Video Transcoding Architectures and Techniques: An Overview

Anthony Vetro, Charilaos Christopoulos, and Huifang Sun

The development of multimedia systems has had a major influence in the area of image and video coding. The problem of interactivity and integration of video data with computer, cellular, and television systems is relatively new and subject to a great deal of research world wide.

As the number of networks, types of devices, and content representation formats increase, interoperability between different systems and different networks is becoming more important. Thus, devices such as gateways, multipoint control units, and servers must be developed to provide a seamless interaction between content creation and consumption. Transcoding of video content is one key technology to make this possible. In general, a transcoder relays video signals from a transmitter in one system to a receiver in another system (or network).

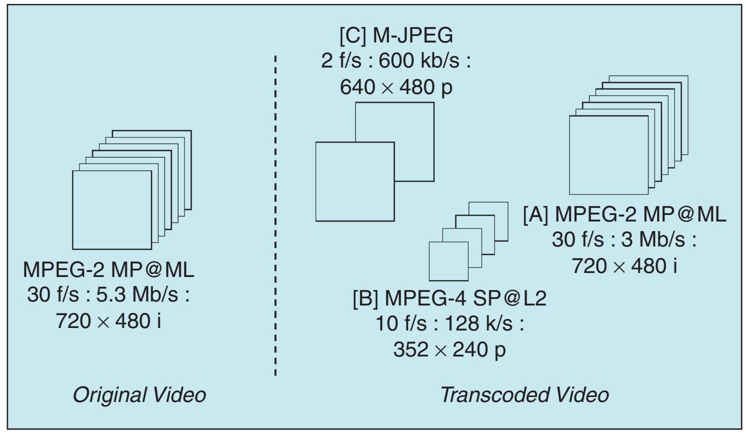
Generally speaking, transcoding can be defined as the conversion of one coded signal to another. While this definition can be interpreted quite broadly, it should be noted that research on video transcoding is usually very focused. In the earliest work on transcoding, the majority of interest focused on reducing the bit rate to meet an available channel capacity. Additionally, researchers investigated conversions between constant bit-rate (CBR) streams and variable bit-rate (VBR) streams to facilitate more efficient transport of video. As time moved on and mobile devices with limited display and processing power became a reality, transcoding to achieve spatial resolution reduction, as well as temporal resolution reduction, has also been studied. Furthermore, with the introduction of packet radio services over mobile access networks, error-resilience video transcoding has gained a significant amount of attention lately, where the aim is to increase the resilience of the original bit stream to transmission errors. Some of these common transcoding operations are illustrated in Figure 1.

In all of these cases, it is always possible to use a cascaded pixel-domain approach that decodes the original signal,

1053-5888/03/$17.00©2003IEEE

performs the appropriate intermediate processing (if any), and fully reencodes the processed signal subject to any new constraints. While we also view this as a form of transcoding, it is often very costly to do so and more efficient techniques are typically utilized. This quest for efficiency is the major driving force behind most of the transcoding activity that we have seen so far. Of course, any gains in efficiency should have a minimal impact on the quality of the transcoded video.

Throughout this article, we concentrate on the transcoding of block-based video coding schemes that use hybrid discrete cosine transform (DCT) and motion compensation (MC). In such schemes, the frames of the video sequence are divided into macroblocks (MBs), where each MB typically consists of a luminance block (e.g., of size 16×16, or alternatively, four 8×8 blocks) along with corresponding chrominance blocks (e.g., 8×8 Cb and 8×8 Cr). This article emphasizes the processing that is done on the luminance components of the video. In general, the chrominance components can be handled similarly and will not be discussed in this article.



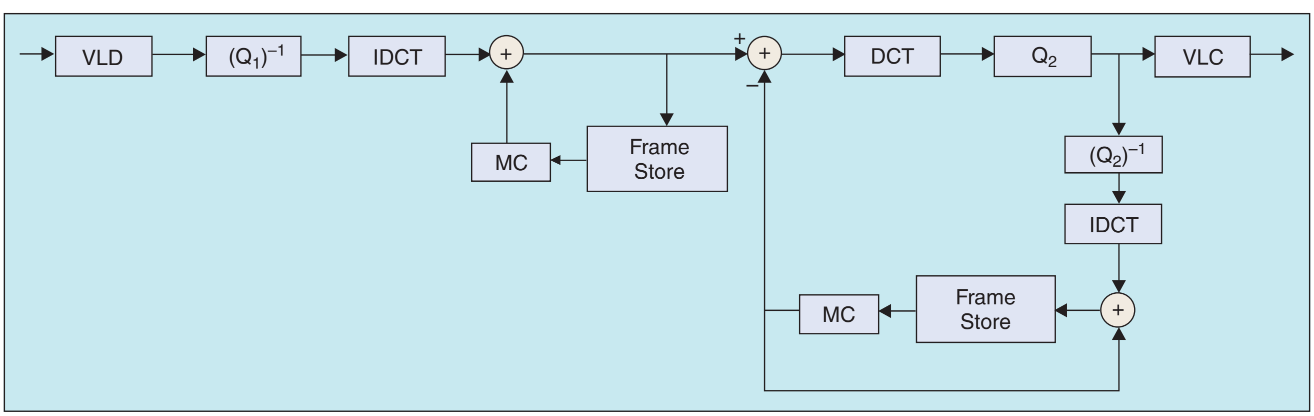
1. Illustration of common video transcoding operations. Original video is encoded in an MPEG-2 format (Main Profile at Main Level = MP@ML) at 5.3 Mb/s. The input resolution is 720×480 i (interlaced), and the temporal rate is 30 frames-per-second (f/s). [A] Original video is transcoded to a reduced bit-rate of 3 Mb/s. [B] Original video is transcoded to an MPEG-4 format (Simple Profile at Level 2 = SP@L2) at 128 kb/s. The output resolution is 352×240 p (progressive) and the temporal rate is 10 f/s. [C] Original video is transcoded to a Motion-JPEG (M-JPEG) sequence of images at a temporal rate of 2 f/s, bit-rate of 600 kb/s, and output resolution of 640×480 p.

The article is organized as follows. We first provide an overview of the techniques used for bit-rate reduction and the corresponding architectures that have been proposed. Then, we describe recent advances regarding spatial and temporal resolution reduction techniques and architectures. Additionally, a brief overview of error resilient transcoding is also provided, as well as a discussion of scalable coding techniques and how they relate to video transcoding. Finally, this article ends with concluding remarks, including pointers to other works on video transcoding that have not been covered in this article, as well as some future directions.

Bit-Rate Reduction

The objective of bit-rate reduction is to reduce the bit rate while maintaining low complexity and achieving the highest quality possible. Applications requiring this type of conversion include television broadcast and Internet streaming. Ideally, the quality of the reduced rate bit stream should have the quality of a bit stream directly generated with the reduced rate. The most straightforward way to achieve this is to decode the video bit stream and fully reencode the reconstructed signal at the new rate. This approach is illustrated in Figure 2. The best performance can be achieved by calculating new motion vectors and mode decisions for every MB at the new rate [2]. However, significant complexity saving can be achieved, while still maintaining acceptable quality, by reusing information contained in the original incoming bit streams and also considering simplified architectures [1]-[7].

In the following, we review the progress made over the past few years on bit-rate reduction architectures and techniques, where the focus has been centered on two specific aspects, complexity and drift reduction. Drift can be explained as the blurring or smoothing of successively predicted frames. It is caused by the loss of high frequency data, which creates a mismatch between the actual refer-



2. Illustration of cascaded pixel-domain transcoding architecture for bit-rate reduction.

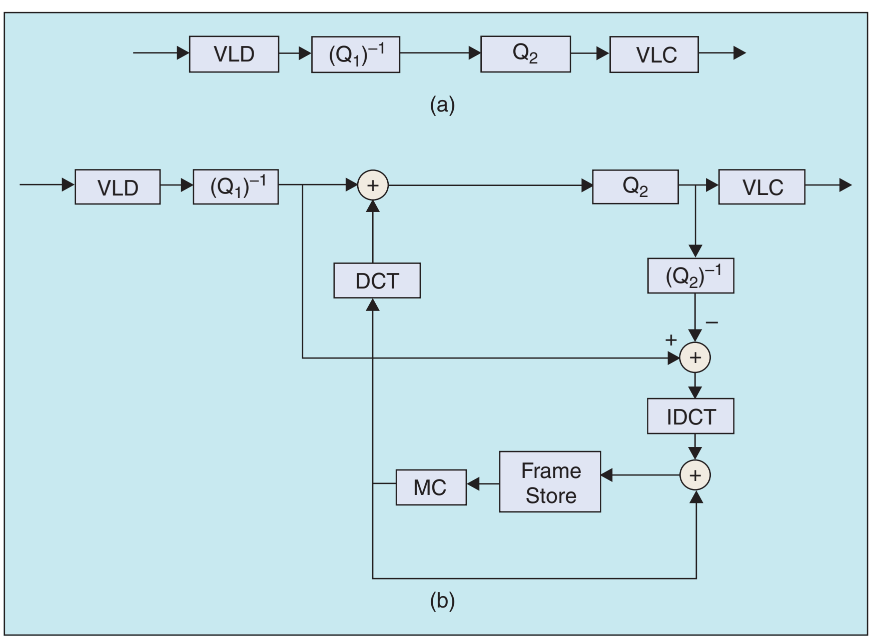
ence frame used for prediction in the encoder and the degraded reference frame used for prediction in the transcoder and decoder. To demonstrate the tradeoff between complexity and quality, we will consider two types of systems, a closed-loop and an open-loop system.

Transcoding Architectures

Figure 3 shows an open-loop system in (a) and a closed-loop systems in (b). In the open-loop system, the bit stream is variable-length decoded (VLD) to extract the variable-length code words corresponding to the quantized DCT coefficients, as well as MB data corresponding to the motion vectors and other MB-level information. In this scheme, the quantized coefficients are inverse quantized and then simply requantized to satisfy the new output bit rate. Finally, the requantized coefficients and stored MB-level information are variable length coded (VLC). An alternative open-loop scheme, which is not illustrated here, but is even less complex than the one shown in Figure 3(a), is to directly cut high frequency data from each MB [2]. To cut the high frequency data without actually doing the VLD, a bit profile for the AC coefficients is maintained. As MBs are processed, code words corresponding to high-frequency coefficients are eliminated as needed so that the target bit rate is met. Along similar lines, techniques to determine the optimal breakpoint of non-zero DCT coefficients (in a zig-zag order) were presented in [3]. This procedure is carried out for each MB, so that distortion is minimized and rate constraints are satisfied. These two alternatives to requantization may also be used in the closed-loop systems described below, but their impact on the overall complexity is less. Regardless of the techniques used to achieve the reduced rate, open-loop systems are relatively simple since a frame memory is not required and there is no need for an inverse IDCT. In terms of quality, better coding efficiency can be obtained by the requantization approach since the variable-length codes that are used for the requantized data will be more efficient. However, open-loop architectures are subject to drift.

In general, the reason for drift is due to the loss of high-frequency information. Beginning with the I-frame, which is a reference for the next P-frame, high-frequency information is discarded by the transcoder to meet the new target bit rate. Incoming residual blocks are also subject to this loss. When a decoder receives this transcoded bit stream, it will decode the I-frame with reduced quality and store it in memory. When it is time to decode the next P-frame, the degraded I-frame is used as a predictive component and added to a degraded residual component. Considering that the purpose of the residual is to accurately represent the difference between the original signal and the motion-compensated prediction and now both the residual and predictive components are different than what was originally derived by the encoder, errors would be introduced in the reconstructed frame. This error is a result of the mismatch between the predictive and residual components. As time goes on, this mismatch progressively increases, resulting in the reconstructed frames becoming severely degraded.

The architecture shown in Figure 3(b) is a closed-loop system and aims to eliminate the mismatch between predictive and residual components by approximating the cascaded decoder-encoder architecture [4]. The main difference in structure between the cascaded pixel-domain architecture and this simplified scheme is that reconstruction in the cascaded pixel-domain architecture is performed in the spatial domain, thereby requiring two reconstruction loops with one DCT and two IDCTs. On the other hand, in the simplified structure that is shown in Figure 3(b), only one reconstruction loop is required with one DCT and one IDCT. In this structure, some arithmetic inaccuracy is introduced due to the nonlinear nature in which the reconstruction loops are combined. However, it has been found the approximation has little effect on the quality [4]. With the exception of this slight inaccuracy, this architecture is mathematically equivalent to a cascaded decoder-encoder approach. In [8], additional causes of drift, e.g., due to floating-point inaccuracies, have been further studied. Overall though, in comparison to the open-loop architectures discussed earlier, drift is eliminated since the mismatch between predictive and residual components is compensated for.



3. Simplified transcoding architectures for bit-rate reduction: (a) open-loop, partial decoding to DCT coefficients, then requantize and (b) closed-loop, drift compensation for requantized data.

Motion Compensation in the DCT Domain

The closed-loop architecture described in the previous section provides an effective transcoding structure in which the MB reconstruction is performed in the DCT domain. However, since the memory stores spatial domain pixels, the additional DCT/IDCT is still needed. This can be avoided though by utilizing the compressed-domain methods for MC proposed by Chang and Messerschmidt [9]. In this way, it is possible to reconstruct reference frames without decoding to the spatial domain; several architectures describing this reconstruction process in the compressed domain have been proposed [10]-[12]. It was found that decoding completely in the compressed-domain could yield equivalent quality to spatial-domain decoding [10]. However, this was achieved with floating-point matrix multiplication and proved to be quite costly. In [12] this computation was simplified by approximating the floating-point elements by power-of-two fractions so that shift operations could be used, and in [13], simplifications have been achieved through matrix decomposition techniques.

Regardless of the simplification applied, once the reconstruction has been accomplished in the compressed domain, one can easily requantize the drift-free blocks and VLC the quantized data to yield the desired bit stream. In [12], the bit reallocation has been accomplished using the Lagrangian multiplier method. In this formulation, sets of quantizer steps are found for a group of MBs so that the average distortion caused by transcoding error is minimized.

In [14], further simplifications of the DCT-based MC process were achieved by exploiting the fact that the stored DCT coefficients in the transcoder are mainly concentrated in low-frequency areas. Therefore, only a few low-frequency coefficients are significant and an accurate approximation to the MC process that uses all coefficients can be made.

CBR to VBR Conversion

While the above architectures have focused on general bit-rate reduction techniques for the purpose of transmitting video over band-limited channels, the conversion between CBR and VBR streams to facilitate more efficient transport of video has also been studied [16]. In this work, the authors exploit the available channel bandwidth of an ATM network and adapt the CBR streams accordingly. This is accomplished by first reducing the bit stream to a VBR stream with a reduced average rate and then segmenting the VBR stream into cells and controlling the cell generation rate by a traffic shaping algorithm.

Simulation Results

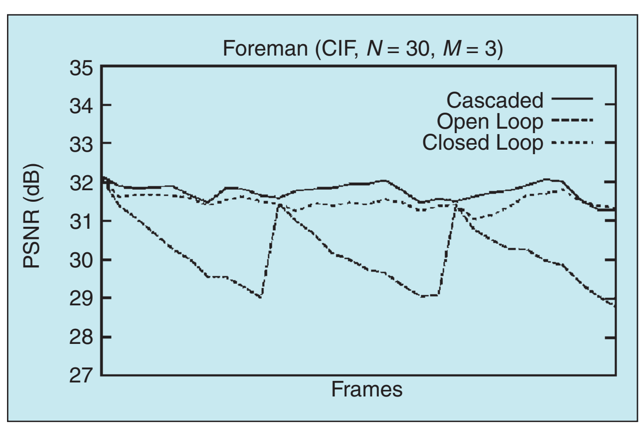
In Figure 4, a frame-based comparison of the quality among the cascaded pixel-domain, open-loop, and closed-loop architectures is shown. The input to the transcoder is an MPEG-1 video bit stream of the Foreman sequence at CIF resolution coded at 2 Mb/s with GOP (group of pictures) structure of $N = 30$ and $M = 3$. A total of 90 frames is used in this experiment. The transcoded output is in the MPEG-4 Visual format (Simple Profile) and reencoded with a fixed quantization parameter of 15. To illustrate the effect of drift in this plot, the peak signal-to-noise ratio (PSNR) of the luminance component for only the I- and P-frames is shown. It is evident that the open-loop architecture suffers from severe drift, and the quality of the simplified closed-loop architecture is very close to that of the cascaded pixel-domain architecture.

It should be emphasized that the main point of the results presented here is to illustrate the drift problem with the open-loop architecture and the drift compensation capabilities with the simplified closed-loop architecture. Although the results would vary slightly with different syntax formats, GOP parameters, bit rates, and sequences, we maintain that the overall impact of these factors would not alter the conclusions of this result. The same holds true for results presented later.

As the number of networks, types of devices, and content representation formats increase, interoperability between different systems and different networks is becoming more important.

Spatial Resolution Reduction

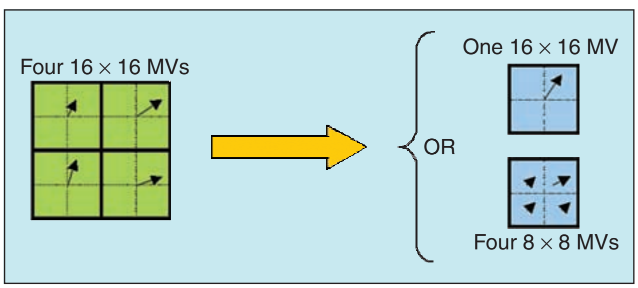
These days, a massive amount of compressed video content captured at a high spatial resolution and encoded with high quality is being created. Two of the major catalysts feeding this phenomenon are the growing popularity of DVD and the availability of broadband access networks. With the emergence of mobile multimedia-capable devices and the desire for users to access video originally captured in a high resolution, there is a strong need



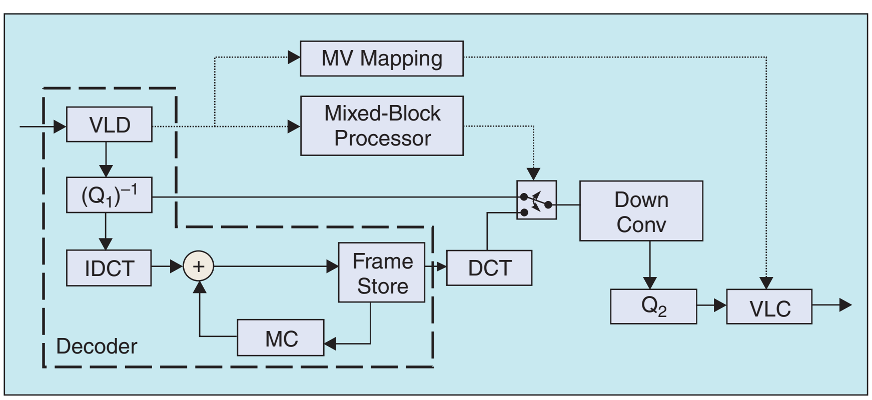
▲ *4. Frame-based comparison of PSNR of the luminance component for cascaded pixel-domain, open-loop, and closed-loop architectures for bit-rate reduction.*

for efficient ways to reduce the spatial resolution of video for delivery to such devices.

Similar to bit-rate reduction, the cascaded pixel-domain architecture for reduced spatial resolution transcoding refers to the subsequent decoding, spatial-domain down-sampling, followed by a full reencoding. From the literature, we find that some researchers have focused on the efficient reuse of MB-level data in the context of the cascaded pixel-domain architecture, while others have explored the possibility of new alternate architectures. In [6] and [7], the problems associated with mapping motion vectors and MB-level data were addressed. The performance of motion vector refinement techniques in the context of resolution conversion was also studied in this work. The primary focus of the work in [17] was on motion vector scaling techniques. In [18], the authors propose to use DCT-domain down-scaling and MC for transcoding, in which an algorithm to decide whether to code the MB as intra, inter without MC, or inter with MC, was proposed. With the proposed two-loop architecture from [18], computational savings of 40% have been reported with a minimal loss in quality. In [19], a comprehensive study on the transcoding to lower spatio-temporal resolutions and to different encoding formats has been provided based on the reuse of motion parameters. In this work, a full decoding and encoding loop was employed. With the reuse of MB information, a significant reduction in processing time was achieved. This work was extended to the DCT domain in [20]. In [21], the source of drift errors for reduced spatial resolution reduction transcoding was analyzed. Based on this analysis, several new architectures, including an intra-refresh architecture, were proposed.



▲ *5. Illustration of 4:1 and 1:1 motion vector mapping.*



▲ *6. Intra-refresh architecture for reduced spatial resolution transcoding.*

In the following, the key points from the above works are reviewed, including motion vector scaling algorithms, DCT-domain down conversion, and the mapping of MB-level information to the lower spatial resolution. Also, the concepts of the intra-refresh architecture will be discussed. Throughout this section, a reduction factor of two in both the horizontal and vertical resolution is assumed. Extensions of the described techniques to other noninteger scaling factors are considered in [22] and [23], however due to limitations in space, those techniques are not covered in this article.

Motion Vector Mapping

When down-sampling four MBs to one MB, the associated motion vectors have to be mapped, where the number of motion vectors that may be associated with an MB depends on the standard being used and the coding tools available in a given profile or extension. Figure 5 illustrates the general problem of motion vector mapping. Several methods to perform the particular mapping illustrated in this figure have been described in past works [6], [7], [17], [19], [24]. To map from four motion vectors, i.e., one for each MB in a group, to one motion vector for the newly formed MB, a weighted average or median filters can be applied. This is referred to as a 4:1 mapping. However, with certain compression standards, such as MPEG-4 Visual [25] and H.263 [26], there is support in the syntax for advanced prediction modes that allow one motion vector per 8×8 luminance block. (It should be noted that the use of one motion vector per 8×8 block in the H.263 standard is supported in the extensions defined by the standard, i.e., this tool is not supported in the baseline specification.) In this case, each motion vector is mapped from a 16×16 MB in the original resolution to an 8×8 block in the reduced resolution MB with appropriate scaling by two. This is referred as a 1:1 mapping. While 1:1 mapping provides a more accurate representation of the motion, it is sometimes inefficient to use since more bits must be used to code four motion vectors. An optimal mapping would adaptively select the best mapping based on a rate-distortion criterion. A good evaluation of the quality that can be achieved using the different motion vector mapping algorithms can be found in [6], [7], and [19].

Because MPEG-2 supports interlaced video, we also need to consider field-based MV mapping. In [27], the top-field motion vector was simply used. An alternative scheme that averages the top and bottom field motion vectors under certain conditions was proposed in [21]. However, it is our opinion that the appropriate motion vector mapping technique is dependent on the down-conversion scheme used. We feel this is particularly important for interlaced data, where the target output may be a progressive frame. However, further study on the relation between motion vector mapping and the texture down-conversion is needed to confirm this.

DCT-Domain Down Conversion

The most intuitive way to perform down conversion in the DCT domain is to only retain the low-frequency coefficients of each block and recompose the new MB using the compositing techniques proposed in [9]. Specifically, for conversion by a factor of 2, only the 4×4 DCT coefficients of each 8×8 block in a MB are retained; these low frequency coefficients from each block are then used to form the output MB. A set of DCT-domain filters can been derived by cascading these two operations. More sophisticated filters that attempt to retain more of the high frequency information, such as the filters derived in [28] and [29] and references therein, may also be considered. The filters used in this work perform the down-conversion operations on the rows and columns of the MB using separable one-dimensional filters. These down-conversion filters can be applied in both the horizontal and vertical directions and to both frame-DCT and field-DCT blocks. Variations of this filtering approach to convert field-DCT blocks to frame-DCT blocks, and vice-versa, have also been derived in [10].

Conversion of MB Type

In transcoding video bit streams to a lower spatial resolution, a group of four MBs in the original video corresponds to one MB in the transcoded video. To ensure that the down-sampling process will not generate an output MB in which its subblocks have different coding modes, e.g., both inter- and intra-subblocks within a single MB, the mapping of MB modes to the lower resolution must be considered. Three possible methods to overcome this problem when a so-called mixed block is encountered are outlined below [6], [21].

In the first method, ZeroOut, the MB modes of the mixed MBs are all modified to intermode. The MVs for the intra-MBs are reset to zero and so are corresponding DCT coefficients. In this way, the input MBs that have been converted are replicated with data from corresponding blocks in the reference frame. The second method, Intra-Inter, maps all MBs to intermode, but the motion vectors for the intra-MBs are predicted. The prediction can be based on the data in neighboring blocks, which can include both texture and motion data. As an alternative, we can simply set the motion vector to be zero, depending on which produces less residual. In an encoder, the mean absolute difference of the residual blocks is typically used for mode decision. The same principles can be applied here. Based on the predicted motion vector, a new residual for the modified MB must be calculated. In the third method, Inter-Intra, the MB modes are all modified to intramode. In this case, there is no motion information associated with the reduced-resolution MB, therefore all associated motion vector data is reset to zero and the intra-DCT coefficients are generated to replace the inter-DCT coefficients.

When it comes to processing requirements, there are tradeoffs that could be made between spatial and temporal resolution.

It should be noted that to implement the Intra-Inter and Inter-Intra methods, we need a decoding loop to reconstruct the full-resolution picture. The reconstructed data is used as a reference to convert the DCT coefficients from intra-to-inter or inter-to-intra. For a sequence of frames with a small amount of motion and a low-level of detail, the low complexity strategy of zero-out can be used. Otherwise, either Intra-Inter or Inter-Intra should be used. The performance of Inter-Intra is a little better than intra-inter, because Inter-Intra can stop drift propagation by transforming interblocks to intrablocks.

Intra-Refresh Architecture

In reduced resolution transcoding, drift error is caused by many factors, such as requantization, motion vector truncation, and down-sampling. Such errors can only propagate through intercoded blocks. By converting some percentage of intercoded blocks to intracoded blocks, drift propagation can be controlled. In the past, the concept of intra-refresh has successfully been applied to error-resilience coding schemes [30], and it has been found that the same principle is also very useful for reducing the drift in a transcoder [21].

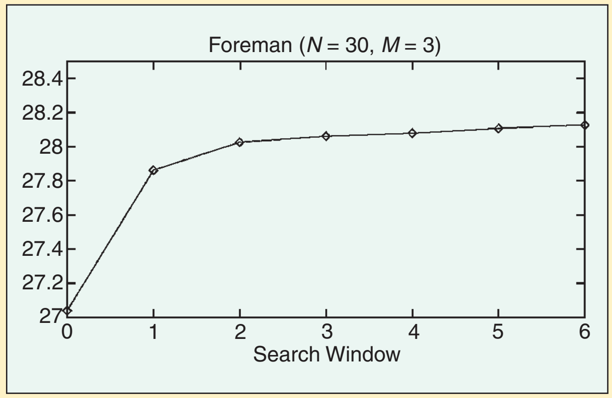
The intra-refresh architecture for spatial resolution reduction is illustrated in Figure 6. In this scheme, output MBs are subject to a DCT-domain down-conversion, requantization, and variable-length coding. Output MBs are either derived directly from the input bit stream, i.e., after variable-length decoding and inverse quantization, or retrieved from the frame store and subject to a DCT. Output blocks that originate from the frame store are independent of other data, hence coded as intrablocks; there is no picture drift associated with these blocks.

The decision to code an intrablock from the frame store depends on the MB coding modes and picture statistics. In the first case, based on the coding mode, an output MB is converted if the possibility of a mixed block is detected. In the second case, based on the picture statistics, the motion vector and residual data are used to detect blocks that are likely to contribute to a larger drift error. For this case, picture quality can be maintained by employing an intracoded block in its place. Of course, the increase in the number of intrablocks must be compensated for by the rate control by adjusting the quantization parameters so that the target rate can accurately be met. This is needed since intrablocks usually require more bits to code. Further details on the rate control can be found in [21].

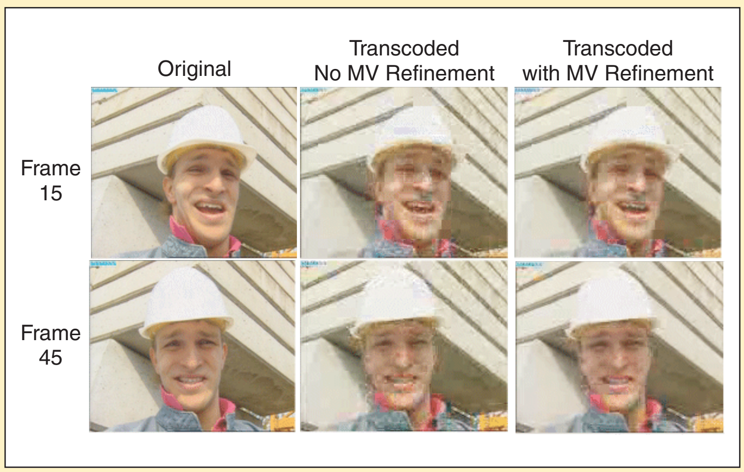
Motion Vector Refinement

In all of the transcoding methods described here, significant complexity is reduced by assuming that the motion vectors computed at the original bit rate are simply reused in the reduced-rate bit stream. It has been shown that reusing the motion vectors in this way leads to nonoptimal transcoding results due to the mismatch between prediction and residual components [6], [7], [15]. To overcome this loss of quality without performing a full motion reestimation, motion vector refinement schemes have been proposed. Typically, the search window used for motion vector refinement is relatively small compared to the original search window, e.g., [–2, +2]. This not only keeps the added complexity down, but it provides a significant amount of the achievable gains. Such schemes can be easily used with most bit-rate reduction architectures for improved quality, as well as the spatial and temporal resolution reduction architectures. A comparison of results obtained with and without motion vector refinement is presented below in the context of spatial resolution reduction. The impact on the search window size is also illustrated. Additional techniques and simulation results for motion vector refinement can be found in [6], [7], [15], [36], and [37].

The simulation results provided here illustrate the impact of motion vector refinement techniques for spatial resolution reduction. We use the same input bit stream as used in Figure 4, i.e., CIF resolution Foreman coded as an MPEG-1 video bit stream at 2 Mb/s with a GOP structure of $N = 30, M = 3$. The QCIF resolution output is transcoded to an MPEG-4 visual format (simple profile) with a bit rate of 64 kb/s and frame-rate of 10 f/s. It can be seen from the plot in Figure 7 that the average PSNR of the luminance component increases as a function of the search window size. However, a very small search window achieves the majority of the gain. This is due to the fact that the majority of blocks find the best-matching motion vector (according to the specified criterion) within this range. Increasing the search window farther allows more blocks to find their best match; since the number of blocks that will find a better match is smaller, however, the overall gain is less. It should be noted that finding a better match will decrease the residuals that need to be coded for each MB, hence allowing a finer quantization (better quality) under the same rate constraints. In Figure 8, sample frames are displayed to compare the visual quality of transcoded frames with and without motion vector refinement. It is evident from these frames that the motion vector refinement process eliminates a significant amount of noise in the reconstructed output.



▲ *7. Average PSNR of the luminance component as a function of the motion vector refinement search window.*



▲ *8. Sample frames to compare the transcoded quality with and without motion vector refinement. For the transcoded output using refinement, a search window of three is used.*

Temporal Resolution Reduction

Reducing the temporal resolution of a video bit stream is a technique that may be used to reduce the bit-rate requirements imposed by a network, to maintain higher quality of coded frames, or to satisfy processing limitations imposed by a terminal. For instance, a mobile terminal equipped with a 266-MHz general-purpose processor may only be capable of decoding and displaying 10 f/s. In another instance, the terminal may simply wish to conserve its battery life at the cost of receiving fewer frames. In both of these instances, one should keep in mind the dependencies that exist, such as the particular coding format, the given spatial resolution, power consumption properties, as well as the efficiency of the implementation. Also, when it comes to processing requirements, there are tradeoffs that could be made between spatial and temporal resolution.

As discussed earlier, motion vectors from the original bit stream are typically reused in bit-rate reduction and spatial resolution reduction transcoders to speed up the reencoding process. In the case of spatial resolution reduction, the input motion vectors are mapped to the lower spatial resolution. For temporal resolution reduction, we are faced with a similar problem in that it is necessary to estimate the motion vectors from the current frame to the previous nonskipped frame that will serve as a reference frame in the receiver. The general problem is illustrated in Figure 9.

Solutions to this problem have been proposed in [15], [32], and [33]. Assuming a pixel-domain transcoding architecture, this reestimation of motion vectors is all that needs to be done since new residuals corresponding to the reestimated motion vectors will be calculated. However, if a DCT-domain transcoding architecture is used, a method of reestimating the residuals in the DCT domain is needed. A solution to this problem has been described in [34]. In [35], the issue of motion vector and residual mapping has been addressed in the context of a combined spatio-temporal reduction in the DCT domain based on the intra-refresh architecture described earlier. The key points of these techniques will be discussed in the following.

Motion Vector Reestimation

As described in [15], [32], and [33], the problem of reestimating a new motion vector from the current frame to a previous nonskipped frame can be solved by tracing the motion vectors back to the desired reference frame. Since the predicted blocks in the current frame are generally overlapping with multiple blocks, bilinear interpolation of the motion vectors in the previous skipped frame may be used, where the weighting of each input motion vector is proportional to the amount of overlap with the predicted block. In the place of this bilinear interpolation, a dominant vector selection scheme as proposed in [15] and [35] may also be used, where the motion vector associated with the largest overlapping region is chosen.

To trace back to the desired reference frame in the case of skipping multiple frames, the above process can be repeated. It is suggested, however, that a refinement of the resulting motion vector be performed for better coding efficiency. In [33], an algorithm to determine an appropriate search range based on the motion vector magnitudes and the number of frames skipped has been proposed. To dynamically determine the number of skipped frames and maintain smooth playback, frame rate control based on characteristics of the video content have also been proposed [32], [33].

Residual Reestimation

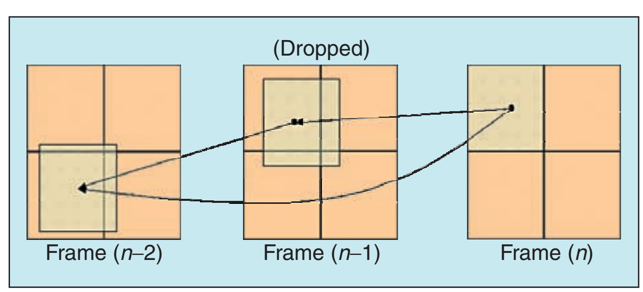
The problem of estimating a new residual for temporal resolution reduction is primarily an issue for DCT-domain transcoding architectures. With pixel-domain architectures, the residual between the current frame and the new reference frame can be easily computed given the new motion vector estimates. For DCT-domain transcoding architectures, this calculation should be done directly using DCT-domain MC techniques [9]. A novel architecture to compute this new residual in the DCT domain has been presented in [34] and [35]. In this work, the authors utilize direct addition of DCT coefficients for MBs without MC, as well as an error-compensating feedback loop for motion-compensated MBs. The combination of these techniques has been shown to reduce requantization errors incurred during transcoding, and do so with less computational complexity.

Looking to the future of video transcoding, there are still many topics that require further study. One problem is finding an optimal transcoding strategy.

Error-Resilience Transcoding

Transmitting video over wireless channels requires taking into account the conditions in which the video will be transmitted. In general, wireless channels have low bandwidth and higher error rate than wired channels. Error-resilience transcoding for video over wireless channels is needed in this case and has been studied in [38] and [39].

In [38], the authors present a method that is built on three steps. First, they use a transcoder that injects spatial and temporal resilience into an encoded bit stream where the amount of resilience is tailored to the content of the video and the prevailing error conditions, as characterized by bit error rate. The transcoder increases the spatial resilience by reducing the number of blocks per slice and increases the temporal resilience by increasing the proportion of intra-blocks that are transmitted at each frame. Since the bit rate increases due to the error resilience, the transcoder achieves the (same) input bit rate at the output by dropping less significant coefficients as it increases resilience. Second, they derive analytical models that characterize how corruption propagates in a video that is compressed using motion-compensated encoding and subjected to bit errors. Third, they use rate distortion



▲ 9. *Motion vector reestimation. Since Frame (n – 1) is dropped, a new motion vector to predict Frame (n) from Frame (n – 2) is estimated.*

Scalable Coding

For years, scalable video coding schemes have been explored by the video coding community. The holy grail of scalable video coding is to encode the video once and then by simply truncating certain layers or bits from the original stream, lower qualities, spatial resolutions, and/or temporal resolutions could be obtained. Ideally, this scalable representation of the video should be achieved without any impact on the coding efficiency, i.e., the truncated scalable stream (at lower rate, spatial, and/or temporal resolution) should produce the same reconstructed quality as a single-layer bit stream in which the video was coded directly under the same conditions and constraints, notably with the same bit rate.

We begin with an overview of traditional scalable coding schemes, e.g., as defined by MPEG-2 Video [40], where the signal is encoded into a base layer and a few enhancement layers, in which the enhancement layers add spatial, temporal, and/or SNR quality to the reconstructed base layer. Specifically, the enhancement layer in SNR scalability adds refinement data for the DCT coefficients of the base layer. With spatial scalability, the first enhancement layer uses predictions from the base layer without the use of motion vectors. In this case, the layers can have different frame sizes, frame rates, and chrominance formats. In contrast to spatial scalability, the enhancement layer in temporal scalability uses predictions from the base layer using motion vectors, and while the layers must have the same spatial resolution and chrominance formats, they may have different frame rates. The MPEG-2 Video standard supports each of these scalable modes, as well as hybrid scalability, which is the combination of two or more types of scalability.

More recently, a new form of scalability, known as fine granular scalability (FGS), has been developed and adopted by the MPEG-4 Visual standard [41]. In contrast to conventional scalable coding schemes, FGS allows for a much finer scaling of bits in the enhancement layer [42]. This is accomplished through a bit-plane coding method of DCT coefficients in the enhancement layer, which allows the enhancement layer bit stream to be truncated at any point. In this way, the quality of the reconstructed frames is rather proportional to the number of enhancement bits received. The standard itself does not specify how the rate allocation, or equivalently, the truncation of bits on a per frame basis, is done, it only specifies how a truncated bit stream is decoded. In [43] and [44], optimal rate allocation strategies that essentially truncate the FGS enhancement layer have been proposed. Another variation of the FGS scheme, known as FGS-temporal, combines the FGS techniques with temporal scalability [45].

Although the primary focus of this article is on video, it is worthwhile to mention the current state-of-the art in scalable image coding, namely the JPEG-2000 standard [46], [47]. This coder also employs bit-plane coding techniques to achieve scalability, where both SNR and spatial scalability are supported. In contrast to existing scalable video coding schemes that typically rely on a nonscalable base layer and which are based on the DCT, this coder does not rely on separate base and enhancement layers and is based on the discrete wavelet transform (DWT). The coding scheme employed by JPEG-2000 is often referred to as an embedded coding scheme since the bits that correspond to the various qualities and spatial resolutions can be organized into the bit stream syntax in a manner that allows the progressive reconstruction of images and arbitrary truncation at any point in the stream.

To make a comparison between scalable coding and transcoding is rather complex since they address the same problem from different points of view. Scalable coding specifies the data format at the encoding stage independently of the transmission requirements, while transcoding converts the existing data format to meet the current transmission requirements. Although scalable coding can provide low-cost flexibility to meet the target bit rate, spatial resolution, and temporal resolution, traditional schemes sacrifice the coding efficiency compared to single-layered coding. Considering a cascaded transcoding architecture that fully decodes and reencodes the video according to the new requirement, its coding performance will always be better than traditional scalable coding; this has been shown in at least one study [6]. Certainly, more study on this topic is needed that accounts for the latest scalable coding schemes, as well as a wider range of test conditions and test sequences. Also, when it comes to comparing coding efficiency, metrics and procedures are needed to objectively compare the results at various spatio-temporal resolutions, and end-to-end distortion measures under realistic network conditions must also be considered. Steps in this direction are now being made within the MPEG community [48], [49], and recent advances in video coding are showing that the possibility for an efficient universally scalable coding scheme is within reach; e.g., see [50] and [51].

In addition to the issue of coding efficiency, which is likely to be solved soon, scalable coding will need to define the application space that it could occupy. For instance, content providers for high-quality mainstream applications, such as DTV and DVD, have already adopted single-layer MPEG-2 Video coding as the default format, hence a large number of MPEG-2 coded video content already exists. To access these existing MPEG-2 video contents from various devices with varying terminal and network capabilities, transcoding is needed. For this reason, research on video transcoding of single-layer streams has flourished and is not likely to go away anytime soon. However, in the short term, scalable coding may satisfy a wide range of video applications outside this space, and in the long term, we should not dismiss the fact that a scalable coding format could replace existing coding formats. For now, scalable coding and transcoding should not be viewed as opposing or competing technologies. Instead, they are technologies that meet different needs in a given application space and it is likely that they can happily coexist.

theory to compute the optimal allocation of bit rate between spatial resilience, temporal resilience, and source rate. Furthermore, they use the analytical models to generate the resilience rate-distortion functions that are used to compute the optimal resilience. The transcoder then injects this optimal resilience into the bit stream. Simulation results show that using a transcoder to optimally adjust the resilience improves video quality in the presence of errors while maintaining the same input bit rate.

In [39], the authors propose error-resilience video transcoding for internetwork communications using a general packet radio services (GPRS) mobile access network. The error-resilience transcoding takes place in a proxy, which provides the necessary output rate with the required amount of robustness. Here we use two error-resilience coding schemes: adaptive intra refresh (AIR) and feedback control signaling (FCS). The schemes can work independently or combined. Since both AIR and FCS increase the bit rate, a simple bit-rate regulation mechanism is needed that adapts the quantization parameters accordingly. The system uses two primary control feedback mechanisms. First, feedback signals that contain information related to the output channel conditions, such as bit error rate, delay, lost/received packets, etc. Based on the received feedback, AIR and/or FCS can be used to insert the necessary robustness to the transcoded data. For example, in the case of increased bit error conditions, AIR is used as the major resilience block to stop the potential error accumulation effects resulting from transmission errors, e.g., high motion areas are transcoded to intracoded MBs which don't require MC. The second control feedback mechanism comprises adaptive rate transcoding. This requires a feedback signaling method for the control of the output bit rate from the video transcoder. In this way, the signaling is originated from the output video frame buffer within the network-monitoring module, which continuously monitors the flow conditions. In case of underflow, a signal is returned to the transcoder for an increase in bit rate. In case of overflow, the signal indicates to the transcoder that it should decrease the bit rate. This is a relatively straightforward rate-controlling scheme for a congestion control. Experiments showed superior transcoding performances over the error-prone GPRS channels to the nonresilient video.

Concluding Remarks

There are additional video transcoding schemes that have been developed and proposed, but have not been covered here. Included are object-based transcoding [48], transcoding between scalable and single-layer video [53]-[55], and various types of format conversions [56]-[60]. Joint transcoding of multiple streams has been presented in [61] and system layer filtering has been described in [62]. Finally, transcoding techniques that facilitate trick-play modes, e.g., fast forward and reverse playback, have been discussed in [63] and [64].

With the emergence of mobile multimedia-capable devices and the desire for users to access video originally captured in a high resolution, there is a strong need for efficient ways to reduce the spatial resolution of video for delivery to such devices.

Looking to the future of video transcoding, there are still many topics that require further study. One problem is finding an optimal transcoding strategy. Given several transcoding operations that would satisfy given constraints, a means for deciding the best one in a dynamic way has yet to be determined. Work to construct utility functions that gauge a user's satisfaction of a coded video bit stream was introduced in [65]. In this work, features are first extracted from the video, then machine learning and classification techniques are used to estimate the subjective/objective quality of the video coded according to the transcoding operation. Extensions to this work have been considered in [66]. Another approach that uses tables to define the relationship between quality, coding parameters, and constraints has been proposed in [67]. From a somewhat different perspective, initial work on modeling the mean-squared error yielded by various transcoding operations has been presented in [68]. Overall, further study is needed toward a complete algorithm that can measure and compare quality across spatio-temporal scales, possibly taking into account subjective factors, and account for a wide range of potential constraints (e.g., terminal, network, and user characteristics). Another topic is the transcoding of encrypted bit streams. The problems associated with the transcoding of encrypted bit streams include breaches in security by decrypting and reencrypting within the network, as well as computational issues. These problems have been circumvented in [69] with a secure scalable streaming format that combines scalable coding techniques with a progressive encryption technique. However, handling this for non-scalable video and streams encrypted with traditional encryption techniques is still an open issue.

Acknowledgments

The authors thank Prof. Fernando Pereira and Dr. Jeongnam Youn for their time spent reviewing this manuscript.

Anthony Vetro is with Mitsubishi Electric Research Labs in Murray Hill, New Jersey, where he is currently a senior principal member of the Technical Staff. He received the

Ph.D. degree in electrical engineering from Polytechnic University, Brooklyn, New York, and his main research interests are in the areas of video coding and transmission, with an emphasis on content scaling and rate allocation. He has published a number of papers in these areas and has been an active participant in MPEG standards for several years, where he is now serving as editor for MPEG-21 Part 7, Digital Item Adaptation.

Charilaos Christopoulos received the B.Sc. in physics from the University of Patras in 1989; the M.Sc. in software engineering from the University of Liverpool, UK, in 1991; and the Ph.D. in video coding from the University of Patras in 1996. From 1993 to 1995 he was a research fellow at the Free University of Brussels. He joined Ericsson Research in 1995, where he is now manager of Ericsson's MediaLab. He has been actively involved in JPEG2000, MPEG-7, and MPEG-21 standardization activities, serving as head of the Swedish delegation in ISO/SC29/ WG01 (JPEG/JBIG), editor of the JPEG 2000 Verification Model, and co-editor of the JPEG 2000 standard. He holds 15 Swedish filed/granted patents in the field of image and video processing, and he is author/co-author of about 40 journal and conference publications. His research interests include image and video processing, mobile communications, and three-dimensional and virtual/ augmented reality.

Huifang Sun received the Ph.D. degree in electrical engineering from University of Ottawa, Canada. In 1995, he joined Mitsubishi Electric Research Laboratories, where he now serves as deputy director of the Murray Hill Lab. Prior to joining Mitsubishi, he was with the Electrical Engineering Department of Fairleigh Dickinson University and later at Sarnoff Corporation, where he received the AD-HDTV Team Award in 1992 and Technical Achievement Award for optimization and specification of the Grand Alliance HDTV video compression algorithm in 1994. His research interests include digital video/image compression and digital communication. He has published more than 120 journal and conference papers and holds 18 U.S. patents. He is an IEEE Fellow.

References

[1] Y. Nakajima, H. Hori, and T. Kanoh, "Rate conversion of MPEG coded video by requantization process," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, 1995, pp. 408-411.

[2] H. Sun, W. Kwok, and J. Zdepski, "Architectures for MPEG compressed bitstream scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 191-199, Apr. 1996.

[3] A. Eleftheriadis, "Dynamic rate shaping of compressed digital video," Ph.D. dissertation, Dept. Elec. Eng., Columbia Univ., New York, June 1995.

[4] P. Assunçno and M. Ghanbari, "Post-processing of MPEG-2 coded video for transmission at lower bit-rates," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Atlanta, GA, 1996, pp. 1998-2001.

[5] G. Kessman, R. Hellinghuizen, F. Hoeksma, and G. Heidman, "Transcoding of MPEG bitstreams," *Signal Processing: Image Communications*, vol. 8, no. 6, pp. 481-500, Sept. 1996.

[6] N. Bjork and C. Christopoulos, "Transcoder architectures for video coding," *IEEE Trans. Consumer Electron.*, vol. 44, pp. 88-98, Feb. 1998.

[7] N. Björk and C. Christopoulos, "Transcoder architectures for video coding," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing,* Seattle, WA, May 1998, pp. 2813-2816.

[8] J. Youn, J. Xin, and M.T. Sun, "Fast video transcoding architectures for networked multimedia," in *Proc. IEEE Int. Symp. Circuits and Systems*, Geneva, Switzerland, vol. 4, May 2000, pp. 25-28.

[9] S.F. Chang and D.G. Messerschmidt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1-11, Jan. 1995.

[10] H. Sun, A. Vetro, J. Bao, and T. Poon, "A new approach for memory-efficient ATV decoding," *IEEE Trans. Consumer Electron.*, vol. 43, pp. 517-525, Aug. 1997.

[11] J. Wang and S. Yu, "Dynamic rate scaling of coded digital video for IVOD applications," *IEEE Trans. Consumer Electron.*, vol. 44, pp. 743-749, Aug. 1998.

[12] P. Assunção and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 953-967, Dec. 1998.

[13] N. Merhav, "Multiplication-free approximate algorithms for compressed-domain linear operations on images," *IEEE Trans. Image Processing*, vol. 8, pp. 247-254, Feb. 1999.

[14] C.W. Lin and Y.R. Lee, "Fast algorithms for DCT-domain video transcoding," in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, vol. 1, Sept. 2001, pp. 421-424.

[15] J. Youn, M.T. Sun, and C.W. Lin, "Motion vector refinement for high performance transcoding," *IEEE Trans. Multimedia*, vol. 1, pp. 30-40, Mar. 1999.

[16] M. Yong, Q.F. Zhu, and V. Eyuboglu, "VBR transport of CBR-encoded video over ATM networks," in *Proc. 6th Int. Workshop Packet Video*, Portland, OR, Sept. 1994.

[17] B. Shen, I.K. Sethi, and B. Vasudev, "Adaptive motion vector re-sampling for compressed video downscaling," in *Proc. IEEE Int. Conf. Image Processing*, Santa Barbara, CA, vol. 1, Oct. 1997, pp. 771-774.

[18] W. Zhu, K.H. Yang, and M.J. Beacken, "CIF-to-QCIF video bitstream down-conversion in the DCT domain," *Bell Labs Tech. J.*, vol. 3, no. 3, pp. 21-29, July-Sept. 1998.

[19] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Trans. Multimedia*, vol. 2, pp. 101-110, June 2000.

[20] T. Shanableh and M. Ghanbari, "Transcoding architectures for DCT-domain heterogeneous video transcoding," in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, vol. 1, Sept. 2001, pp. 433-436.

[21] P. Yin, A. Vetro, B. Lui, and H. Sun, "Drift compensation for reduced spatial resolution transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 1009-1020, Nov. 2002.

[22] J. Xin, M.-T. Sun, K. Chun, and B.S. Choi, "Motion re-estimation for HDTV to SDTV transcoding," in *Proc. IEEE Symp. Circuits and Systems*, Scottsdale, AZ, vol. 4, May 2002, pp. 715-718.

[23] B.C. Song, T.H. Kim, and K.W. Chun, "Efficient video transcoding with scan format conversion," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, NY, vol. 1, Sept. 2002, pp. 709-712.

[24] P. Yin, M. Wu, and B. Lui, "Video transcoding by reducing spatial resolution," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, BC, Canada, vol. 1, Oct. 2000, 972-975.

[25] *Coding of Audio-Visual Objects—Part 2: Visual, 2nd Ed.*, ISO/IEC 14496-2:2001, 2001.

[26] *Video Coding for Low Bit-Rate Communications*, ITU-T Recommendation H.263+, 1998.

[27] S.J. Wee, J.G. Apostolopoulos, and N. Feamster, "Field-to-frame transcoding with spatial and temporal downsampling," in *Proc. IEEE Int. Conf. Image Processing*, Kobe, Japan, Oct. 1999 [CD-ROM].

[28] A.N. Skodras and C. Christopoulos, "Down-sampling of compressed images in the DCT domain," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Rhodes, Greece, Sept. 1998, pp. 1713-1716.

[29] A. Vetro, H. Sun, P. DaGraca, and T. Poon, "Minimum drift architectures for three-layer scalable DTV decoding," *IEEE Trans. Consumer Electron.*, vol. 44, pp. 527-536, Aug. 1998.

[30] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Select Areas Commun.*, vol. 18, pp. 1012-1032, June 2000.

[31] A. Vetro, T. Hata, N. Kuwahara, H. Kalva, and S. Sekiguchi, "Complexity-quality evaluation of transcoding architectures for reduced spatial resolution," *IEEE Trans. Consumer Electron.*, vol. 48, pp. 515-521, Aug. 2002.

[32] A. Lan and J.N. Hwang, "Context dependent reference frame placement for MPEG video coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Munich, Germany, Apr. 1997, pp. 2997-3000.

[33] J.N. Hwang, T.D. Wu, and C.W. Lin, "Dynamic frame-skipping in video transcoding," in *Proc. IEEE Workshop Multimedia Signal Processing*, Redondo Beach, CA, Dec. 1998, pp. 616-621.

[34] K.T. Fung, Y.L. Chan, and W.C. Siu, "New architecture for dynamic frame-skipping transcoder," *IEEE Trans. Image Processing*, vol. 11, pp. 886-900, Aug. 2002.

[35] A. Vetro, P. Yin, B. Liu, and H. Sun, "Reduced spatio-temporal transcoding using an intra-refresh technique," in *Proc. IEEE Int. Symp. Circuits and Systems*, Scottsdale, AZ, vol. 4, May 2002, pp. 723-726.

[36] M.J. Chen, M.C. Chu, and C.W. Pan, "Efficient motion estimation algorithm for reduced frame-rate video transcoder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 269-275, Apr. 2002.

[37] J. Xin, M.T. Sun, and K. Chun, "Motion-re-estimation for MPEG-2 to MPEG-4 simple profile transcoding," in *Proc. Int. Workshop Packet Video*, Pittsburgh, PA, Apr. 2002 [CD-ROM].

[38] G. de los Reyes, A.R. Reibman, S.-F. Chang, and J.C.-I. Chuang, "Error-resilience transcoding for video over wireless channels," *IEEE J. Select Areas Commun.*, vol. 18, pp. 1063-1074, June 2000.

[39] S. Dogan, A. Cellatoglu, M. Uyguroglu, A.H. Sadka, and A.M. Kondoz, "Error-resilient video transcoding for robust inter-network communications using GPRS," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 453-464, June 2002.

[40] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video, 2nd ed.*, ISO/IEC 13818-2:2000, 2000.

[41] *Coding of Audio-Visual Objects—Part 2 Visual—Amendment 2: Streaming Video Profiles*, ISO/IEC 14496-2:2001/Amd 2:2002, 2002.

[42] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 301–317, Mar. 2001.

[43] Q. Wang, F. Wu, S. Li, Z. Xiong, Y.Q. Zhang, and Y. Zhong, "A new rate allocation scheme for progressive fine granular scalable coding," in *Proc. IEEE Int. Symp. Circuits and Systems*, Sydney, Australia, vol. 2, May 2001, 397-400.

[44] X.M. Zhang, A. Vetro, Y.Q. Shi, and H. Sun, "Constant quality constrained rate allocation for FGS coded video," in *Proc. SPIE Conf. on Visual Communications and Image Processing*, San Jose, CA, Jan. 2002, pp. 817-827.

[45] M. van der Schaar and H. Radha, "A hybrid temporal-SNR fine-granular scalability for internet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 318-331, Mar. 2001.

[46] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Mag.*, vol. 18, pp. 36-58, Sept. 2001.

[47] C. Christopoulos, A.N. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consumer Electron.*, vol. 46, pp. 1103-1127, Nov. 2000.

[48] *The Status of Interframe Wavelet Coding Exploration in MPEG*, ISO/IEC JTC1/SC29/WG11 N4928, July 2002.

[49] *Description of Exploration Experiments in Scalable Video Coding*, ISO/IEC JTC1/SC29/WG11 N5168, Oct. 2002.

[50] *Contributions to Interframe Wavelet and Scalable Video Coding*, ISO/IEC JTC1/SC29/WG11 m9034, Oct. 2002.

[51] *Fully Scalable 3-D Overcomplete Wavelet Video Coding Using Adaptive Motion Compensated Temporal Filtering*, ISO/IEC JTC1/SC29/WG11 m9037, Oct. 2002.

[52] A. Vetro, H. Sun, and Y. Wang, "Object-based transcoding for adaptive video content delivery," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 387-401, Mar. 2001.

[53] Y.C. Lin, C.N. Wang, T. Chiang, A. Vetro, and H. Sun, "Efficient FGS-to-single layer transcoding," in *Proc. IEEE Int. Conf. Consumer Electronics*, Los Angeles, CA, June 2002, pp. 134-135.

[54] Y.P. Tan and Y.Q. Liang, "Methods and needs for transcoding MPEG-4 fine granularity scalability video," in *Proc. IEEE Int. Symp. Circuits and Systems*, Scottsdale, AZ, vol. 4, May 2002, pp. 719-722.

[55] E. Barrau, "MPEG video transcoding to a fine-granular scalable format," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, NY, vol. 1, Sept. 2002, pp. 717-720.

[56] J.L. Wu, S.J. Huang, Y.M. Huang, C.T. Hsu, and J. Shiu, "An efficient JPEG to MPEG-1 transcoding algorithm," *IEEE Trans. Consumer Electron.*, vol. 42, pp. 447-457, Aug. 1996.

[57] N. Memon and R. Rodilia, "Transcoding GIF images to JPEG-LS," *IEEE Trans. Consumer Electron.*, vol. 43, pp. 423-429, Aug. 1997.

[58] N. Feamster and S. Wee, "An MPEG-2 to H.263 transcoder," in *Proc. SPIE Conf. Voice, Video Data Communications*, Boston, MA, Sept. 1999.

[59] H. Kato, H. Yanagihara, Y. Nakajima, and Y. Hatori, "A fast motion estimation algorithm for DV to MPEG-2 conversion," in *Proc. IEEE Int. Conf. Consumer Electronics*, Los Angeles, CA, June 2002, pp. 140-141.

[60] W. Lin, D. Bushmitch, R. Mudumbai, and Y. Wang, "Design and implementation of a high-quality DV50-MPEG2 software transcoder," in *Proc. IEEE Int. Conf. Consumer Electronics*, Los Angeles, CA, June 2002, pp. 142-143.

[61] H. Sorial, W.E. Lynch, and A. Vincent, "Joint transcoding of multiple MPEG video bitstreams," in *Proc. IEEE Int. Symp. Circuits Systems*, Orlando, FL, May 1999.

[62] S. Gopalakrishnan, D. Reininger, and M. Ott, "Realtime MPEG system stream transcoder for heterogeneous networks," in *Proc. Packet Video Workshop*, New York, Apr. 1999 [CD-ROM].

[63] S. Wee, "Reversing motion vector fields," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, IL, Oct. 1998.

[64] Y.P. Tan, Y.Q. Liang, and J. Yu, "Video transcoding for fast forward/reverse video playback," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, NY, vol. 1, Sept. 2002, pp. 713-716.

[65] R. Liao, P. Bocheck, A. Campbell, and S.F. Chang, "Content-aware network adaptation for MPEG-4," in *Proc. Int. Workshop Network Operating Systems Support Digital Audio Video (NOSSDAV)*, Basking Ridge, NJ, June 1999.

[66] *Description of Utility Function Based Optimum Transcoding*, ISO/IEC JTC1/SC29/WG11 m8319, May 2002.

[67] *NOTABLE: NOrmative TABLE Definition Enabling Terminal QoS*, ISO/IEC JTC1/SC29/WG11 m8310, May 2002.

[68] P. Yin, A. Vetro, and B. Liu, "Rate-distortion models for video transcoding," in *Proc. SPIE Conf. Image Video Communications Processing*, Santa Clara, CA, Jan. 2003.

[69] S. Wee and J. Apostolopoulos, "Secure scalable streaming enabling transcoding without decryption," in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, vol. 1, Sept. 2001, pp. 437-440.