DAVVI: A Prototype for the Next Generation Multimedia Entertainment Platform†

Dag Johansen[1,2], Håvard Johansen[1], Tjalve Aarflot[1], Joseph Hurley[1], Åge Kvalnes[1],
Cathal Gurrin[1,5], Sorin Sav[5], Bjørn Olstad[2,3], Erik Aaberg[4],
Tore Endestad[7], Haakon Riiser[7], Carsten Griwodz[7,8] , Pål Halvorsen[7,8]
[1]University of Tromsø [2]Fast Search & Transfer [3]Microsoft [4]Schibsted
[5]Dublin College University [7]University of Oslo [8]Simula Research Laboratory

ABSTRACT

In this demo, we present *DAVVI*, a prototype of the next generation multimedia entertainment platform. It delivers multi-quality video content in a torrent-similar way like known systems from Move Networks, Microsoft and Apple do. However, it also provides a brand new, personalized user experience. Through applied search, personalization and recommendation technologies, end-users can efficiently search and retrieve highlights and combine arbitrary events in a customized manner using drag and drop. The created playlists of video segments are then delivered back to the system to improve future search and recommendation results. Here, we demonstrate this system using a soccer example.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Video

General Terms

Design, Experimentation, Performance

Keywords

Video search and recommendation, annotation, personalization, torrent-like dissemination

1. INTRODUCTION

Currently, there are many services delivering video over the Internet ranging from live event streaming replacing the traditional television broadcasting to uploading, sharing and delivery of personalized content. Many people therefore think the time is ripe for radical changes in entertainment and news production systems in order to integrate emerging Internet technologies and services frequently used by the new generation of multimedia consumers.

To meet the requirements of users familiar with services like Facebook, YouTube, blogs, etc. and to improve search and personalization functionality, we are developing part of the next generation multimedia technology. In the context of Schibsted, one of Europe's largest media companies, we are prototyping a next generation system for search and recommendation soccer service. Our system, called *DAVVI*, aims to provide a personalized, topic-based user experience blurring the distinction between content producers and consumers. This requires a closer integration of existing video delivery systems, search and recommendation systems, and social networking systems. The current prototype has implemented the first two, i.e., efficient video playout and video search and recommendation. In this demo, we demonstrate our system in a soccer scenario.

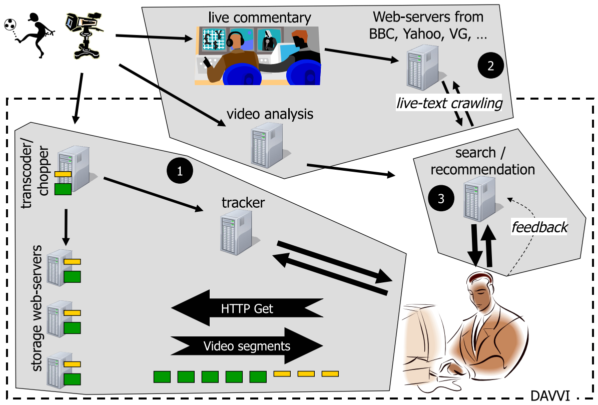


Figure 1: Architecture

2. DAVVI

The DAVVI components developed can be broadly categorized in three different classes. They are ❶ segmentation & dissemination; ❷ annotation; and ❸ search & recommendation. An overview of the architecture and how the different components interact is given in figure 1.

2.1 Segmentation & dissemination

To scale streaming servers and achieve load balancing, a lot of work has proposed streaming from multiple sources. Furthermore, the idea of using discrete media object units (segments) has for a long time been successfully applied to simplify computations. Since such systems have been proven very scalable [2], we aim for a similar architecture.

As shown in figure 1, the input video streams are first loaded into a video segmenter and transcoder. Based on experiments, we use two-second segments which is efficient both with respect to coding efficiency and search granularity. Adaption to available resources and reduced latency when jumping within or between streams are achieved by coding each segment in several qualities (different colors in figure 1). Moreover, to be able to retrieve arbitrary segments and continuously play them out in any order, each segment must be a self-contained video, i.e., formats including a meta-data header only in the start of the stream cannot be used. Our system can accommodate other formats, but

†Sponsored by the iAD research based innovation centre, project 174867.

MM'09, October 19–24, 2009, Beijing, China.
ACM 978-1-60558-608-3/09/10.

we currently use closed group of pictures using H.264 video and MP3 audio. To reduce start-up and jump latencies, due to timestamp problems with libavcodec which we use for encoding and because of large MPEG transport stream container overheads, the media data is packed in a custom made container reducing the bit rate with about 10%.

To disseminate the multi-quality video streams, the segments are distributed to web-servers out in the Internet, and we use a torrent-like approach over HTTP for data delivery. Such approaches have recently gained new interest with systems like Move Networks [2], Microsoft's Smooth Streaming [4] and Apple's HTTP Live streaming [6]. Thus, video segments are downloaded using HTTP GET requests, and different segments might the be retrieved from different geographical locations, i.e., a tracker holds up-to-date information about the individual segment locations.

Finally, in the end-system, our efficient, 2000-code-line player (compared to traditional large, all-eating players) can seamlessly jump from one quality level to another. The player optimizations give really fast startups and jumps (see table 1)[1] when combined with retrieving the small segments in a low-rate first and switching to a higher quality if enough time and bandwidth.

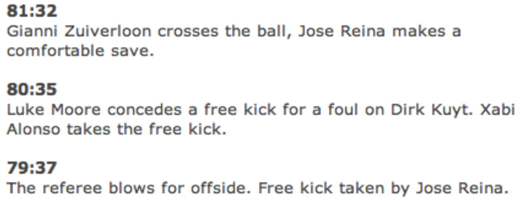
System (storage location)	Latency (ms)
SmoothHD (demo [4], Akamai Oslo)	320, 533, 880
SmoothHD (Tromsø/arctic Norway)	400, 956, 1500
Move Networks (demo [2]), Level 3, California)	560, 716, 800
Move Networks (VG live [5], Level 3, California)	640, 1022, 1400
DAVVI (Tromsø/arctic Norway)	280, 328, 360
DAVVI (Oslo)	320, 328, 360

Table 1: Jump/interaction latencies for various systems measured from Oslo (min, avg, max)

2.2 Metadata and annotation

To analyze and annotate the videos, we have developed several event extraction tools. These tools identify events like goals, crowds cheering, etc. However, this process is complex, not yet accurate enough and resource demanding.

On the other hand, there exists a lot of related, but untaped metadata information on the Internet. A large number of TV broadcasting and news paper cites provide services like *live text commentary web pages* [1, 3, 5] which easily are retrieved on the fly and processed in real-time. For example, for each game covered by VG Live [5], Schibsted has two people giving live text commentaries. As shown in figure 2, a detailed, minute-by-minute textual information about the game is presented. To convert this unstructured commentary text to annotation metadata, we currently have a semi-automatic crawler parsing these pages. Thus, our users can search and query for game events using a much richer set of keywords, e.g., particular names and phrases like "volley", "sliding tackle", "blocking", "offside" and "heading". We then use the given time stamps to identify the respective video interval and retrieve data some seconds before and after the annotated time. The most complex task in this operation is synchronizing the coarse-granular, out-of-sync textual timestamps with the video objects. However, tools have been implemented for this purpose, and the event extraction tools are used to improve the default event intervals generated by the web crawling and text annotations.



81:32
Gianni Zuiverloon crosses the ball, Jose Reina makes a comfortable save.

80:35
Luke Moore concedes a free kick for a foul on Dirk Kuyt. Xabi Alonso takes the free kick.

79:37
The referee blows for offside. Free kick taken by Jose Reina.

Figure 2: BBC Live Text Commentary snippet [1]

2.3 Search & recommendation

Based on the metadata aggregation, video annotation and indexing, users can query for a broad range of events using keywords found in the live text commentary. For example, in the soccer scenario, a user can query for "sliding tackles by Steven Gerrard". The video annotations are then used by the search engine to return a playlist in the form of an event description, video object identifier and a time interval (and several other metadata descriptors). This playlist is sent to the video dissemination system which retrieves the respective video segments. In order to enable low latency start-ups and fast jumping to the next events, the system can retrieve the first segments of the first events in the playlist to create bookmarks. However, the current prototype is jumping fast enough (see section 2.1) so this feature is by default off.

The playlist is presented to the user as a list of thumbnails taken from the corresponding video interval and traditional metadata like which game it is taken from, i.e., date, teams and result, and with a mouse-over-thumbnail function, the user is presented the live text commentary. The search results may be played back one by one clicking on the individual events, using a play-all button or by generating a personalized video sequence using drag-and-drop. Finally, to further improve ranking in future search and recommendations, information about which events are really played back is fed back to the search engine.

3. DEMO

In this demo, we present DAVVI. We demonstrate how video segments are retrieved from several geographical locations, the fast jumping using the search bar and the seamless switching between multi-quality video. We will also bring some examples of not yet processed live commentary text for on-site parsing and indexing, and show the effect of the search results including this information. Finally, we will demonstrate the search and recommendation functionality with the drag-and-drop composition of a personalized video.

4. REFERENCES

[1] BBC SPORT | Football | Premier League | Live Scores. http://news.bbc.co.uk/sport2/hi/football/eng_prem/-live_scores/.

[2] Move Networks. http://www.movenetworks.com/.

[3] Premier League Football | Premiership Results & Fixtures - Yahoo! Eurosport UK. http://uk.eurosport.yahoo.com/football/premier-league/.

[4] SmoothHD. http://www.smoothhd.com/.

[5] VG Nett Live. http://vglive.no/.

[6] R. Pantos (ed). HTTP Live Streaming. http://www.ietf.org/-internet-drafts/draft-pantos-http-live-streaming-00.txt.

[1] Notice that load and propagation delay from storage may differ, but the average RTTs are far less than the differences.