-temporal information such as location within frame or a frame-based event (e.g. a goal in a hockey game). The footage, either live or annotated, is sent across the network over our transport abstraction architecture either for further processing (e.g. background subtraction or player tracking) or immediate delivery to the mobile platform for viewing. Our mobile platforms receive the multi-view footage and present it to the

user. Through a variety of view and object selection mechanisms the user can navigate the footage in a meaningful way.

We intend our framework to incorporate video from many different sources; we have developed an abstraction over camera and video types for simple access on any platform to live or pre-recorded image data. The Unified Camera Framework (UCF) [5], is a camera access scheme for uniform camera access and configuration. We have created an abstraction layer over image and camera access which supports existing standards, and can support new cameras through its extensible framework. In order to provide an access system with the necessary level of abstraction, UCF contains: a generic image description; uniform access to cameras across platforms; configuration of all cameras up to their capabilities; and an addressing scheme to allow access to any camera on the network.

3.1 Hypermedia Creation

Our research domain currently focuses on annotation and viewing of ice hockey games filmed from multiple angles. This provides a rich video data set and generalizes to many other types of multi-camera annotation and viewing contexts. The use of many cameras is common in sports broadcasting, to ensure a good view of the various players and the array of different actions.

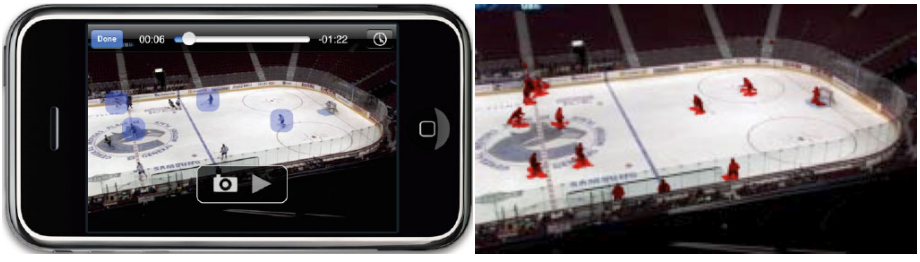
We imagine that environments such as ours will be critical for support of future hypermedia. Interactive, annotated video allows users to share particular content in a scene without requiring the notion of a shared video clip. In effect, annotated video provides anchors that provide means of non-linearly moving through spatio-temporal video content. For example, in sports, this functionality supports the following example functions:

1. Users can specify that a particular object should remain in view so that cameras angles are automatically adjusted to keep a player in view
2. Links to hypertext data such as statistics associated with the selected player
3. Users can send hypervideo anchors to indicate a particular event, so the
4. entire clip need not be viewed
5. Users can view the event from a particular perspective

Authoring operates in two modes - object selected and object not selected. The main difference is the visualization of active areas uses bounding boxes to give more information to the user annotating the video; once the user has selected an object to annotate (or created a new one) all other active areas are hidden and only the current object is visible, to avoid click confusion, a state where the active area visualizations hide the video and make it harder for the user to see the current object.

Multi-view video is also changed from a single Video Player with view switching to a multiple Video Player interface so that more than one view can be seen simultaneously. This provides the user with different perspectives from which they can disambiguate players when occluded in the active view. Each Video Player has its own editable Information View to provide information on the object.

Semantic Description. The description of multi-view video content for rich media application underpins the framework we have designed. We have several requirements of the description: 1) Incremental updates; 2) Temporally variable and static content; and 3) Multiple platform/system support.



(a) iPhone Video Player (b) Nokia Video Player

Fig. 2. The iPhone (a) and Nokia (b) rich media players for multi-view video. The iPhone within-video objects are highlighted with relatively large icons for simpler touch access. The Nokia version uses background masks cropped with the annotated bounding boxes to highlight individual objects, which can be selected using the 4-way joystick on the phone.

The semantic description we provide uses a hierarchical structure to emphasize the nature of items belonging to larger components e.g. frames belonging to cameras. The format uses a top-level node (Session) to indicate which event or part of the event the data corresponds to; all other nodes are held within this one. There are then three other high-level nodes to partition the logical parts of the data: Cameras, which contains information on the devices used to capture the event as well as information corresponding to data collected by the devices, such as Frames; Scene, a description of the event as a whole which allows for three dimensional locations to be defined, and other scene information; and finally Objects, which holds all temporally constant information about the event (such as a hockey player's name or number).

3.2 Processing and Delivery

In our framework images and content are optionally processed by servers and sent across the network to the mobile viewing platforms. This allows us to add visual effects, use computer vision and also resample and compress the images to be more suitable for transfer to mobile devices. We use the Hive [2] layered architecture transport middleware to accomplish both the processing and delivery in a single step.

Hive was created to help simplify distributed processing and development of reusable modules. A Hive system consists of a number of drones which are connected into one or more swarms by an application. The term drone is used to describe a device or service which uses Hive for communication and is remotely configurable.

Drones can also be connected together to form a processing pipeline (swarm) [4]. Configuration and connection commands are issued by applications to set up a swarm to accomplish a specific task. Applications can construct multiple swarms in order to perform various complicated tasks simultaneously then collate the results. Applications and drones are both Hive modules.

Due to the abstraction layers over each task in the Hive architecture, we can substitute different implementations for the layers. For this paper we wrote separate layers for mobile devices, mainly due to the separate platforms and differing APIs, but this also allowed us to create new a mode for the Hive drones and applications: single process execution of the Hive communication system.

(a) iPhone View Flow (b) iPhone View Grid

Fig. 3. There are three view selection methods on the mobile video player. Both mobile versions support instant switching (left or right) to an adjacent view. Our iPhone version also supports View Flow (a), which is similar to Cover Flow in other iPhone applications, except our version uses video thumbnails. The second mode (b) is the View Grid which lays out all the available views in a grid.

3.3 Mobile Viewing

We have developed multi-view hypermedia viewers for the iPhone and Nokia devices. Both use object visualization for interaction, and provide methods for switching views. The iPhone version is a more complete system due to the simpler developer platform and advanced graphics [6]. Viewing of hypermedia is not an intuitive operation and so we define some requirements our system must fulfill:

Visualization. The problem is that visual annotations cover the regions they follow which in turn might confuse the information the video content conveys. We can conclude that one requirement for a hypermedia viewer is to visualize annotations with a minimum of user distraction in the way they perceive the video information. The Nokia player in Figure 2(b) demonstrates our use of background masks highlighting the players in a minimally obtrusive way (the red is emphasized for demonstration purposes).

Interaction. The dimension of an annotation's sensitive regions must be large enough in respect to their movement to enable users to easily interact with them. The larger selection boxes in Figure 2(a) show the increased target area for interaction. These boxes are only visible when selection mode is activated, so that when viewing the video they are not obscuring content.

Multi-View Video. The novelty of our system is partly defined through the integration of multi-view video content such as for sport events. There are different concepts for how to support the switching of views; Regardless of the method, we require a mechanism to assist users to retain orientation between views: a system enabling an arbitrary switch between views would risk users losing their sense of orientation and might be unable to follow the game properly.

4 Conclusion

We have presented a work-in-progress framework for the creation, delivery and consumption of multi-view hypermedia, with targeted application in sports

broad- casts. Our framework connects components created for video capture, authoring and annotation of multi-view video, distributed processing and delivery of images with semantic data, and viewing of multi-view hypermedia on two mobile platforms. We are working on a unified architecture for viewing platforms such that each platform should look and feel the same, and we are developing intuitive mental models for interaction with hypermedia on mobile devices.

References

1. Add Digital Information to Video (October 2009), http://www.adivi.net/
2. Afrah, A., Miller, G., Parks, D., Finke, M., Fels, S.: Hive: A distributed system for vision processing. In: Proc. of the Int. Conf. on Distributed Smart Cameras, pp. 1–9 (September 2008)
3. Bove, M., Dakss, J., Agamanolis, S., Chalom, E.: Hyperlinked television research at the mit media laboratory. IBM System 39 (2000)
4. Miller, G., Afrah, A., Fels, S.: Rapid vision application development using hive. In: Proc. Conference on Computer Vision Theory and Applications (2009)
5. Miller, G., Fels, S.: Uniform access to the cameraverse. In: International Conference on Distributed Smart Cameras. IEEE, Los Alamitos (2010)
6. Miller, G., Fels, S., Finke, M., Motz, W., Eagleston, W., Eagleston, C.: Minidiver: A novel mobile media playback interface for rich video content on an iphone. In: Proc. International Conference on Entertainment Computing (September 2009)
7. Pollone, M., Rusconi, M., Tua, R.: From hyper-film to hyper-web: The challeng- ing continuation of a european project. In: Electronic Imaging and the Visual Arts Conference, Florence, Italy (2002)
8. Sawhney, N., Balcom, D., Smith, I.: Hypercafe: Narrative and aethetic properties of hypervideo. In: ACM Conference on Hypertext. ACM, New York (1996)
9. Sawhney, N., Balcom, D., Smith, I.: Authoring and navigating video in space and time: An approach towards hypervideo. IEEE Multimedia (October 1997)
10. VideoClix Authoring Software (October 2009), http://www.videoclix.tv/