Scalable Video Coding and the xStreamer

Alexis Rombaut\textsuperscript{1}, Brecht Vermeulen\textsuperscript{1}, and Piet Demeester\textsuperscript{1}

Department of Information Technology, Ghent University - IBBT - IMEC, G.
Crommenlaan 8 bus 201, B-9050 Ghent, Belgium,
{alexis.rombaut,brecht.vermeulen,pier.demeester}@intec.ugent.be

1 Scalable Video Coding

The problem is scaling video in order to adapt it to the various needs or preferences of end users as well as to the varying terminal capabilities or network conditions. Current solutions use simulcast in which multiple variants of the same content are sent simultaneously. However, such solutions are very bandwidth inefficient. The goal of SVC is twofold: firstly, the bit rate of the single SVC video must be significantly less than the sum of the bit rates in the simulcast scenario and secondly, scaling the encoded video must be computationally inexpensive. SVC in an extension of the H.264/AVC coding standard. SVC divides the video stream into layers with different visual importance. Removing less important layers still retains a quality that is relatively high to the bit rate of the remaining bit stream. The downside of SVC is it requires higher bit rates to obtain the same quality level of a single layer encoding due to the overhead of the additional layers.

Three scaling options are available under SVC:

- Temporal (T): reducing the frame rate
- Spatial (D): reducing the resolution
- Quality (Q): reducing the quality

Fig. 1. SVC encoding, showing in particular the 5 temporal layers (T), 2 spatial layers (D), and 2 quality layers (Q).
Figure 1 shows an overview of an SVC encoding with 5 temporal layers (T), 2 spatial layers (D), and 2 quality layers (Q).

2 xStreamer

The problem is that many network related projects employ video streaming as use case. Many researchers spend considerable time writing his or her own video tool or streamer for each project. The xStreamer aims to avoid this redundant work by creating an easily customizable application for research. The xStreamer can be described as a Modular Multimedia Streamer:

- Modular: combine basic steps
- Multimedia: supports video and audio
- Streamer: multitude of connection types

This modular framework was developed from scratch within the IBCN research group and will be shortly made open source. Figure 2 shows the schematic of a streamer configuration for the xstreamer that sends for example the different SVC layers over three separate connections.

3 Use case: a practical implementation of SVC

The last-mile (cf. figure 3) has limited bandwidth and forms a bottleneck link for video applications. Current solutions fail to offer sufficient Quality of Service (QoS) for the last-mile because they allow this packet loss to occur randomly in a stream while packet loss, depending on which part or layer of the video, has strong impact on the visual quality. The goal is to concentrate the packet loss in the least important layers of a video by combining Scalable Video Coding

![Fig. 2. Differentiated streamer configuration for the xStreamer.](image-url)
Fig. 3. Network overview, showing in particular the last-mile between the access multiplexer and the home gateway, with up to $N$ simultaneous video streams through this link.

(SVC) which offers graceful degradation of quality when removing video layers with Partial Buffer Sharing (PBS) which drops layers proportional to their importance. Yet this combination is not without problems: simulations reveal extensive oscillations in the buffer occupation leading to buffer overflows. Detailed analysis solved these problems and demonstrated it can concentrate the packet loss in the least important video layers.
3rd Gent University
and
Keio University
Global COE Workshop

Proceedings
Monday, 23rd February 2009
At Gent University