IEEE International Conference on Image Processing, Kobe, Japan, October 1999.

FIELD-TO-FRAME TRANSCODING WITH SPATIAL AND TEMPORAL DOWNSAMPLING

Susie J. Wee, John G. Apostolopoulos

Hewlett-Packard Laboratories
Palo Alto, CA USA

Nick Feamster

Massachusetts Institute of Technology
Cambridge, MA USA

ABSTRACT

We present an algorithm for transcoding high-rate compressed bitstreams containing field-coded interlaced video to lower-rate compressed bitstreams containing frame-coded progressive video. We focus on MPEG-2 to H.263 transcoding, however these results can be extended to other lower-rate video compression standards including MPEG-4 simple profile and MPEG-1. A conventional approach to the transcoding problem involves decoding the input bitstream, spatially and temporally downsampling the decoded frames, and re-encoding the result. The proposed transcoder achieves improved performance by exploiting the details of the MPEG-2 and H.263 compression standards when performing interlaced to progressive (or field to frame) conversion with spatial downsampling and frame-rate reduction. The transcoder reduces the MPEG-2 decoding requirements by temporally downsampling the data at the bitstream level and reduces the H.263 encoding requirements by largely bypassing H.263 motion estimation by reusing the motion vectors and coding modes given in the input bitstream. In software implementations, the proposed approach achieved a 5x speedup over the conventional approach with only a 0.3 and 0.5 dB loss in PSNR for the Carousel and Bus sequences.

1. INTRODUCTION

Video communication requires the seamless delivery of video content to a broad range of users with different bandwidth and resource constraints. The video attributes and compression standards used by the source signal, the communication channel, and the client device however are far from seamless. Thus, efficient transcoding algorithms must be designed to fix these mismatches and provide users with a seamless experience. A likely scenario involves the MPEG-2 and H.263 standards. MPEG-2 was designed for high-quality, high-rate applications and is used in digital television and DVD. H.263 was designed for low-rate communication over ISDN and analog telephone lines. An MPEG-2 to H.263 transcoder enables the transmission of MPEG program material over lower-rate communication channels such as ISDN lines, analog telephone lines, the internet, and wireless links.

In this paper, we propose a transcoding algorithm that converts a high-rate interlaced MPEG-2 bitstream to a low-rate progressive H.263 bitstream. Bitrate reduction is achieved with spatial and temporal downsampling and requantization, and the details of the MPEG-2 and H.263 compression standards are exploited when performing interlaced to progressive (or field to frame) conversion. Previous transcoding work has examined tradeoffs in picture quality and computational complexity when transcoding for bitrate and/or resolution reduction within the MPEG standard or within the H.261/3 standard [1, 2, 3]. To the authors knowledge, the only work reported on transcoding between the two standards was in [4], where an MPEG to H.263 transcoder was developed for progressive input and output bitstreams when retaining the full temporal frame rate of the video. Our work differs in that our algorithm was designed with the goal of creating a simple transcoder that supports field-coded interlaced video bitstreams and achieves bitrate reduction by allowing temporal frame rate reductions in addition to spatial resolution reduction and requantization.

2. PROBLEM DESCRIPTION

We focus on the problem of transcoding a given MPEG-2 bitstream to a lower-rate H.263 bitstream. The problem can be described by considering the conventional approach shown in Figure 1. An MPEG bitstream is first decoded into its decompressed video frames. These high-resolution video frames are then downsampled to form a video sequence with a lower spatial resolution and frame rate. This sequence is then re-encoded into a lower-rate H.263 bitstream.

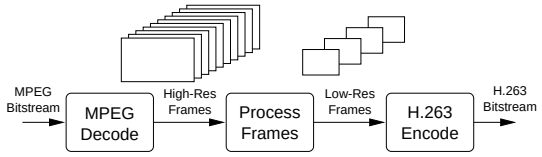

Figure 1: *Conventional approach.*

This conventional approach to transcoding is inefficient in its use of computational and memory resources. The goal of this work is to design computation- and memory-efficient algorithms that achieve MPEG-2 to H.263 transcoding with minimal loss in picture quality.

A number of issues arise when designing an MPEG-2 to H.263 transcoding algorithm. While both standards are based on block motion compensation and the block DCT, there are many differences that must be addressed. A few of these differences are listed below:

- Interlaced vs. progressive video format: MPEG-2 allows interlaced video formats for applications including digital television and DVD. H.263 only supports progressive formats.
- Number of I frames: MPEG uses more frequent I frames to enable random access into compressed bitstreams. H.263 uses fewer I frames to achieve better compression.
- Frame coding types: MPEG allows pictures to be coded as I, P, or B frames while H.263 allows pictures to be coded as I or P frames or optionally as PB frames. In MPEG any number of B frames may be included between a pair of I or P frames, while in H.263 the PB mode allows at most one.
- Prediction modes: In support of interlaced video formats, MPEG-2 allows field-based prediction, frame-based prediction, and 16x8 field-based prediction. H.263 only supports frame-based prediction but optionally allows an advanced prediction mode in which 4 motion vectors are allowed per macroblock.
- Motion vector restrictions: MPEG motion vectors must point inside the picture, while H.263 has an unrestricted motion vector mode which allows motion vectors to point outside the picture. The benefits of this mode can be significant, especially for lower-resolution sequences where the boundary macroblocks account for a larger percentage of the video.

3. PROPOSED TRANSCODING ALGORITHM

We developed an MPEG-2 to H.263 transcoding algorithm that addresses each of the factors listed above. The proposed algorithm can be described by the series of block diagrams shown in Figure 2.

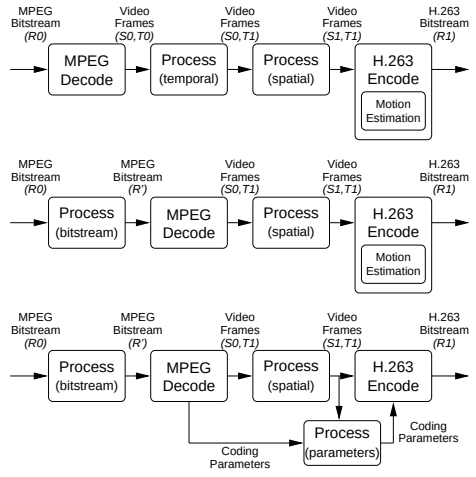

Figure 2: *Development of the proposed approach.*

The top diagram shows the conventional approach to transcoding. First the input MPEG bitstream with bitrate $R0$ is decoded into its decompressed video frames, which have a spatial resolution and temporal frame rate of $S0$ and $T0$. These frames are then processed temporally to a lower frame rate $T1 < T0$ by dropping appropriate frames. The spatial resolution is then reduced to $S1 < S0$ by spatially downsampling the remaining frames. The resulting frames with resolution $S1, T1$ are then re-encoded into an H.263 bitstream with a final bitrate of $R1 < R0$. The memory requirements of this approach are high because of the frame stores required to store the video frames at resolution $S0, T0$. The computational requirements are high because of the operations necessary to decode all the MPEG frames and to perform motion estimation in the H.263 encoder.

The middle diagram shows an improved approach to the problem. By exploiting the picture start codes and frame prediction types used in MPEG, the frame rate of the input bitstream can be reduced prior to MPEG decompression. Specifically, in order to reduce the temporal frame rate, rather than decoding the entire MPEG bitstream and subsequently dropping frames, one may instead examine the bitstream for picture startcodes, determine the picture type from the picture header, then selectively discard the bits that correspond to B pictures. The resulting lower-rate $R' < R0$ bitstream can be decoded into video frames with resolution $S0, T1$. The limitation is that the temporal frame rate can only be reduced by restricted factors, e.g. in the case where two B frames are used between the I and P frames, the temporal frame rate can only be reduced by a factor of 3. The advantages are the reduced processing requirements needed for MPEG decoding and the reduced memory requirements achieved by eliminating the need to store the higher frame rate sequence. In this approach, the computational requirements are still high due to the motion estimation that must be performed in the H.263 encoder.

The bottom diagram shows the proposed approach for MPEG-2 to H.263 transcoding. Once again, the temporal frame rate is reduced at the bitstream layer by exploiting the picture start codes and picture headers. In addition, the computational requirements of the H.263 encoder are reduced by deriving its coding parameters from those given in the input MPEG bitstream. This is advantageous because some of the computations that need to be performed in the H.263 encoder may have already been performed by the original MPEG-2 encoder and may be represented in the transcoder's input MPEG bitstream. Rather than blindly recomputing this information from the decoded, downsampled video frames, the encoder can exploit the information contained in the input bitstream. In other words, much of the information that is derived in the original MPEG encoder can be reused in the transcoder. Specifically, the motion vectors and prediction modes from the input MPEG bitstream are used to estimate the motion vectors and prediction modes used in the H.263 encoder, thus largely bypassing the expensive motion estimation performed in a conventional encoder. A detailed description of the transcoder is given in section 5.

4. MPEG-2 INTERLACED VIDEO CODING

Most video compression algorithms including H.263 and MPEG-1 are designed for progressive video sequences. MPEG-2 was designed to support interlaced video sequences, where two fields make one frame. MPEG-2 provides a number of coding options to support interlaced video. First, each in-

terlaced video frame can be coded as a frame picture in which the two fields are coded as a single unit or as a field picture in which the fields are coded sequentially. Next, MPEG-2 allows macroblocks to be coded in one of five motion compensation modes: frame prediction for frame pictures, field prediction for frame pictures, field prediction for field pictures, 16x8 prediction for field pictures, and dual prime motion compensation [5]. The frame picture and field picture prediction dependencies are shown in Figure 3. For frame pictures, the top and bottom reference fields are the top and bottom fields of the previous I or P frame. For field pictures, the top and bottom reference fields are the most recent top and bottom fields. For example, if the top field is specified to be first, then MVs from the top field can point to the top or bottom fields in the previous frame, while MVs from the bottom field can point to the top field of the current frame or the bottom field of the previous frame. Our discussion focuses on P-frame prediction because the transcoder only processes the MPEG I and P frames. We also focus on field picture coding of interlaced video, and we will not consider dual prime motion compensation.

In MPEG field picture coding, each field is divided into 16x16 macroblocks, each of which can be coded with field prediction or 16x8 motion compensation. In field prediction, the 16x16 field macroblock will contain a field selection bit which indicates whether the prediction is based on the top or bottom reference field and a motion vector which points to the 16x16 region in the appropriate field. In 16x8 prediction, the 16x16 field macroblock is divided into its upper and lower halves, each of which contains 16x8 pixels. Each half has a field selection bit which specifies whether the top or bottom reference field is used and a motion vector which points to the 16x8 pixel region in the appropriate field.

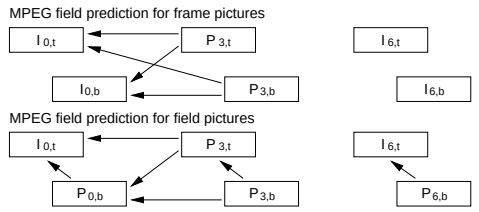

Figure 3: *MPEG-2 motion compensation for interlaced video.* The arrows show the allowable prediction dependencies. The subscripts denote the frame number and field type.

5. AN MPEG-2 TO H.263 TRANSCODER

A number of design decisions must be made when creating an algorithm that transcodes interlaced MPEG-2 video bitstreams to lower-rate progressive video bitstreams. While the input parameters are set by the given input bitstream and while some output parameters may be specified, a number of degrees of freedom will likely exist for other output parameters. For example, the transcoder may be required to reduce the bitrate by a set amount, but freedom may be available in whether this is accomplished by spatial or temporal downsampling or requantization or some combination thereof.

Our design decisions were made with the goal of creating a simple, computationally efficient transcoder. Many of our choices were based on similarities and differences in the details of the MPEG-2 and H.263 coding standards listed in section 2. In this section, we describe the proposed transcoder and discuss the motivation behind many of our design decisions.

5.1. Description

A block diagram of the proposed MPEG-2 to H.263 transcoder is shown in Figure 4. The transcoder accepts an MPEG IPB bitstream as input. The bitstream is scanned for picture start codes and the picture headers are examined to determine the frame type. The bits corresponding to B frames are discarded, while the remaining bits are passed on to the MPEG IP decoder. The decoded frames are downsampled to the appropriate spatial resolution and then passed to the modified H.263 IP encoder. This encoder differs from a conventional H.263 encoder in that it does not perform conventional motion estimation; rather, it uses motion vectors and coding modes computed from the MPEG motion vectors and coding modes and the decoded, downsampled frames. There are a number of ways that this can be done [6]. We choose a partial search method in which the MPEG motion vectors and coding modes are used to form one or more initial estimates for each H.263 motion vector. A set of candidate motion vectors is generated; this set may include each initial estimate and its neighboring vectors, where the size of the neighborhood can vary depending on the available computational resources. The set of candidate motion vectors is tested on the decoded, downsampled frames and the best vector is chosen based on a criteria such as residual energy. A half-pel refinement may be performed and the final mode decision (inter or intra) is then made.

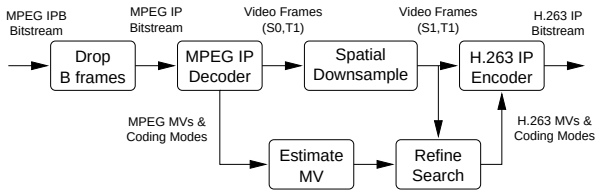

Figure 4: *Detailed block diagram of transcoder.*

5.2. Design Considerations and Details

5.2.1. Source and Target Parameters

Spatial and temporal resolutions: The correspondence between the input and output coded video frames is shown in Figure 5. We reduce the horizontal and vertical spatial resolutions by factors of two because the MPEG-2 interlaced field format provides a natural factor of two reduction in the vertical spatial resolution. Thus, the spatial downsampling is performed by simply extracting the top field of the MPEG-2 interlaced video frame and horizontally downsampling it by a factor of two. This simple spatial downsampling method allows us to avoid the difficulties associated with interlaced to progressive conversions. We reduce the temporal resolution by a factor of three. As discussed earlier, the MPEG-2 picture start codes, picture headers, and

prediction rules make it possible to efficiently discard B-frame data from the bitstream without impacting the remaining I and P frames. Note that even though only the top fields of the MPEG I and P frames are used in the H.263 encoder, both the top and bottom fields must be decoded because of the prediction dependencies that result from the MPEG-2 interlaced field coding modes.

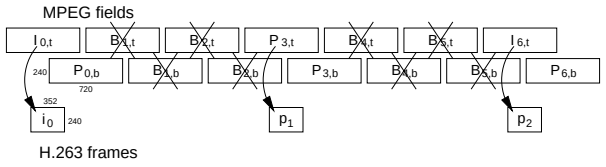

Figure 5: *Correspondence between the input MPEG-2 fields and the output H.263 frames.*

Frame coding types: MPEG-2 allows I, P, and B frames while H.263 allows I and P frames and optionally PB frames. With sufficient memory and computational capabilities, an algorithm can be designed to transcode from any input MPEG coding pattern to any output H.263 coding pattern; such an algorithm was developed by Shanableh in [4]. Rather than transcoding between arbitrary coding patterns, we determine the coding pattern of the target H.263 bitstream based on the coding pattern of the source MPEG-2 bitstream. By aligning the coding patterns of the input and output bitstreams and allowing temporal downsampling, we can achieve a significant improvement in computational efficiency.

Specifically, a natural alignment between the two standards can be obtained by dropping the MPEG B frames and converting the remaining MPEG I and P frames to H.263 I and P frames, thus exploiting the similar roles of P frames in the two standards and exploiting the ease in which B frame data can be discarded from an MPEG-2 bitstream without affecting the remaining I and P frames. Since MPEG-2 sequences typically use an IBBPBBPBB structure, dropping the B frames results in a factor of three reduction in frame rate. While H.263 allows an advanced coding mode of PB pictures, we do not use it because it does not align well with MPEG's IBBPBBPBB structure.

The problem that remains is to convert the MPEG-coded interlaced I and P frames to the spatially downsampled H.263-coded progressive I and P frames. The problem of frame conversions can be thought of as manipulating prediction dependencies in the compressed data; this topic was addressed in [7] for MPEG progressive frame conversions. Our MPEG-2 to H.263 transcoding algorithm requires three types of frame conversions: (1) MPEG I field to H.263 I frame, (2) MPEG I field to H.263 P frame, and (3) MPEG P field to H.263 P frame. The first is straightforward. The latter two require the transcoder to efficiently calculate the H.263 motion vectors and coding modes from those given in the MPEG-2 bitstream. When using the partial search method described in section 5.1, the first step is to create one or more initial estimates of each H.263 motion vector from the MPEG-2 motion vectors. In the following two sections, we discuss the methods used to accomplish this for MPEG I field to H.263 P frame conversions and for MPEG P field to H.263 P frame conversions.

5.2.2. I field to P frame conversion

MPEG applications generally use frequent I frames to enable random access into the bitstream, while H.263 applications generally use fewer I frames to achieve higher compression ratios. In the proposed transcoder, it is often necessary to convert MPEG I fields to H.263 P frames. Motion information generally is not given for MPEG I frames, therefore we must generate the resulting H.263 P frame motion vectors by using the motion information provided for the surrounding frames. In our simplest solution, the motion vector that was computed for the same spatial location in the previous H.263 P frames is used as the initial estimate of the H.263 motion vector in the current frame. This is a rather crude approach for estimating the motion vectors but ranks well in computational simplicity. The performance depends on the temporal correlation of the motion in the coded video. Improved estimates can be obtained by considering the previous frame's higher resolution MPEG-2 motion vectors rather than its computed lower resolution H.263 motion vectors and by considering the motion vectors from the next MPEG-2 P frame as well; however this requires higher complexity so that the future motion vectors can be retrieved. If additional resources are available, the motion vectors from the intermediate MPEG-2 B frames can also be considered; these would provide a better representation of the the motion between the two frames, but at a much greater complexity.

5.2.3. P field to P frame conversion

As discussed in section 5.2.1, the transcoder horizontally downsamples the top field of the decoded frames by a factor of two. After horizontal downsampling, 32x16 regions in the MPEG-2 field correspond to 16x16 regions of the H.263 frame, i.e. two adjacent MPEG-2 field macroblocks correspond to one H.263 frame macroblock. In this section, we discuss our method for choosing the initial estimates of the H.263 motion vectors from the motion vectors given in the two corresponding MPEG-2 macroblocks. These initial estimates are used to compute the final H.263 motion vectors and coding modes as described in section 5.1.

The MPEG-2 prediction modes used for field coding of interlaced video were discussed in section 4. We focus on MPEG-2 field pictures for simplicity; straightforward modifications can be made to accommodate MPEG-2 frame pictures. Since each MPEG-2 macroblock can have zero (intra), one (field prediction), or two (16x8 prediction) motion vectors and since two MPEG-2 macroblocks correspond to one H.263 macroblock; we will have anywhere from zero to four motion vectors to consider when estimating one H.263 motion vector. The task at hand is to rank these MPEG-2 motion vectors according to how well they perform as an initial estimate of the H.263 motion vector.

We rank the MPEG-2 motion vectors based on the their relevance to the H.263 vector. The relevance is measured by two factors: the field correspondence and the size of the region it represents. Since the H.263 frames correspond to the top fields of the CCIR-601 frames, the MPEG-2 motion vectors that point to the top reference fields provide a better estimate and therefore are considered more relevant than those that point to the bottom reference fields. MPEG-2 motion vectors can correspond to 16x16 or 16x8 regions which after horizontal downsampling correspond to

8x16 or 8x8 regions in the H.263 macroblock; the motion vectors corresponding to larger regions are considered to have higher relevance. After ranking, the motion vectors are adjusted for the horizontal downsampling factor and the highest ranked vectors are selected as the initial estimates of the H.263 motion vector. While the proposed approach leads to good estimates of the motion vectors, further refinement and computations can lead to improved estimates and thereby result in improved prediction and coding performance for the output H.263 bitstream. For example, if additional computational resources are available it may be useful to also consider the motion vectors given in neighboring MPEG-2 macroblocks as well as the motion vectors given in the bottom field of the current MPEG-2 frame.

6. EXPERIMENTAL RESULTS

The conventional and proposed MPEG-2 to H.263 transcoders were implemented in software based on the public-domain MPEG-2 and H.263 code [8, 9]. The input MPEG-2 video bitstreams contained coded CCIR-601 interlaced video sequences which have 720x480 pixels/frame at 30 frames/sec. Each frame consisted of two fields, each with 720x240 pixels. The MPEG-2 video was field coded at a data rate of 5 Mbps. These values are representative of those typically used for coding NTSC video. Temporal downsampling was achieved by discarding the MPEG-2 bits corresponding to B frames. Spatial downsampling was achieved by extracting the top field of the decoded CCIR-601 frame and horizontally filtering and downsampling this field using the filters described in MPEG-2 Test Model 5; the rightmost 8 columns were discarded to form the SIF-resolution frame. The output H.263 bitstreams contained SIF progressive sequences, which have 352x240 pixels at 10 frames/sec. The resulting H.263 bitstreams had data rates of 500 kbps.

The partial search method was used to compute the output H.263 motion vectors and coding modes from those given in the input MPEG-2 bitstream. In these plots, we report results for which only one initial estimate was allowed for the partial search. The set of candidate motion vectors consisted of the initial estimate and its neighboring vectors where the size of the neighborhood was varied from 0 to +/-7 pixels. For the MPEG-2 I field to H.263 P frame conversion, the H.263 motion vector computed for the previous frame was used as the initial estimate. For the MPEG-2 P field to H.263 P frame conversion, the top ranked motion vector was used as the initial estimate.

The resulting PSNR plots for the Bus and Carousel MPEG test sequences are shown in Figure 6. In software simulations, the proposed transcoding algorithm achieved a 5x speedup over the conventional approach with the PSNR performance shown in the plots. The top trace in each plot shows the PSNRs of the decoded, downsampled I and P frames of the original MPEG-2 bitstream. The remaining traces show the PSNRs of the H.263 decoded frames of the transcoded bitstreams. From bottom to top, the traces show the PSNRs that result when increasing the neighborhood size from 0 to +/-4 pixels and the PSNRs that result when performing a conventional motion estimation search in the transcoder. A more in-depth coverage of the MPEG-2 to H.263 transcoding experiments is given in [10].

There are a number of avenues for future work. First,

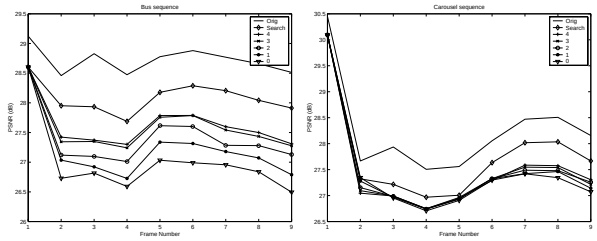

Figure 6: *PSNR Plots for the Bus and Carousel sequences.*

while the proposed approach leads to good estimates of the motion vectors, further refinement and computations can lead to improved prediction and coding performance and are currently being investigated. Next, it is likely that these results vary with bitrate and resolution. In some regimes the coding error may be dominated by quantization noise, while in others it may be dominated by motion vector errors. Additional work should be performed to better understand the transcoding performance as a function of the source and target bitrates and resolutions and the input coded video sequence. Finally, unlike MPEG-2, H.263 has advanced coding modes to support motion vectors for 8x8 blocks and unrestricted motion vectors which point outside the picture. These modes can provide significant improvement in H.263 coding performance and are currently being incorporated in our transcoder.

7. REFERENCES

[1] H. Sun, W. Kwok, and J. Zdepski, "Architectures for MPEG compressed bitstream scaling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, April 1996.

[2] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding MPEG bitstreams," *Signal Processing: Image Communication*, vol. 8, September 1996.

[3] N. Bjork and C. Christopoulos, "Transcoder architectures for video coding," in *IEEE International Conference on Image Processing*, (Seattle, WA), May 1998.

[4] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding MPEG:1,2 to H.263," in *International Packet Video Workshop*, (New York City, NY), April 1999.

[5] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG2.* Digital Multimedia Standards Series, Chapman and Hall, 1997.

[6] S. Wee, "Reversing motion vector fields," in *IEEE International Conference on Image Processing*, (Chicago, IL), October 1998.

[7] S. Wee, "Manipulating temporal dependencies in compressed video data with applications to compressed-domain processing of MPEG video," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Phoenix, AZ), March 1999.

[8] MPEG Software Simulation Group, http://www.mpeg.org/MPEG/MSSG, *MPEG-2 Video Codec*, 1.2 ed.

[9] University of British Columbia, http://spmg.ece.ubc.ca/h263plus, *H.263+ Public-Domain Code*, 3.2 ed.

[10] N. Feamster and S. Wee, "An MPEG-2 to H.263 transcoder," in *SPIE Voice, Video, and Data Communications Conference*, (Boston, MA), September 1999.