# Context-Aware Scalable Multimedia Content Delivery Platform for Heterogeneous Mobile Devices

*Kwong Huang Goh, Jo Yew Tham and Tianxi Zhang*
Institute for Infocomm Research, A*STAR, Singapore
{khgoh, jytham, tzhang@i2r.a-star.edu.sg}

*Timo Laakko*
VTT Technical Research Centre of Finland, Finland
{Timo.Laakko@vtt.fi}

*Abstract*—The vision of making any multimedia content to be always accessible to every mobile user over the network is great challenge. There are two major obstacles. First, the network capacity limits the amount and the quality of multimedia content that can be made available to every user at each time instant. Secondly, it is extremely difficult for the user to browse and search for the desired content from the huge multimedia library that is available. The work in this paper aims to address these obstacles to a certain extent. In order to overcome the adverse effect of the bandwidth limitation on the quality and quantity of the delivered multimedia content, we propose using the SVC and adaptive layered streaming approach. The second problem is addressed by utilizing context-aware personal content adaptation and efficient metadata processing to reduce the burden of user navigation in the large pool of media content. A proof-of-concept prototype that integrates the two technologies was developed. Cross country test trials were conducted to demonstrate the capabilities and practical use cases of the integrated context-aware scalable multimedia content delivery system for heterogeneous mobile devices.

***Index Terms*—Context, Scalable, Multimedia, Delivery**

## I. INTRODUCTION

Mobile devices have become a common and essential commodity for everyone. In recent years, multimedia features are being integrated into mobile devices. Unsurprisingly, the demand for mobile multimedia content and services has been on the rise.

### A. The Desired User Experience

For an ideal user experience, any multimedia content should be readily accessible on-demand over the network at anytime. From the user perspective, it is an essential requirement to have as smooth multimedia services as possible, e.g., based on his/her personal contextual habits independently from the applied heterogeneous delivery channel. This is a difficult task with the currently popular video encoding and streaming technologies, i.e., TCP/RTP streaming of H.264 videos. This is because H.264 video stream does not allow bitstream truncation for adaptation. Therefore, in order to cater for different network bandwidth and playback device capabilities, multiple copies of a single video has to be generated. An example is the different resolution options available at YouTube and Apple Movie Trailers. However, the conditions and quality of the delivery channel can change in the duration of a video stream. When this happens, current available technology is not able to automatically upgrade or downgrade the bitstream rate for improved video quality playback.

### B. Scalable Video Streaming for Heterogeneous Devices

Different mobile devices have different processing and display capabilities. Moreover, the same device model is likely to have different bandwidth constraints which depend on the user subscription and the network conditions. Given such heterogeneous conditions of the mobile devices, it is necessary to have custom encoded video streams (in terms quality and rate) to cover for different possible device and environment settings in order to achieve optimal viewing experience. However, this is near impossible with the currently available video coding and streaming technologies in the market. Current video encoding and streaming technologies, such as TCP and RTP streaming of H.264 encoded videos, would have to encode and stream these different quality video streams separately and hence the huge transmission bandwidth is required for all heterogeneous devices. Furthermore, the content management is also tedious for encoding and maintenance of different video quality streams.

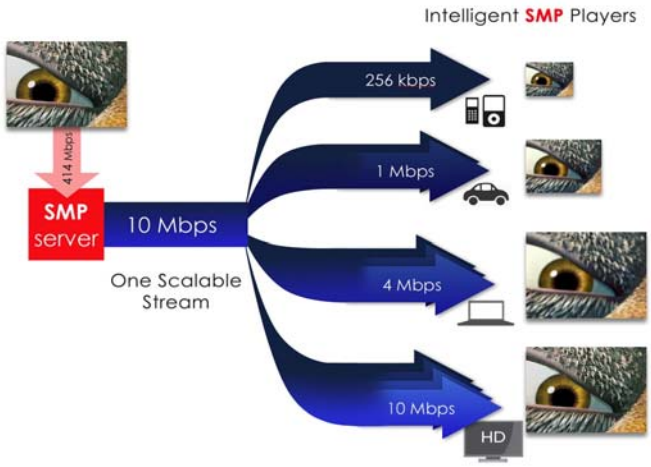


Figure 1. Advantages of scalable video coding & streaming

In this work, we use scalable video coding (SVC) which is an extension of the H.264/AVC standard [1][2][3]. Figure 1 below illustrates the advantages of SVC for heterogeneous streaming. With SVC, a scalable stream can provide adaptively different numbers of video layers to heterogeneous clients, according to the client's processing capability and available bandwidth. In terms of content management, only one-time encoding of each of the video content is required and hence simplified the content management process. Some other related work can be found in [4].

We have integrated the context awareness aspects for positive content viewing experience and the scalable

 ISBN: 978-1-61208-129-8

multimedia content delivery platform, to build a context-aware scalable multimedia **Co**ntent **De**livery platform for heterogeneous mobile devices (CoDe).

The next section of this paper describes the context discovery and personalization of the integrated platform and section III describes the end-to-end scalable multimedia platform from encoding, streaming to decoding and playback. The integrated platform test is described in section IV followed by the conclusions.

## II. Context Discovery and Personalization

Personalization is based on the stored and semantically refined context information of the user. The user preferences are included into the delivery context [5]. Personalization aims to increase the acceptance of the set of information. It helps the user to get relevant content in the current situation. A platform describing personal preferences in each context is developed. The semantic of context information is used as a basis for adaptation and personalization.

### *A. Context-Aware Server*

A user context may contain parts such as: spatio-temporal (place, time), environment, personal, task, social. User context information is derived from lower level context information. A low-level context is composed from different sources (sensors, network connection, user preferences, user agent profile etc), for example, measuring location, 3D acceleration, vibration, time, etc. The context information can also be given explicitly.

The location context is fetched from the GPS (outdoor), Cellular ID (indoor) or WLAN hotspots (indoor). In Cellular ID based positioning, ID of the used base station is sent to the server, where it is searched from the list of base stations and its location is returned. The accuracy of cell ID based positioning is inferior to GPS positioning, but it consumes less battery resource. Similar method is used in the WLAN hotspot detection; the phone scans for unique Basic Service Set Identifiers (BSSID) of available WLAN access points, which are then compared to the predefined list of known WLAN hotspots.

The technical context consists of device properties, such as display size, user agent, compatible formats, battery life, available space and other capabilities and limitations. Static information about the user agent could be collected from UAPROF header included in devices HTTP requests or during the registration of the device. Dynamic information, such as battery status information, should be updated periodically to the context module. Network context keeps up the information about available connection types to adapt the provided content in the most suitable format. Context module could also take advantage of external online context sources such as weather service or global calendar service.

### *B. Service Personalization*

Personalization service retrieves user context and context history information from context management services. A user profile contains information about the user for personalization. Personalization module helps the user to get relevant content and services in the current situation. Table 1 shows the context information used for adaptation and personalization.

| Context Information used for Personalization | |
|---|---|
| Context Data | Used in Movie Recommendation |
| Gender | Yes |
| Age | Yes |
| Language | Yes |
| Interest | Yes |
| Country | Yes |
| Screen | No |
| Time of the day | Yes |
| Time of the week | Yes |
| Network | No |
| Free Time | Yes |
| Mood | Yes |
| Activity | Yes |
| Location | Yes |

Table 1. Context information used for service personalization

### *C. Media Content Analysis, Tagging and Retrieval*

Media content needs to be analysed properly in order to get it utilized appropriately. Content analysis also includes tools for content management, and it takes into account content duration information, numbers of scalable layers encoded, scalable resolutions available, the content genre information, etc. Before the server retrieves relevant content for the specific user, content analysis is required to find out the useful personalized information, which can be user's age, gender, interest, language, etc. Based on this user information, a relationship between multimedia content and user profile can be built up and saved in server's database. Therefore, whenever and wherever the user wants to get their interested multimedia content, the server can satisfy their requirement by simply retrieving relevant information from database.

In order to facilitate the user's search for the multimedia content created by them, tagging mechanism could be used to assist the implementation of personalization. For instance, when user creates a new video, he can assign text-based tags to it which can facilitate video searching via tags. Content analysis can realize the service personalization and tagging can optimize the dynamic freedom of the system.

## III. Scalable Multimedia Platform (SMP)

The scalable multimedia platform (SMP) developed in this work, as depicted in figure 2, is the integration of transcoding non-scalable media content into scalable media content; live layered streaming; server content management; and mobile client device decoders, into a next generation mobile entertainment solution. The SMP scalable content comprises of a base layer and several enhancement layers. Depending on the client's capabilities, only the appropriate audio/video layers or sub-streams will be abstracted from the same copy of

 ISBN: 978-1-61208-129-8

the scalable media content for real-time delivery from the SMP server to the client.

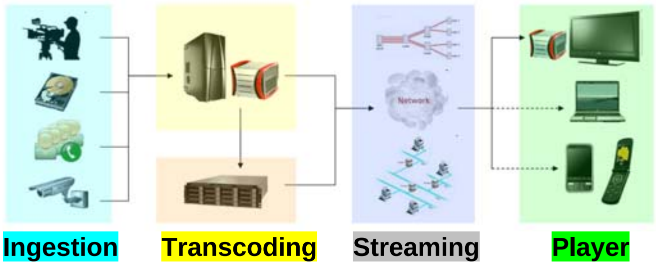


Figure 2. Scalable Multimedia Platform (SMP)

In general, the scalable video content can support scaling in spatial resolutions, bit-rate, and visual quality. The following sub-sections will further describe the details of the main modules of the SMP used in this paper.

### *A. The Media Content Transcoding Module*

The content transcoding module converts a video from any supported compressed format to an ISO/IEC standard-compliant SVC video bitstream with AAC-LC (Advanced Audio Coding, low complexity profile) audio bitstream. Both the video and audio bitstream are multiplexed into the standard MP4 file container format (MPEG-4 Part 14 or ISO/IEC 14496-14). All media content that is stored in the video library is represented using the above scalable video format. The SVC encoder makes use of fast algorithms in [6]. The media track of the SVC compressed video in the MP4 file format is hinted accordingly to support several spatial and temporal resolutions [7]. This module provides the required application interface (API) for specifying the desired scalable encoding parameters of each media file ingested.

### *B. The Scalable Streaming Server*

The streaming server consists of the scalable video streaming module which reads in a particular scalable media file from the video library and streams it in an instant-on-demand mode to the requesting client player. It employs the Real-Time Streaming Protocol (RTSP) with RTP over UDP for the media delivery. A different client player can simultaneously request from the streaming server either the same or a different media stream for playback. A unique feature of this module is that it allows the streaming server to automatically tailor the scalable video stream delivery to each of the requesting client player. This module is responsible for the automated selection of scalable media sub-streams for real-time delivery to the requesting client player.

### *C. Client Decoding and Player*

The client decoding and player SDK comprises of the media buffering and decoding module, and the media streaming and adaptation module. The scalable media buffering and decoding module enables smooth media sub-streams management and decoding of the media for playback. This module also ensures robust networked media delivery and error concealment [8]. The scalable media decoder can also be deployed as a Microsoft's DirectShow filter plug-ins for the windows media player.

The scalable media streaming and adaptation module enables real-time reception of scalable media sub-streams requested from the Server System. It employs the Real-Time Streaming Protocol (RTSP) with RTP over UDP for the media delivery. The client player may request a different version of the media file depending on its own current capabilities such as the available processing power and resolution of the display device. This module automatically adjusts by requesting only the pertinent media sub-streams from the streaming server for delivery and playback on the client player.

## IV. INTEGRATED CODE PLATFORM FOR TEST TRIALS

Figure 3 illustrates the CoDe's service-oriented architectural design between the clients and servers. It highlights the main client-side and server-side modules, together with their software implementation interfaces, communication/network protocols, operation system, and programming language environments. A desktop GUI application that was first developed using Nokia's Qt C++ language. The codes were portable on Windows, Linux, Mac OS as well as Windows Mobile operating systems. The demo application for the streaming test trial is Video-on-Demand (VoD). Subsequent applications such as News-on-Demand or live broadcast events can be added. The client retrieves the VoD metadata from the VoD server backend via Web Services. The Web Server is running Microsoft IIS and exposes the VoD application interfaces through WSDL. The VoD application server is running on Windows Server OS, and operates on Windows J2E framework. The VoD metadata is stored persistently on the MySQL Server.

Multimedia content source is compressed into SVC (Scalable Video Coding) with AAC (Advanced Audio Coding) standard formats, and stored as a hinted MP4 file container format that can be readily streamed via a RTSP/RTP streaming server (such as the Darwin Streaming Server). The compressed content is stored in a file server. The streaming server communicates with the client player via the Scalable Multimedia Platform Protocol (SMPP), which is an extended version of the standard Real-Time Streaming Protocol (RTSP). It is responsible for establishing the hand-shaking with the client player to exchange information about the media file, and for setting up a media session for packet-based streaming via the Real-Time Protocol (RTP) over User Datagram Protocol (UDP). The SMPP further supports dynamic media stream adaptation between client and server.

Client GUI for VoD application was demonstrated during the streaming test trial for Windows based laptop and mobile phones such as Windows-Mobile based smart phones and iPhone. For client-side scalable video playback, each PC/notebook client was installed with the relevant plug-ins, namely the SVC decoder plug-in and SMP protocol (SMPP) streaming plug-in. These plug-in was developed using the Microsoft DirectShow architecture. The plug-ins was integrated into the VoD GUI desktop application (via Nokia's Qt-Phonon framework) or it can also be embedded into a media player (such as the Windows Media Player) that is integrated into a web page. For mobile phone clients such as iPhone, web-based browser for the VoD was used and video streaming is performed via HTTP streaming of the SVC base layer to the iPhone's H.264 player. For Windows Mobile based phone, the PC-based GUI was ported to the Windows-Mobile OS for the VoD trial.

 ISBN: 978-1-61208-129-8 

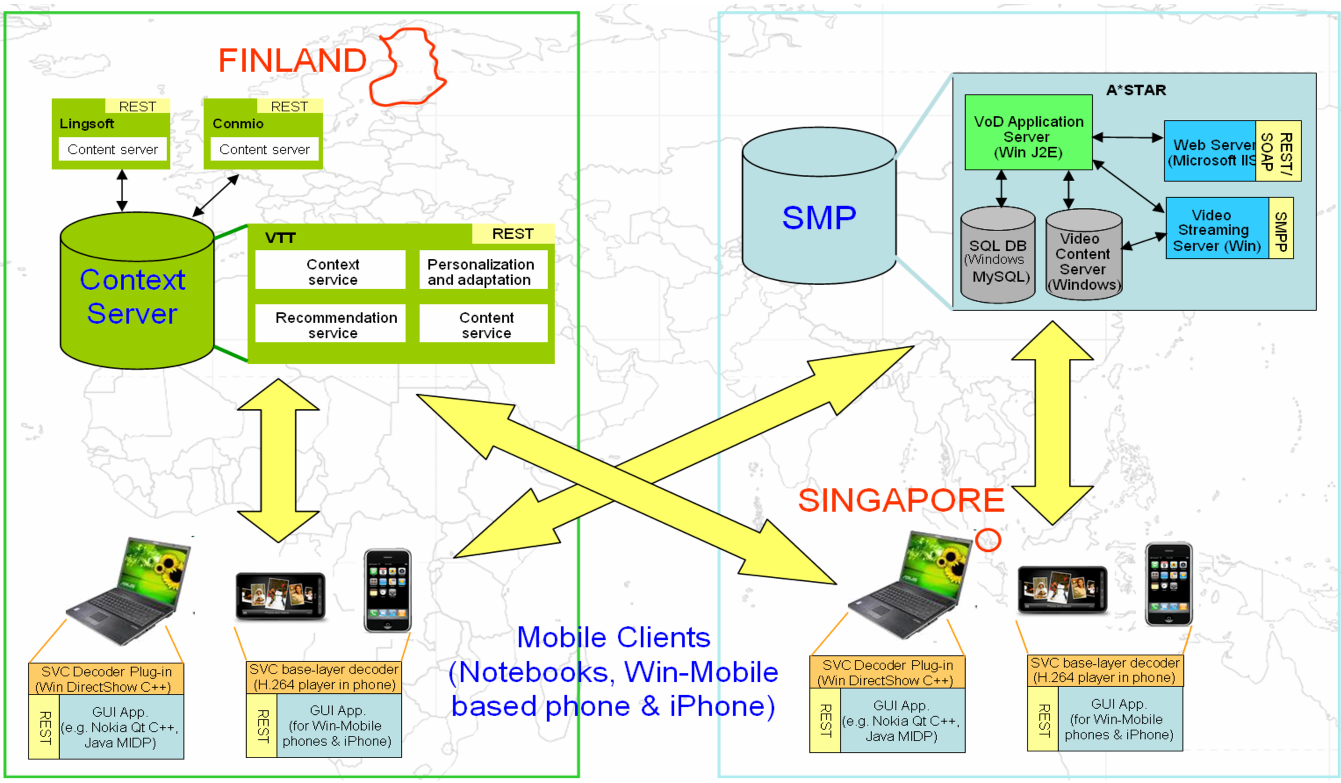


Figure 3. CoDe's service-oriented architectural platform and geographical common trials

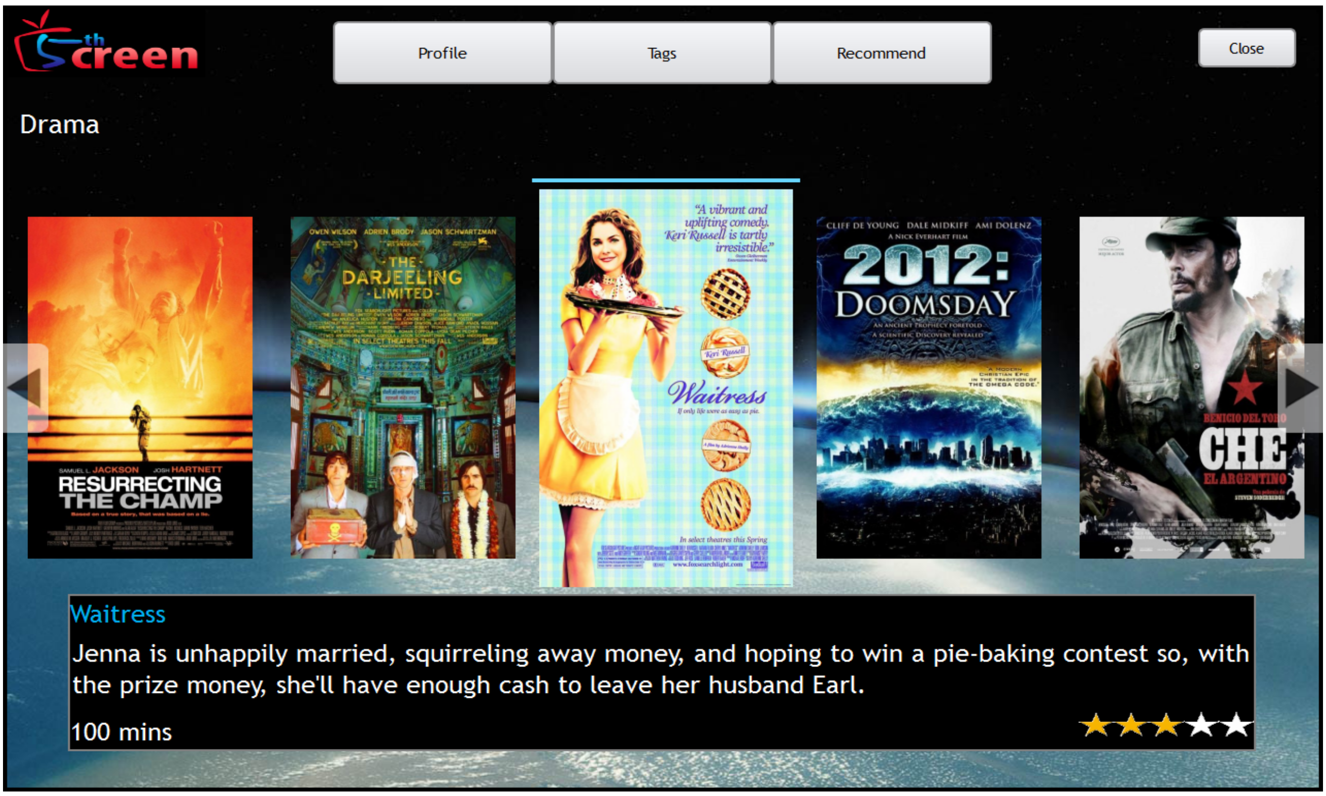


Figure 4. PC client GUI's application named as 5thScreen, showing the context-aware recommended

 ISBN: 978-1-61208-129-8 

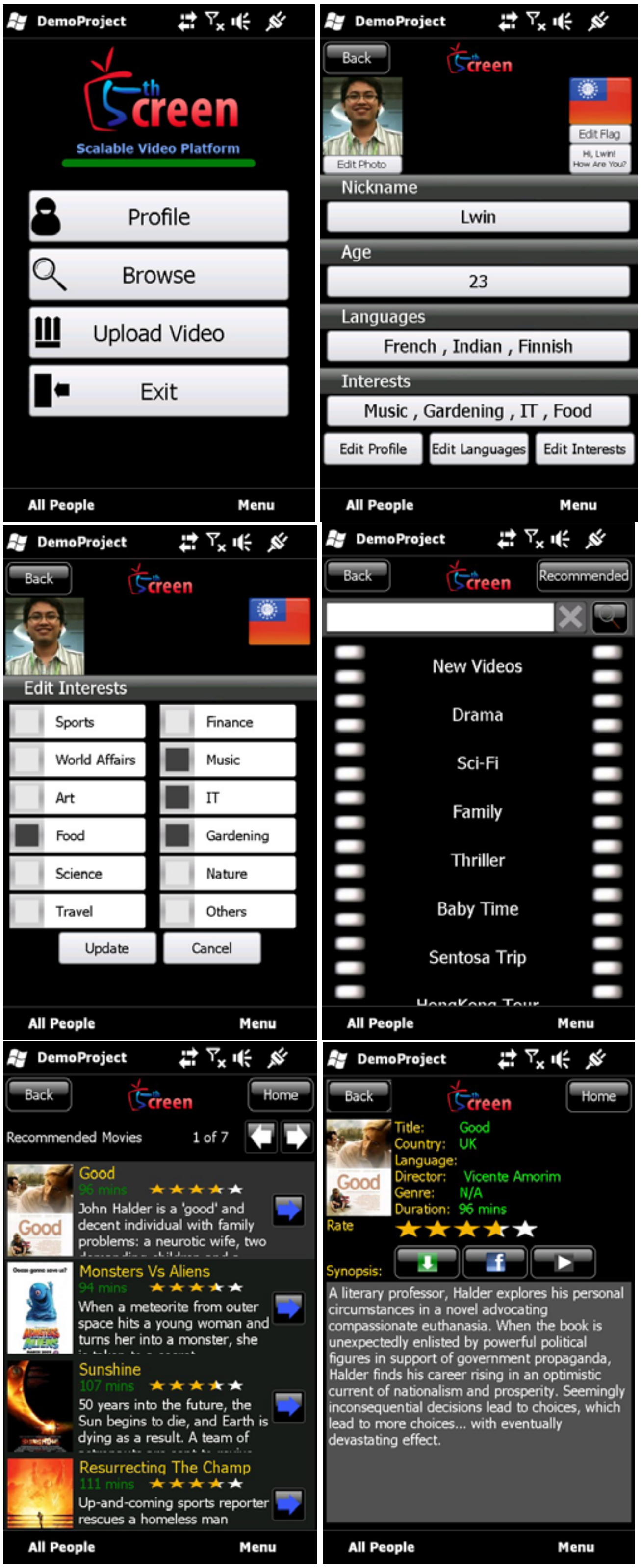


Figure 5. Smart phone VoD application GUI, from top- left: main page, top-right: profile page, middle-left: interest-edit page, middle-right: video tag-list, bottom-left: video browser page and bottom-right: video info page

The context server was located in Finland whereas the scalable multimedia platform was located in Singapore. The cross-country context-aware scalable media streaming test trial has been successfully conducted in both Finland and Singapore. The context client in the laptop and the smart phones communicate with the context server located in Finland to obtain the context information. With the context information the client VoD application then communicate with the database server in Singapore to obtain the list of recommended list of movies for viewing. With information of the client devices such as the CPU, network type and screen sizes, the client can then make request to the streaming server located in Singapore for the suitable scalable video layers to be streamed to the client. Generally, with the context information, the client is able to make good recommendation; and the video streaming is smooth with the correct choice of scalable layer being streamed. Figure 4 shows the laptop client GUI's recommendation of movie list in a browser based on the context-aware information. Figure 5 shows six selected screen shots of the smart phones GUI.

Figure 6 shows the live context-aware scalable media streaming, in which the laptop is streaming the base layer plus 1 resolution enhancement layer via the internet, whereas the smart phone is streaming only the base layer via 3G connection, from the same scalable file stored at the streaming server.

Figure 6. Demo picture of the cross-country context-aware scalable media streaming test

## V. CONCLUSIONS

An integrated context-aware scalable multimedia content delivery for heterogeneous mobile systems is developed and trial-tested for cross-country content streaming. The proof-of-concept prototype of a context-aware scalable media delivery for heterogeneous devices has shown good context-aware use-cases with video streaming for best possible quality under the constraints of client device capability, network conditions and user preferences.

The current proof-of-concept platform only makes use of the context information for video recommendation service and

 ISBN: 978-1-61208-129-8

to decide at the client side the number of scalable layers to be streamed. Full video streaming adaptation, i.e., on-the-fly adaptation to network conditions with error resilience and concealment, is yet to be integrated. The amount of context information used is also quite limited. Future work will address these limitations.

## REFERENCES


[1] JSVM software. Available from CVS repository. :pserver:jvtuser@garcon.ient.rwth-aachen.de:/cvs/. (Last access in Jun 2010).

[2] ITU-T Rec. H.264jISO/IEC 14496-10. Advanced video coding for generic audio-visual services, 2005.

[3] Schwarz, H., Marpe, D., and Wiegand, T.,: "Overview of the scalable video coding extension of the H.264/AVC standard", IEEE Trans. CSVT. v17 i9. pp. 1103-1120.

[4] D. Pichon, P. Seite, and JM. Bonnin, "Context-aware delivery of video content to mobile users", ACM Mobility Conference, ISBN. 978-1-60558-536-9, Nice, France, Sep 2009.

[5] Laakko, T.: Context-Aware Web Content Adaptation for Mobile User Agents. R. Nayak, et al. (Eds.): Evolution of the Web in Artificial Intel. Environ., SCI 130, Springer-Verlag, pp. 69–99, 2008.

[6] W. S. Lee, Y. H. Tan, J. Y. Tham, K. H. Goh, and D. J. Wu: "LACING: An improved motion estimation framework for scalable video coding", ACM International Multimedia Conference, Vancouver, Canada, pp. 165-168, Oct 2008.

[7] H. Gao, J. Y. Tham, K. H. Goh, W. S. Lee, and K. S. Aw, "MP4 File Creator for SVC Adaptive Video Streaming", IEEE Intl Conf Internet Technology and Applications (iTAP), pp. 1-4, Wuhan, China, Apr 2010.

[8] H. Gao, J. Y. Tham, W. S. Lee, and K. H. Goh, "Slice error concealment based on size-adaptive edge-weighted matching and motion vector outlier rejection," Proceedings of the Second APSIPA Annual Summit and Conference (APSIPA ASC 2010), pp. 1058–1063, Biopolis, Singapore, 14-17 December 2010.