Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Multipoint videoconferencing with scalable video coding

ELEFTHERIADIS Alexandros, CIVANLAR M. Reha, SHAPIRO Ofer

(*Layered Media Inc., NJ 07662, USA*)

E-mail: {alex; reha; ofer}@layeredmedia.com

Received Dec. 5, 2005; revision accepted Mar. 1, 2006

Abstract: We describe a system for multipoint videoconferencing that offers extremely low end-to-end delay, low cost and complexity, and high scalability, alongside standard features associated with high-end solutions such as rate matching and personal video layout. The system accommodates heterogeneous receivers and networks based on the Internet Protocol and relies on scalable video coding to provide a coded representation of a source video signal at multiple temporal and spatial resolutions as well as quality levels. These are represented by distinct bitstream components which are created at each end-user encoder. Depending on the specific conferencing environment, some or all of these components are transmitted to a Scalable Video Conferencing Server (SVCS). The SVCS redirects these components to one or more recipients depending on, e.g., the available network conditions and user preferences. The scalable aspect of the video coding technique allows the system to adapt to different network conditions, and also accommodates different end-user requirements (e.g., a user may elect to view another user at a high or low spatial resolution). Performance results concerning flexibility, video quality and delay of the system are presented using the Joint Scalable Video Model (JSVM) of the forthcoming SVC (H.264 Annex G) standard, demonstrating that scalable coding outperforms existing state-of-the-art systems and offers the right platform for building next-generation multipoint videoconferencing systems.

Key words: Multipoint videoconferencing, Scalable video coding, Multipoint control
doi:10.1631/jzus.2006.A0696 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

Modern videoconferencing systems allow two or more participants to communicate with each other in real-time using both audio and video. Conventionally, when more than two participants are present, a star configuration is generally employed, wherein a Multipoint Conferencing Unit (MCU), or bridge, is utilized to connect to all participants and coordinate communications between them. The general architecture is shown in Fig.1.

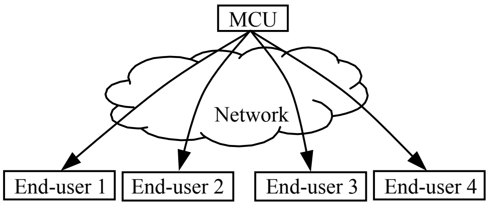

Fig.1 Multipoint videoconferencing system

The MCU's primary tasks are to mix the incoming audio signals so that a single audio stream is transmitted to all participants, and to also mix or composite the video signals into a single stream where portions of a frame show each of the participants.

In conventional systems, MCUs can only offer a single combination of resolutions of individual pictures that have to be the same for all participants. Although users may want to view other participants in different resolutions, for example, with the speaking participant displayed in CIF resolution whereas other attendees (at the time silent) are displayed in QCIF resolution, conventional systems cannot easily provide such functionality. If customization is required for each participant, then a conventional MCU must perform the mixing operation as many times as there are participants of the videoconference. In order to accomplish this task effectively, the MCU must have considerable digital signal processing power, as it must decode multiple audio streams, mix, and re-encode them, and must also decode multiple video streams, composite them into a single frame (with appropriate scaling as needed), and re-encode them again into a single stream. In addition to requiring considerable processing power, these operations also introduce considerable delay. Existing commercial videoconferencing systems utilize special hardware components resulting in highly expensive systems. Furthermore, the quality levels achieved are not commensurate with user expectations. The recent trend towards High Definition (HD) systems (e.g., by LifeSize and other companies), further exacerbate these problems, inherent in the MCU-based architecture.

A critical performance parameter for any videoconferencing system is the end-to-end delay. The requirements for long-distance telephony mandate an end-to-end delay that must be below approximately 200 ms. Higher delays require users to wait before talking, in order to allow video and audio data that may be in-transit to arrive. ITU Recommendation G.114 in fact defines three ranges of one-way network delay for voice applications: 0~150 ms (acceptable for most user applications), 150~400 ms (acceptable provided that administrators and users are aware of its presence and impact), and above 400 ms (unacceptable, except for exceptional cases).

The end-to-end delay can be decomposed into acquisition delay, coding delay, transmission delay, and transport delay. If an MCU is present in the system, then its operation must also be factored into the total end-to-end delay of the system and this may have significant effect on the overall delay.

Another significant component of a videoconferencing system is the network over which it operates. Existing systems utilizing video codecs such as ITU H.261, H.263, and H.264 require a fairly robust communication channel with little or no loss. The required bit rates can range from 64 kbps up to several Mbps. Earlier systems used ISDN lines, whereas newer systems often utilize high-speed Internet connections (xDSL, cable modems, fractional T1, T1 or higher) for high-speed transmission. Although the IP protocol may be used, it is typically implemented in a private or overlay network environment to ensure bandwidth availability. As a result, the operating cost of a high-quality videoconferencing solution must include the substantial cost of implementing and maintaining the required networking infrastructure.

The continuous increase in bandwidth of corporate data networks (e.g., 1 Gbit Ethernet) makes these networks attractive candidates for video transmission eliminating the cost of a dedicated videoconferencing network. At the same time, the end-users personal computer (PC) can be used as the encoding/decoding terminal; indeed, with the addition of a USB-based digital video camera and appropriate software applications to perform the encoding/decoding and network transmission, a so-called “desktop videoconferencing” solution can easily be implemented. In addition, video communication and conferencing capabilities have recently been added to multiple IP communication systems such as IP telephony PBXs, instant messaging, web conferencing, etc. When video communication is added to these systems, both point-to-point and multipoint operation must be supported. The high bandwidth associated with video and the fact that the available network bandwidth can fluctuate widely make it extremely difficult to use these systems in a mission-critical real-time environment.

A further challenge in IP-based networks is their inherent heterogeneity. Users may access videoconferencing services over channels that may have very different bandwidths (e.g., DSL vs Ethernet, or different corporate WAN connections in main and satellite sites). Traditional video codecs such as H.261, H.263 or MPEG-1 and MPEG-2 are designed to provide a single bitstream at a specified bit-rate. Therefore, their end-to-end use on heterogeneous networks is not practical, and requires transcoding at the MCU. Moreover, the designs of these codecs assume that the network can provide a constant bit rate, practically error-free channel between the sender and the receiver.

The H-series codecs, designed specifically for person-to-person communication applications, offer some additional features to increase robustness in the presence of channel errors, but are still only tolerant to a very small percentage of packet losses. Thus, these codec designs are not particularly suitable for best-effort networks. It is possible, however, to transport these bitstreams in their entirety over a channel that offers high Quality of Service (QoS).

Specifically for IP-based networks this is possible using Differentiated Services (DiffServ). This solution, however, is highly expensive if applied to the total video bandwidth as it uses a large percentage of precious network resources.

An additional limitation of traditional, single layer coding is that if a lower spatial resolution is required, the full resolution signal must be received and decoded (thus wasting bandwidth and computational resources), with downscaling performed at the receiver or MCU. Support for multiple resolutions is essential in videoconferencing, as one goal is to fit as many participants as possible into a specific screen area.

Due to these challenges, many end users have found that deployment of video communication is only practical for applications that directly provide tangible cost savings (for example travel replacement application) and not for daily collaborative communication.

Over the years some solutions have been suggested to alleviate these problems. A particularly effective one is the video switching MCU that provides several unprocessed or lightly processed streams to each of the conferencing participants, thus avoiding the delay and cost of hardware problems. However these solution are still challenged by high packet loss sensitivity and difficulties in providing different bit rates or resolutions to different participants.

All of the above mentioned problems can be eliminated using scalable video. For instance, the base layer of scalable video can be transmitted using a high-QoS channel while enhancement layers can be transmitted via a best-effort channel. Doing so, the users are guaranteed to receive video with at least a minimum level of quality (the base layer), and the entire video data need not be carried over the expensive high-QoS connection.

Although scalable coding has been part of standards such as MPEG-2, it has not been used in the marketplace. The increased cost and complexity associated with scalable coding, as well as the lack of widespread use of IP-based communication channels suitable for video have been considerable impediments to widespread adoption of the technology. Furthermore, the value-added to broadcast-oriented video services has not been considered compelling enough to warrant a technology change. We believe that high quality point-to-point and multipoint video conferencing is a natural application for scalable coding, and one where the technology solves pressing, otherwise insurmountable problems providing better quality at a lower cost.

In this paper, we describe a scalable codec solution for multipoint videoconferencing that can offer bandwidth, temporal and spatial resolution, quality, and computational power scalability based on the emerging SVC (Scalable Video Coding) standard developed by JVT (Joint Video Team, a collaborative effort between the ITU VCEG and ISO MPEG groups). This solution can be used with a new MCU architecture, referred to as the Scalable Video Conferencing Server (SVCS), which provides:

(1) Continuous presence (multiple people can be seen simultaneously);

(2) Personal view or layout (each participant chooses his/her own view of the other participants, such as CIF vs QCIF);

(3) Rate matching (each participant may be connected via a network connection with different bandwidth, and needs to receive his/her own data rate from the SVCS);

(4) Efficient use of network's QoS resources;

(5) Minimal additional delay due to the MCU.

The new MCU architecture based on scalable video is presented in the next section. SVC configurations suitable for this application are discussed in Section 3. Experimental performance analysis results verifying the system-wide efficiencies achieved by SVC are provided in Section 4. Concluding remarks are presented in Section 5.

MCU ARCHITECTURE WITH SCALABLE VIDEO

In a scalable multipoint video coding architecture, each participant transmits a scalable bitstream (base plus one or more enhancement streams) to the SVCS. This is shown in Fig.2.

The transmission is performed using a corresponding number of physical or virtual channels. Network management considerations suggest the use of as few channels as possible; with DiffServ a minimum of two channels (or RTP ports) have to be used.

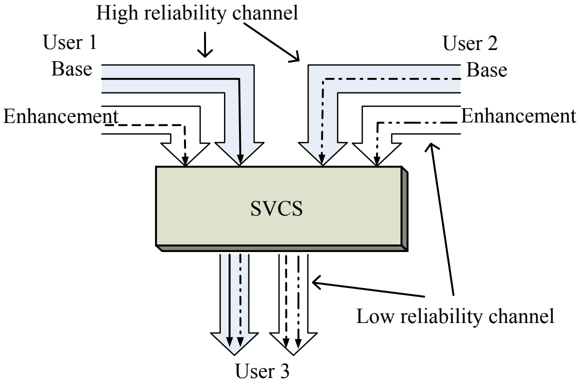

Fig.2 Videoconferencing system using scalable video

The base layer channel is assumed to offer high QoS (High Reliability Channel, HRC), whereas the enhancement stream channel(s) offer lower or no QoS (Low Reliability Channel, LRC). Losses in the enhancement streams will thus result in a graceful degradation of picture quality, with the base layer having the lowest guaranteed quality. It is also possible to transmit some enhancement layer streams through the HRC (assuming bandwidth availability), thus ensuring a higher minimum guaranteed quality.

The SVCS may accordingly select the correct amount and type of information that is required based on the properties and/or settings at the particular location and forward only that information. The selection may be based on, for example, the recipient's bandwidth and desired video resolution(s). No or minimal signal processing is required of the SVCS in this respect; the SVCS may simply read the packet headers of the incoming data and selectively forward the appropriate packets to each of the participants. The various incoming packets are aggregated to two or more channels (for each participant), so that base layer packets are transmitted over the protected channel.

Use of scalable video eliminates the need to decode and encode the video on the SVCS and therefore provides zero algorithmic delay. An additional important benefit is that the video quality is improved, since tandem encoding (repeated encoding/decoding passes) is known to reduce video quality by 0.5~1.5 dB (Chang and Eleftheriadis, 1994). Most significantly, the computational requirements on the SVCS are reduced by approximately two orders of magnitude. This means that a single SVCS can serve a very large number of sessions and/or participants, offering excellent scalability for large-scale deployment.

Fig.3 compares the operation of a traditional MCU and an SVCS. The traditional MCU needs to decode, compose/mix, and re-encode the incoming streams. The SVCS, on the other hand, behaves as application-level router with no or minimal processing of the incoming packets. When Fine Grain Scalability (FGS) is used, the SVCS may in addition implement truncation of the appropriate packets. Truncation, however, is a trivial operation in that only simple rewriting of the packet length value is required.

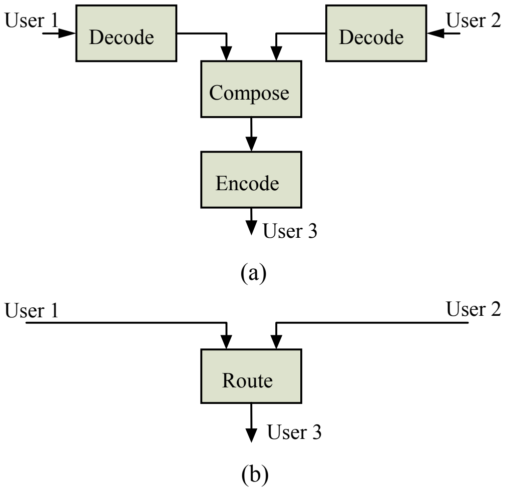

Fig.3 MCU (a) vs SVCS (b) architecture

The end-user terminal is also considerably different for MCU and SVCS architectures. The terminal encodes the local user's video and audio input, and decodes at the same time as many streams of video and audio as the number of participants. Contrary to traditional MCU architectures where compositing and mixing is done at the MCU, here these operations are performed at the end-user terminal itself. The system can be implemented in software (or combination of hardware and software) on a general-purpose PC. This type of system has been reported in the literature based on the use of simulcast before (Civanlar *et al.*, 1997). Current mainstream PC processing speeds already offer enough processing capability for real-time encoding and decoding of several scalable streams. The relative simplicity of the end-user terminal further improves the cost advantage of this architecture, and enables its seamless integration into existing computer communication systems in both corporate as well as home environments.

SCALABLE CODING FOR MULTIPOINT VIDEO CONFERENCING

The SVC standard, as is currently defined (Reichel *et al*., 2005a), offers several tools for scalable coding: spatial/temporal scalability, coarse and fine-grain quality scalability (CGS/FGS), as well as multiple reference frames. The Joint Scalable Video Model (JSVM) is a reference software implementation of an encoder and decoder that is used during the development of the standard (Reichel *et al*., 2005b). In the following we assume the reader some familiarity with basic AVC and SVC terminology.

The scalability features of SVC are built on a pyramidal structure. Temporal scalability is accomplished through a GOP-like structure where a series of B frames are coded between two P frames (e.g., PBBBP). The P frames (currently referred to as "key pictures") together with the very first I frame (IDR frame) form a first temporal scalability layer. The coding of B frames in the JSVM is performed using a hierarchical structure, in which dyadic decomposition is used to construct temporal layers of increased temporal resolution. For example, in the series P1B2B3B4P5, B3 is coded with reference to P1 and P5, whereas B2 is coded through P1 and B3, and B4 through B3 and P5. We see then that B3 can be used to add another temporal layer to that of the P frames (doubling the frame rate), with B2 and B4 creating a third layer that completes the temporal decomposition pyramid. B frame coding induces an additional coding delay, equal to the number of intervening B frames between P frames, assuming instantaneous acquisition/encoding.

Spatial and coarse-grain scalability is based on creating a refinement (in terms of both texture and motion) of the base layer for predicting the enhancement layer at either the increased spatial resolution or increased quality (lower QP). FGS coding is performed by repeated reduction of the quantizer step size and application of an entropy coding process similar to sub-bitplane coding.

Threading and temporal scalability

Use of B frames in a videoconferencing system results in additional delay, and is thus avoided. An alternative mechanism to hierarchical B frames for creating multiple temporal resolutions is the concept of "threads". The threading concept applied in this context was first reported in (Wenger, 1997), and is further developed here for both frame rate control and improved error resilience. We define a thread at a given level as a sequence of pictures that are predicted using pictures either from the same thread, or pictures from a lower level thread. This allows implementation of temporal scalability, since one can eliminate any number of top-level threads without affecting the decoding process of the remaining threads.

Examples of three different threading structures are shown in Fig.4, where each block corresponds to a picture and the arrows indicate the direction, source, and target of prediction. The different structures are labelled as ILP, ILLLP, and IBLBP, as shorthand mnemonics of their organization. For example, for ILLLP, the base layer, L0, is simply a set of P frames spaced four frames apart. The first enhancement layer, L1, has the same frame rate, but prediction is only allowed from the previous L0 frame. The frames of the second enhancement layer, L2, are predicted from the most recent L0 or L1 frame. Thus, while L0 provides one fourth (1:4) of the full temporal resolution, L1 doubles the L0 frame rate (1:2), and L2 doubles the L0+L1 frame rate (1:1, compared with the source frame rate).

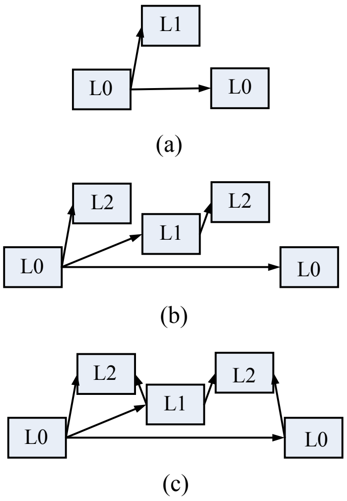

Fig.4 Example threading structures. (a) ILP; (b) ILLLP; (c) IBLBP

Observe that prediction is only performed from pictures of the same or lower threads/layers. The structure is similar to that of hierarchical B frames except that only prediction from the previous frame(s) is allowed. Clearly, additional threading patterns can also be used (e.g., the second L2 frame could use the first L2 frame for prediction purposes). Note that the threading scheme would not be feasible without support for multiple reference frames by the codec.

Spatial/quality scalability

Additional scalability layers can be provided using spatial and quality scalability (SNR). Fig.5 shows an example with spatial scalability (the 'S' blocks are the spatial enhancement layers). In order to decouple the spatial/SNR and temporal scalabilities, all predictions are performed within the same temporal layer and across the same spatial/SNR layer.

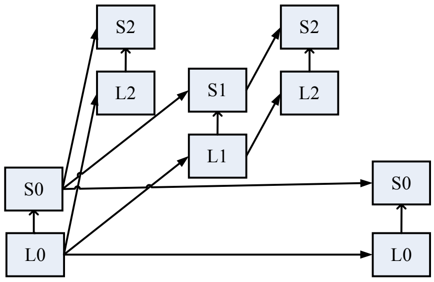

Fig.5 Threading with spatial and/or quality scalability (ILLLP case)

It is also possible to provide spatial scalability using only SNR scalability. Assuming all layers are coded at CIF resolution with quality enhancement layers, one can derive a QCIF resolution picture by decoding only the low quality CIF layer (i.e., without the quality enhancement) at a particular temporal resolution, followed by low-pass filtering and downsampling in each spatial dimension by 2.

Coding of the quality enhancement layer can be accomplished using Coarse Grain Scalability (CGS) or FGS. In the FGS case the Q layer will have the additional flexibility of enabling the use of arbitrary portions of a 'Q' packet (due to the embedded property of the produced bitstream).

PERFORMANCE ANALYSIS USING THE JSVM

The preceding discussion clearly establishes the architectural superiority of SVC for video-conferencing applications. An important consideration, however, are the precise operating points that scalable coding in general, and SVC in particular, enables. Traditionally, comparative analyses in video coding have relied solely on rate-distortion performance (i.e., compression efficiency). With this criterion, a codec that has lower distortion for the same bit rate is considered better. Although issues such as error resilience have been considered as well, in practice compression efficiency has been the dominant factor.

It is clear that scalability introduces a penalty in terms of bit rate compared to a single-layer codec. What is more important, however, is to consider the "system-wide" implications of scalable vs single-layer coding. Indeed, we find that the penalty in bit rate is offset by substantial performance improvement and complexity reduction in other aspects of the system.

In order to illustrate the benefits of scalable video coding for videoconferencing, we have created a "quality-delay" graph. Contrary to rate-distortion diagrams, this graph plots coding quality for a particular encoder configuration vs the end-to-end delay that a particular codec would be subject to in a multipoint videoconferencing scenario. The end-to-end delay includes all sources of delay, and not just the one introduced by the codec alone. By plotting data points on a fixed, constant bandwidth basis, we obtain a clear pictorial indication of the performance of different system configurations.

In order to show results that can be easily reproduced on a very well-known architecture, we used the JSVM 4 software rather than our own software. We should point out that the JSVM code is not optimized for low delay operation and, as a result, the performance differences with respect to coding efficiency are magnified. Even with these limitations, as we will see there is compelling evidence of the superiority of scalable coding over single layer coding.

Coding

For our experiments, we used the "Foreman" test sequence (first 160 frames, CIF, 30 fps). This initial segment of the sequence is similar to the content likely to be found in a videoconferencing setting (and more challenging to encode due to camera motion). We always ran the JSVM with a motion search range of 1 (using fast search), and in single-loop mode. We selected a very small motion search range to reflect the performance of low-complexity end-points.

We first examined single-layer configurations for coding at CIF resolution, 30 fps, and a target bit rate of 400 kbps. We specifically examined IPP, IBP, and IBBBP (hierarchical B) configurations. Although the use of B frames is not desirable in videoconferencing, we included the data points for reference purposes as they are indicative of the maximum possible coding efficiency if delay is ignored. As the open-loop rate control encoder cannot match exactly the desired rate, the results we report are obtained through the use of linear interpolation (on the R-D plane) between *QP* values that engulf the desired output bit rate.

In terms of temporal scalability, we examined threaded structures with two and three temporal layers (with *QP* scaling). The first two configurations are the ILP and ILLLP as depicted in Fig.4. We also include results from the configuration IBLBP; the only difference with ILLLP is that the L2 frames are actually B pictures that are predicted from the two closest L1 or L0 frames. Note that this structure has an additional 1 frame delay.

For spatial scalability, we report results for QCIF and CIF at 30 fps. The spatial enhancement is built on top of the layering structure used to effect temporal scalability (Fig.5). We use the same *QP* in both base and enhancement layers. The results are summarized in Tables 1~3; detailed R-D plots are shown in Fig.6.

Comparing Tables 1 and 2, we observe that threading has an impact of only 0.04 dB for the IPP vs ILP case, and 0.3 dB for ILLLP (all have zero coding delay). Comparing IBP and IBLBP (one frame delay), we observe a drop of 0.24 dB. The IBLBP solution performs at 0.75 dB higher than ILLLP. Table 3 shows the results for the spatially scalable coding case.

Table 1 Single layer coding at 400 kbps

Codec	*Y-PSNR* (dB)
IPP	36.06
IBP	36.75
IBBBP	37.08

Table 2 Threaded coding at 400 kbps

Codec	*Y-PSNR* (dB)
ILP	36.02
ILLLP	35.76
IBLBP	36.51

Table 3 Scalable (CIF/QCIF) coding at 400 kbps

Codec	*Y-PSNR* (dB)
IPP	35.29
ILP	35.39
ILLLP	35.42
IBLBP	36.05

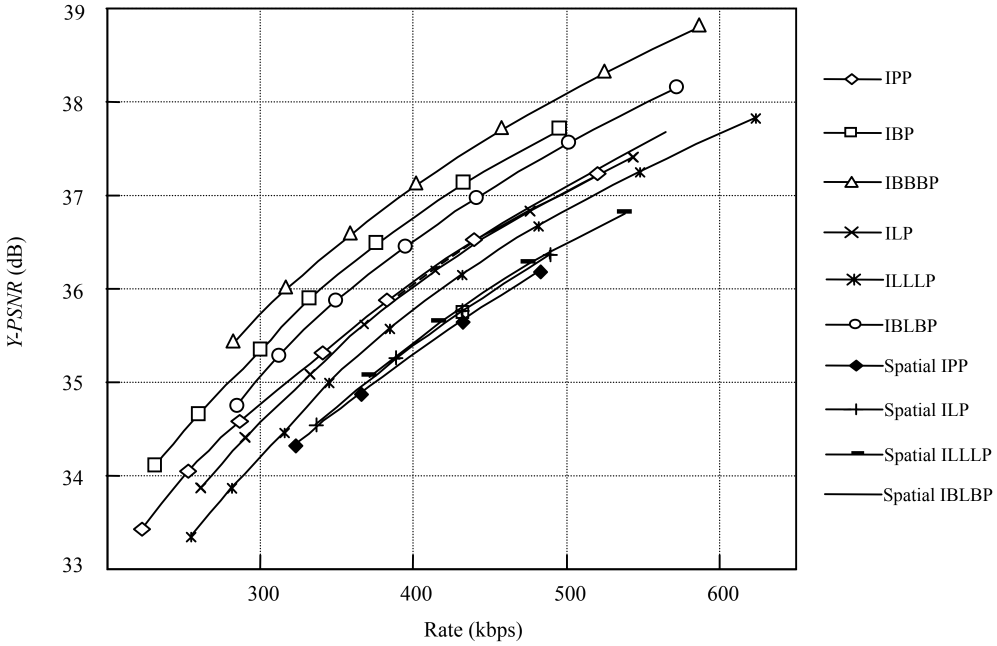

Fig.6 Rate-distortion curves

The quality differential for ILLLP is 0.34 dB compared with the non-spatially scalable case, and in all cases it ranges from 0.3~0.8 dB. We should point out that these numbers do not represent the absolute limits of SVC, and are sample points based on a non-optimized encoder. Still, these (expected) quality drops are more than made up for when other system factors are taken into account.

System modelling

We now examine the delay behavior of the different systems. We use a model that involves a set of parameters that describe different aspects of the system. First, note that the coding delay for each codec structure is proportional to the number of B pictures used. For example, the IBLBP structure has a one-frame delay (33.3 ms). We assume instantaneous acquisition and processing delays, as they are implementation-dependent (they can, however, easily be integrated into our model). We assume an MCU processing delay of 120 ms (decode, compose, and re-encode). The SVCS delay is assumed to be 5 ms.

If a transcoding MCU is used (i.e., not a switching MCU), then an additional quality drop is expected. Although the transcoding error has been characterized in the past (Chang and Eleftheriadis, 1994), here we experimentally determined the actual transcoding error. For this, we first coded the original sequence in CIF at about 400 kbps (380 kbps, *QP*=30), then we decoded the output and downsampled it to QCIF, after which we coded it to 100 kbps. The decoded output is the result of transcoding. The choice of the factor 1/4 for the bitrate is based on the assumption that the participant will be one of the four contained in a 2×2 QCIF matrix, composed as four independent slices at the MCU. The different operations are shown in Fig.7. Quality differences (*Y-PSNR*) are shown next to each stage. The transcoding loss is the difference between *B*-*C* and *B*-*D*, i.e., 0.7 dB.

Finally, we also take into account packet loss, assuming a 5% packet loss rate. To estimate the impact of packet losses for single-layer coding, we use the experiments conducted by Bandyopadhyay *et al.*(2005), where motion vector-based concealment is employed to recover lost frames in H.264 AVC coding. The motion vectors of the previous frame are used as estimates of the motion vector field of the current frame, and are then applied on the previous frame to produce an estimate of the lost frame. At CIF resolution, 30 fps, IPPI encoding with an I period of 1 s, and a 5% loss rate the quality drop is between 2.2 and 6 dB, depending on the sequence and *QP* value. The difference narrows as the *QP* is raised. Here we will assume a 3 dB quality drop. Note that periodic I frames eliminate drift but are impractical for videoconferencing. Furthermore, alternative error resilience techniques (e.g., multiple slices, FMO) will decrease coding efficiency—and hence quality—in a similar way. An additional 1~2.5 dB loss is reported in (Bandyopadhyay *et al.*, 2005) when the packet loss rate increases to 10%.

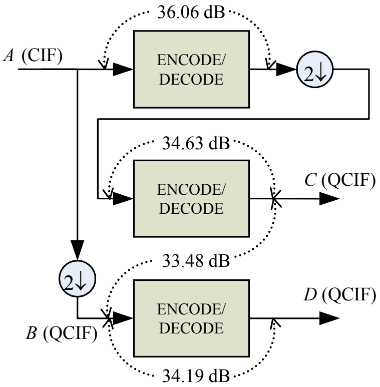

Fig.7 Determination of transcoding penalty

For spatially scalable coding, in order to estimate the quality degradation we performed actual experiments on our own SVCS system. Specifically, we assumed DiffServ operation with 5% and 10% packet loss rates. While the rate is based on all packets, the errors affect only the LRC. We further assume that L0~L2 and S0 packets are both carried on the HRC, with S1 and S2 being transmitted on the LRC (108 kbps). Error concealment was performed in a proprietary way using information from the base layer. Note that error propagation terminates at the next L0/S0 frame. As expected the quality drop is very small, less than 0.4 dB at 10% loss. Clearly, threading, spatial scalability, and DiffServ taken together constitute an extremely powerful combination for mitigating packet losses. The model parameters are summarized in Table 4.

Distortion-delay analysis

We can now examine the different system behaviors by positioning *Y-PSNR* values on a distortion-delay chart. Fig.8 shows the various configurations.

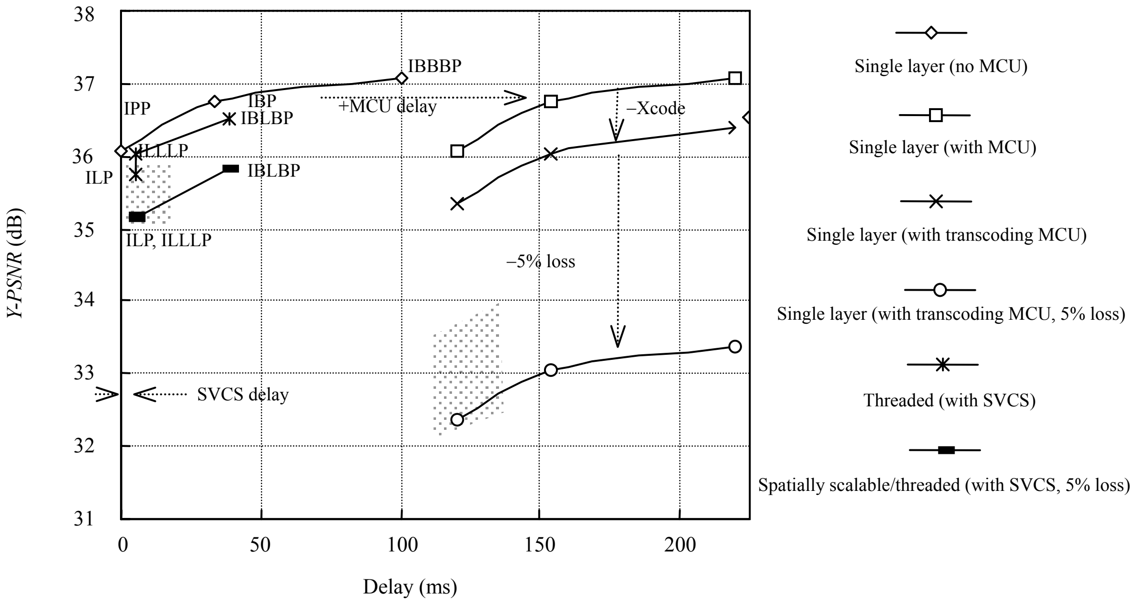

Fig.8 Distortion-delay analysis

Table 4 Model parameters

Parameter	Value
MCU delay	120 ms
SVCS delay	5 ms
Transcoding penalty	0.7 dB
Δ*PSNR* at 5% loss (single layer)	3 dB
Δ*PSNR* at 5% loss (scalable, DiffServ)	0.23 dB

First, to facilitate direct comparison of compression efficiency, we provide plot lines for the different codecs assuming zero system delay. The top-right line represents single-layer coding, with IPP, IBP, and IBBBP structures (from left to right). We observe that as the number of B pictures increases, we obtain the expected increase in both quality and delay. If we take into account the MCU delay, the line is transposed to the right (as indicated by the +MCU arrow. Further translations, this time downwards, are introduced by taking into account transcoding (Xcode), and the packet loss related distortion (+5% loss).

The next line below the top-left depicts the performance of threaded coding, with the sample points corresponding to ILLLP, ILP, and IBLBP respectively (left to right). Observe that the *PSNR* of IBLBP is 0.24 dB less than IBP, for the same coding delay, and 0.8 dB better than ILLLP. Since the SVCS delay is very small, the horizontal translation of the curve is very small. Finally, we also show the results for spatially scalable coding with threading in the lower-left line. Since the vertical translation (*PSNR* drop due to packet loss) is small, we only show the curve for the 5% loss rate (with SVCS delay). The data points correspond to ILP and ILLLP, and IBLBP.

The operating points of an SVCS-based system with 5% loss and that of an MCU-based system with 5% loss and transcoding are shown by the grayed rectangles. As we can see, the SVCS architecture offers a tremendous reduction in system delay and an improvement in *PSNR* of 2.8~3.5 dB, depending on the codec configuration. The delay reduction makes it possible to actually add back some coding delay by including a B picture (something that was considered untenable in traditional videoconferencing), while still being far superior to a traditional MCU. The resulting *PSNR* improvement equates spatially scalable IBLBP with spatially scalable IPP, thus eliminating the threading overhead. Even without B pictures, the quality is 2.8 dB better than single-layer coding while the delay is an order of magnitude less.

CONCLUDING REMARKS

We presented a system-level analysis of scalable-video based multipoint videoconferencing systems, based on the forthcoming SVC standard. Using the

JSVM reference software we demonstrated the superiority of scalable coding for videoconferencing in a realistic IP-based network. By positioning different system configurations on a distortion-delay plot, we showed that scalable video can operate at 2.8~3.5 dB better quality than traditional single-layer coding with 3%~30% of the end-to-end delay, and offer all functionalities associated with high-end systems (rate matching, personalized layout, etc).

Commercial single-layer multipoint videoconferencing systems employ a number of error resilient features (including proprietary solutions) and may have substantial encoder optimization features. For these systems the *PSNR* results will be increased by a certain amount due to increased encoder efficiency, and decreased through the addition of the error resilience features. Furthermore, the behavior of the coded signal in the presence of errors may also differ from the case described here. Our single-layer comparison anchor point (which includes intra refresh every 1 s) is actually underestimating the quality drop and we thus expect actual single-layer operating points similar or lower than those shown here. Our subjective evaluation through Layered Media's SVC-based videoconferencing system further corroborates these quantitative results. This work offers a framework on which different systems can be positioned (in the distortion-delay plane) by their designers for comparative, quantitative evaluation.

The use of scalable video in multipoint videoconferencing applications appears to have unique benefits that may not have equivalents in other video applications (e.g., broadcasting). For example, rate matching for broadcast-type applications may require addition of a media gateway to filter out the enhancement layer; compared with simulcasting, this represents a cost increase for the provider. For videoconferencing, an existing server (the MCU) is greatly simplified and, hence, the net cost is reduced.

Encoder optimization for SVC for either efficiency or resilience is still in its infancy, and we expect significant *PSNR* gains in the near future compared with the JSVM, further solidifying SVC's advantages for videoconferencing systems.

ACKNOWLEDGEMENT

The authors would like to thank Dr. Danny Hong for generating the scalable coding error resilience results, and Dr. Thomas Wiegand for useful discussions regarding SVC.

References

Bandyopadhyay, S., Wu, Z., Pandit, P., Boyce, J., 2005. Frame Loss Error Concealment for H.264/AVC. Doc. JVT-P072. Poznan, Poland.

Chang, S.F., Eleftheriadis, A., 1994. Error Accumulation of Repetitive Image Coding. Proc. IEEE Int'l Symp. on Circuits and Systems. London, England, **3**:201-204.

Civanlar, M.R., Gaglianello, R.D., Cash, G.L., 1997. Efficient multi-resolution, multi-stream video systems using standard codecs. *Journal of VLSI Signal Processing*, **17**(2-3): 269-279.

Reichel, J., Schwartz, H., Wien, M., 2005a. Draft of Scalable Video Coding—Working Draft 4. Joint Video Team, Doc. JVT-Q201. Nice, France.

Reichel, J., Schwartz, H., Wien, M., 2005b. Joint Scalable Video Model JSVM-4. Joint Video Team, Doc. JVT-Q202. Nice, France.

Wenger, S., 1997. Video Redundancy Coding in H.263+. Workshop on Audio-Visual Services for Packet Networks.