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

Multiple description coding with spatial-temporal hybrid interpolation for video streaming in peer-to-peer networks

LU Meng-ting, LIN Chang-kuan, YAO Jason, CHEN Homer H.†

(*Graduate Institute of Communication Engineering, National Taiwan University, Taipei*)

†E-mail: homer@cc.ee.ntu.edu.tw

Received Dec. 2, 2005; revision accepted Feb. 10, 2006

Abstract: In this paper, we present an innovative design of multiple description coding with spatial-temporal hybrid interpolation (MDC-STHI) for peer-to-peer (P2P) video streaming. MDC can be effective in P2P networks because the nature of overlay routing makes path diversity more feasible. However, most MDC schemes require a redesign of video coding systems and are not cost-effective for wide deployment. We base our work on multiple state video coding, a form of MDC that can utilize standard codecs. Two quarter-sized video bit streams are generated as redundancies and embedded in the original-sized streams. With MDC-STHI, the nodes in P2P network can adjust the streaming traffic to satisfy the constraints of their devices and network environment. By design, the redundancies are used to compensate for missing frames, and can also be streamed independently to fulfill certain needs of low rate, low resolution applications. For better error concealment, optimal weights for spatial and temporal interpolation are determined at the source, quantized, and included in redundancies.

Key words: P2P streaming, Multiple state video coding, Multiple description coding (MDC)
doi:10.1631/jzus.2006.A0894 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

Multimedia services over the Internet are becoming popular due to the widespread deployment of broadband access. However, the conventional client-server architecture severely limits the number of simultaneous users, especially for bandwidth intensive applications such as video streaming. P2P networks, on the other hand, offer a solution to the scalability problem. As a node joins a P2P network, it not only consumes resources but also contributes its bandwidth or computation power. By relaying data over P2P networks, users receiving data also help its distribution. This mechanism greatly relieves the load for streaming servers and improves error resilience. In addition, the nature of overlay routing in P2P networks makes path diversity possible, which makes MDC more feasible for P2P video streaming.

Many P2P streaming systems have been proposed, some of them based on successful P2P file sharing systems. For example, GnuStream (Jiang *et al*., 2003) is built on top of Gnutella. A few systems focus on the construction of an application-level multicast tree such as ZigZag (Tran *et al*., 2004). One of the most popular P2P streaming systems is CoolStreaming/DONet (Zhang *et al*., 2005). The subscribers of CoolStreaming form peer groups according to the channels they choose and exchange video packets among peers. Unlike the client-server paradigm that suffers from degrading performance when more users subscribe to the same program, the video quality of CoolStreaming can actually improve with more subscribers because peers gain more choices for data source. This system has been available to the public and extensively evaluated over the Internet, involving more than 10000 ordinary Internet users.

Among so many P2P streaming systems, only a small number of them apply MDC. Padmanabhan *et al*.(2003) introduced data redundancy to CoopNet using MDC. Zink and Mauthe (2004) discussed the

feasibility of MDC for P2P networks and compared it to hierarchically layered encoded video. Khan *et al*.(2004) compared the performance of MDC over P2P networks to that of Content Delivery Networks (CDN). In the above three papers, MDC proved to be effective over P2P networks. However, conventional MDC schemes are too complex and use non-standard video codecs, so they are seldom used in practice. Multiple state video coding, proposed by Apostolopoulos (1999; 2001), has a structure of MDC, is easy to implement, and may still use standard codecs as basis. Zhang and Stevenson (2004) also discussed the error concealment method when parts of the descriptions are lost.

In this paper, we propose a novel design of MDC with spatial-temporal hybrid interpolation (MDC-STHI) that is particularly suitable for P2P video streaming. Based on the multiple state video coding, two streams of lower-resolution pictures are added to enhance the scalability and error resilience. In order to improve the video quality when a description stream is missing, we also introduce a spatial-temporal hybrid interpolation scheme that exploits both spatial and temporal information at the source. This feature allows the decoder to conceal the lost frames more effectively and provide better video quality.

The rest of this paper is organized as follows. Section 2 presents the system architecture of our MDC-STHI for P2P networks. Section 3 describes the proposed spatial-temporal hybrid interpolation scheme. Simulation results and discussion are presented in Section 4. Finally, Section 5 concludes this paper.

MULTIPLE DESCRIPTION CODING WITH SPATIAL-TEMPORAL HYBRID INTERPOLATION

Components of MDC-STHI

Fig.1 shows the conventional MDC scheme. The original video signal is coded into multiple bitstreams such that any one bitstream is sufficient to produce a baseline signal and additional bitstreams can improve the quality. Most designs of MDC are not compatible with standard codecs such as MPEG4, making them unsuitable for quick and cost-effective deployment. On the other hand, multiple state video coding (Apostolopoulos, 1999; 2001), as a form of MDC, can utilize standard codecs for its descriptions. Fig.2 shows the concept of multiple state video coding, where the original video sequence is separated into even and odd streams, each encoded and transmitted independently. Basically each description can reproduce the video at a half frame rate. When both descriptions are received, a full rate video can be played. This method, however, would lower the coding efficiency because the temporal relationship is not fully exploited. As shown in Fig.3, multiple state

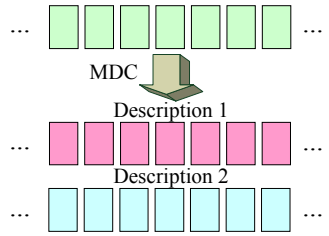

Fig.1 Traditional MDC scheme

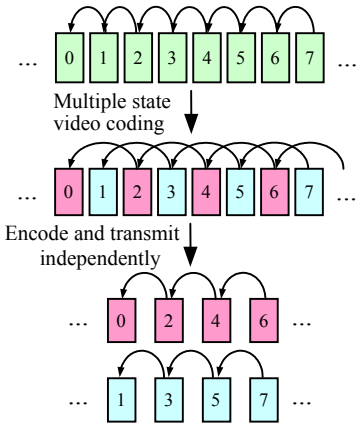

Fig.2 Multiple state video coding

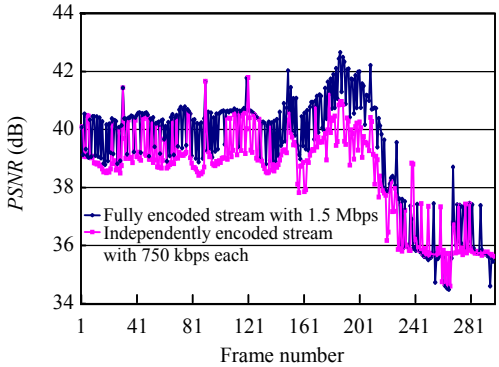

Fig.3 Comparison between normal encoded streams and multiple state video streams

video coding, using MPEG4 for both descriptions, loses about 1 dB in PSNR in comparison to normal MPEG4 video at the same total bit rate.

Based on the concept of multiple state video coding, MDC-STHI consists of four streams as shown in Fig.4. Stream E_f and O_f represent the even and odd streams encoded from the original video. E_q and O_q are similar streams encoded at a low bit rate from the down-sampled version of the original sequence. Our implementation uses 2:1 down sampling in each dimension, resulting in one quarter of the original resolution. While E_q and O_q require extra bits to encode, we will show in the next section that the combination of the four streams has many desirable features for video streaming in P2P networks.

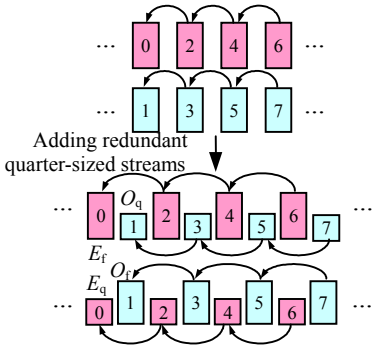

Fig.4 MDC-STHI

MDC-STHI for P2P streaming applications

Fig.5 illustrates a possible P2P networking situation using MDC-STHI. Generally, Internet users with various computational capabilities are connected through networks of different bandwidths, and in a P2P environment like CoolStreaming, they may be randomly grouped as peers for video streaming. In Fig.5, the number of circles indicates roughly the amount of resources at the node. Thus, N_3 has enough bandwidth and computing power to retrieve both streams E_f+O_q and O_f+E_q from N_1 and N_2. In turn, N_4, N_5, and N_6 can receive appropriate combinations of streams from N_3 according to their available resources. For example, N_4 may retrieve only E_q or O_q from N_3 because it uses a mobile device with a narrowband wireless connection and a small-sized screen that best displays quarter-sized videos. On the other hand, N_6 can accommodate all streams from N_3 and may further distribute the video streams intelligently to other peers. Compared to other MDC designs, MDC-STHI offers superior flexibility for P2P video streaming as the four streams are independently encoded and a node may adaptively deliver suitable combinations according to the peers' capabilities.

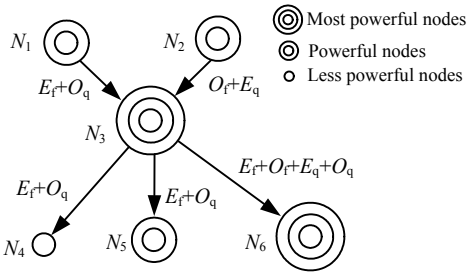

Fig.5 MDC-STHI over a P2P network

The combination of E_f+O_q or O_f+E_q poses an interesting challenge as it contains two streams of different resolutions. During transmission, the bits from O_q may piggyback in the same IP packet with E_f of the previous frame, and vice versa for O_q. This saves the overhead and ensures same time arrival of adjacent frames. In the next section, we will elaborate on an innovative hybrid interpolation scheme for such combinations.

HYBRID INTERPOLATION SCHEME

This section first describes the overall structure of our codec and hybrid interpolation scheme. The spatial interpolation is then discussed, followed by the temporal interpolation, and the two methods are combined for the best result.

The decoder and encoder structures of our design are shown in Fig.6 and Fig.7 respectively. We only explain the description stream containing E_f and O_q as the other combination works the same way.

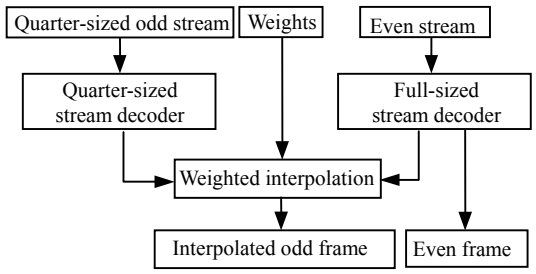

Fig.6 Decoder block diagram

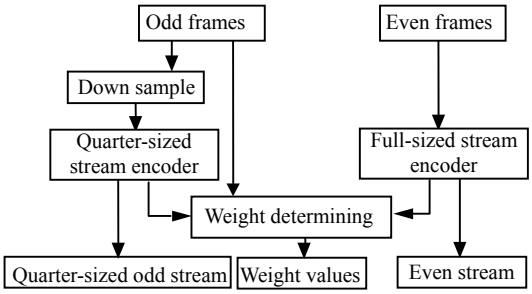

Fig.7 Encoder block diagram

In the decoder, we can get the best video quality if both descriptions are received and there is no need to construct interpolated frames. However, if O_f is lost, we need to compensate for it with O_q. The hybrid interpolated frame can be computed from Eq.(1),

$$X'_{i,j} = w^{\text{Spatial}}_{bi,bj} \cdot X^{\text{Spatial}}_{i,j} + w^{\text{Temporal}}_{bi,bj} \cdot X^{\text{Temporal}}_{i,j}, \quad (w^{\text{Spatial}}_{bi,bj} + w^{\text{Temporal}}_{bi,bj} = 1), \tag{1}$$

where $X^{\text{Spatial}}_{i,j}$ is pixel value at position (i, j) of the spatially interpolated frame; $X^{\text{Temporal}}_{i,j}$ is pixel value at position (i, j) of the temporally interpolated frame; $w^{\text{Spatial}}_{bi,bj}$ is weight value for spatial interpolation of the block at position (bi, bj); $w^{\text{Temporal}}_{bi,bj}$ is weight value for temporal interpolation of the block at position (bi, bj); $X'_{i,j}$ is pixel value at position (i, j) of the weighted interpolated frame; $X_{i,j}$ is pixel value at position (i, j) of the original odd frame.

The corresponding block diagram is shown in Fig.8. The decoded E_f and O_q are used to produce the temporal and spatial interpolated frame, and the missing frame is replaced by a linear combination of the two interpolated frames with the weights determined at the encoder.

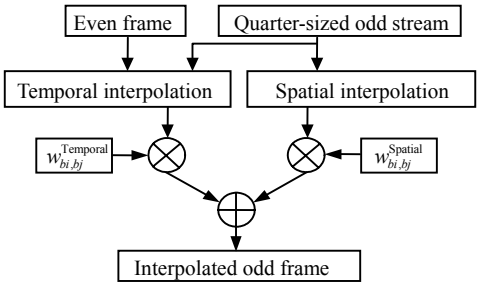

Fig.8 Hybrid interpolation block diagram

In the encoder, even frames are encoded normally while odd frames are down-sampled to quarter size before encoding. Then, they are used to produce the temporal and spatial interpolated frames. In order to determine the optimal weight for each block, we compute the squared error for each possible weighted sum of the two interpolated frames when compared with the original frame. The weights producing the highest *PSNR* are chosen as the optimal weights, which are then quantized and transmitted along with O_q.

Spatial interpolation

We use a simple spatial interpolation method in our system. As shown in Fig.9, the black circles represent the original pixels of quarter-sized frames, and the white circles represent the pixels we want to interpolate. The interpolated pixel values can be calculated from the following equations:

$$\begin{aligned}
Pixel_1 &= (Pixel_a + Pixel_c)/2,\\
Pixel_2 &= (Pixel_a + Pixel_b)/2,\\
Pixel_3 &= (Pixel_a + Pixel_b + Pixel_c + Pixel_d)/4,\\
Pixel_4 &= (Pixel_c + Pixel_d)/2,\\
Pixel_5 &= (Pixel_b + Pixel_d)/2.
\end{aligned}$$

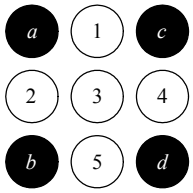

Fig.9 Bilinear spatial interpolation

Temporal interpolation

We use motion-compensated interpolation as our temporal interpolation. First, the even frames are down-sampled to quarter size. Then, by using the down-sampled even frame as the reference frame, we can find the motion vectors for each 8×8 block in the quarter-sized odd frame. Finally, by pasting the corresponding 16×16 blocks of the even frame pointed to by motion vectors, we can get the temporally interpolated frame.

Weight determination

Because the sum of temporal weight value and

spatial weight value is 1, we only need to determine one of them. The best weight value can be calculated based on the following Eq.(2):

$$W_{bi,bj}^{\text{Temporal}} = \arg\min_{w} \sum_{i'=1}^{16}\sum_{j'=1}^{16} \{X_{16bi+i',16bj+j'} - [X_{16bi+i',16bj+j'}^{\text{Temporal}} w + X_{16bi+i',16bj+j'}^{\text{Spatial}} \times (1-w)]\}^2. \quad (2)$$

In Eq.(2), we get the best weight value by comparing the concealed block and the original block. During searching for the minimal sum of squared difference (SSD), we can find the best weight value. The precision of the quantized weight value is another important factor. We tested different weight precisions and the results are shown in Fig.10 showing that the performance is almost the same for precision smaller than 1/3. Therefore, we set our precision to be 1/3 and it can be represented with 2 bits only.

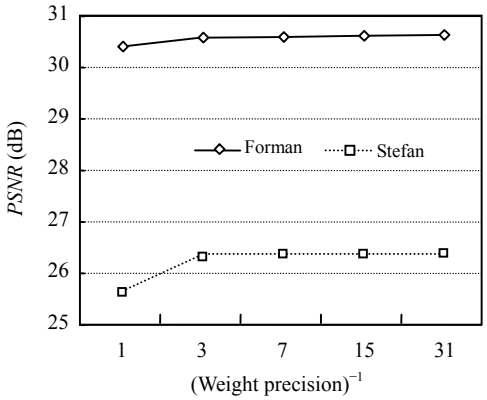

Fig.10 Average *PSNR* values for using different weight precisions

SIMULATION RESULTS

In our experiments, we use Foreman sequence with CIF resolution. All test sequences are encoded with only I and P frames with I frame interval being 30 frames. The bit rate for each full-sized stream is 750 kbps, and that for each quarter-sized stream is 150 kbps. We set the precision of the weight value to 1/3, so the quantized weight value can be represented with 2 bits if no compression is applied. The transmission of weight information therefore requires only about 12 kbps or 1.3% of the total bandwidth. We simulate the scenario when one description stream is lost or deliberately omitted due to bandwidth limitation, and only the full-sized even frames and quarter-sized odd frames are available, meaning that E_f and O_q are interpolated to produce a full-sized video sequence. The *PSNR* values of interpolated odd frames are shown in Fig.11. From this figure, we can conclude that the weight information does improve the overall video quality. We also show the framesfrom different interpolation methods in Fig.12, with the perceptual quality being also better with the proposed hybrid interpolation scheme. Fig.13 shows the distribution of the weight values. We get this picture by setting the pixel values according to Eq.(3):

$$X_{i,j} = W_{bi,bj}^{\text{Temporal}} \times \frac{256}{(\text{Weight resolution})^{-1}+1}. \quad (3)$$

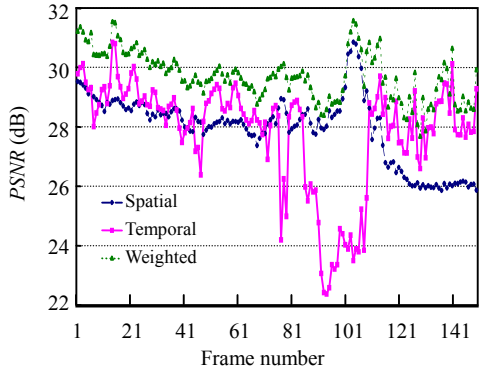

Fig.11 *PSNR* values for different interpolation schemes

CONCLUSION AND FUTURE WORK

In this paper, we present a novel video coding scheme named MDC-STHI. The proposed system can be particularly useful in P2P video streaming, and the concealment mechanism also proves to be effective.

Currently we are incorporating MDC-STHI in an experimental P2P network in which peer nodes can exploit the flexibility of our design to achieve better performance. For better coding efficiency, the overhead for transmitting quantized weight values can be compressed based on the temporal and spatial relationships. The complexity for finding the best weight can also be reduced by taking the motion vector from adjacent or previous blocks.

(a) (b)

(c) (d)

Fig.12 The 75th frame of (a) the original odd stream; (b) the weighted interpolated odd stream; (c) the spatially interpolated odd stream; and (d) the temporally interpolated odd stream

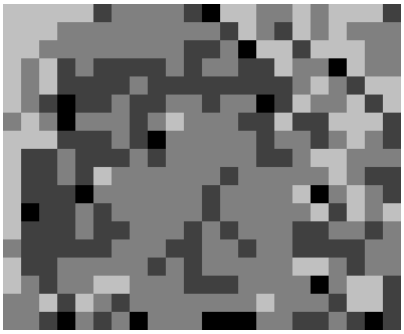

Fig.13 Weight value distribution of the 75th frame of the weighted interpolated odd stream

ACKNOWLEDGEMENT

This work was supported in part by grants from Intel and the National Science Council of Taiwan under contracts NSC 94-2219-E-002-016, NSC 94-2219-E-002-012 and NSC 94-2725-E-002-006-PAE.

References

Apostolopoulos, J.G., 1999. Error-Resilient Video Compression via Multiple State Streams. Proc. International Workshop on Very Low Bitrate Video Coding (VLBV'99), p.168-171.

Apostolopoulos, J.G., 2001. Reliable Video Communication over Lossy Packet Networks Using Multiple State Encoding and Path Diversity. Proc. of Visual Communications and Image Processing (VCIP) 2001, **4310**: 392-409.

Jiang, X., Dong, Y., Xu, D., Bhargava, B., 2003. GnuStream: a P2P Media Streaming System Prototype. Proc. International Conference on Multimedia and Expo (ICME) 2003.

Khan, S., Schollmeier, R., Seinbach, E., 2004. A Performance Comparison of Multiple Description Video Streaming in Peer-to-Peer and Content Delivery Networks. Proc. International Conference on Multimedia and Expo (ICME).

Padmanabhan, V.N., Wang, H.J., Chou, P.A., 2003. Resilient Peer-to-Peer Streaming. Proc. IEEE International Conference on Network Protocols, p.240-247.

Tran, D.A., Hua, K.A., Do, T.T., 2004. A peer-to-peer architecture for media streaming. *IEEE Journal on Selected Areas in Communications*, **22**(1):121-133. [doi:10.1109/JSAC.2003.818803]

Zhang, G., Stevenson, R.L., 2004. Efficient Error Recovery for Multiple Description Video Coding. Proc. of International Conference on Image Processing (ICIP) 2004, p.829-832.

Zhang, X., Liu, J., Li, B., Yum, T.S.P., 2005. DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming. IEEE INFOCOM'05. Miami, FL, USA.

Zink, M., Mauthe, A., 2004. P2P Streaming using Multiple Description Coded Video. Proc. Euromicro Conference, p.240-247.