Delft University of Technology

Computer Engineering Laboratory

Mekelweg 4, 2628 CD

Delft, The Netherlands

Web: http://ce.et.tudelft.nl

# Video Coding and Transcoding: A Review

Roya Choupani, Stephan Wong, and Mehmet Tolun

**Abstract**

Video streaming over the Internet has gained popularity during the recent years which is mainly the result of the introduction of video-conferencing and video-telephony. These in turn have made it possible to bring to life many applications such as transmitting video over the Internet and telephone lines, surveillance and monitoring, telemedicine (medical consultation and diagnosis at a distance), and computer based training and education. These applications need a large bandwidth which is not available in all cases. Many video encoding standards have been introduced to deal with video compression and transmission problems. In this study, we have discussed the main technical features of the most important video coding standards in a comparative approach. The appropriateness of these features is application and transmission environment dependant. Manipulating video stream features or video transcoding methods are discussed as well.

**Index Terms**

Video Coding, Video Streaming, Transcoding

## I. Introduction

Digital Video Coding is the fundamental technology for many applications, such as Digital TV Broadcasting, Distance Learning, Video on Demand, Video Telephony and Video Conferencing. The problem of storing and transmitting digital video has been the topic of research for many years[1] [7] [33]. The video coding problem is considered as investigating the methods of efficiently representing a video in binary form and therefore it can be classified as an image compression problem[5]. What is different about video compression is that a video may be considered as a stream of frames taken consecutively at very short intervals and is viewed and understood by some end user(s)[2]. These facts brings about many important features which can be exploited to reduce transmitted data size. First of all, short time interval between consecutive frames means that the contents should be very close to each other and only the difference may be encoded and transmitted. Secondly, since the final user of the video stream is a human, the biological features of human visual system should be considered and the priority in preserving data should be based on these features[6]. A low quality displaying device can not visualize many details of a high quality and high resolution image or video. To separate important data from less important details, special mathematical transforms such as Discrete Cosine Transform ***DCT*** or Discrete Wavelet Transform ***DWT*** are necessary [5]. However, the final codewords generated may experience different repetition frequencies. This statistical redundancy may also be removed by assigning variable length codes to the codewords. A typical video coder therefore, includes:

- Motion Detector

  Motion detector compares the current frame with the last frame in the video stream. The aim here is eliminating temporal redundancy by transmitting the difference between the consecutive frames. As a result a vector describing the motion and the difference with the previous frame after considering the motion is encoded.

- Transformer

  A mathematical transformation is applied to obtain the amount of information content at each frequency band. This type of transformation can make the selective elimination of less important information possible.

- Quantizer

  The output of the mathematical transformer is quantized to reduce the data size. This step however, adds data inaccuracy to some extent.

- Variable Length Encoder

  each quantized output of the transformer is assigned a variable length code depending on its frequency.

A general structure for a video coder/decoder is given in figure 1.

On the other hand, the available communication technologies are extremely heterogeneous. Adapting the media content to the characteristics of different networks in order to obtain video delivery with acceptable quality is thus an important issue [1]. Video transcoding converts one compressed video bitstream into another with a different encoding standard, size, bit rate, or frame rate. The goal of transcoding is to enable exchanging data between heterogeneous multimedia networks and reducing the complexity and transmission time by avoiding the total decoding and re-encoding of a video stream. In this paper we will first consider more common video coding standards. Then we also review the basic video transcoding methods and the main issues in their implementation.

## II. Data Redundancy

Data redundancy is the main hinderance in effective implementation of video based applications especially those requiring a fast or real time and high quality transmission.

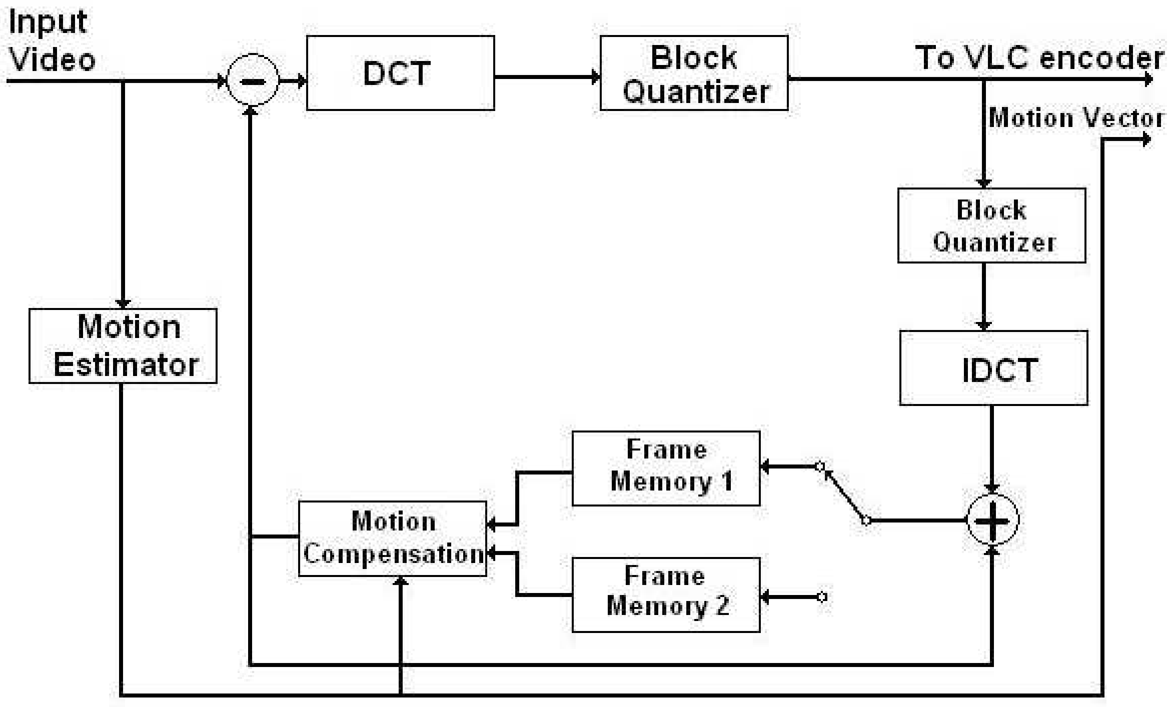


Fig. 1. Typical video encoder/decoder structure

Data redundancy may be in different forms including temporal redundancy, psychological redundancy, spatial redundancy, and statistical redundancy. Each of these redundancies is discussed separately in the following sections.

### *A. Temporal Redundancy*

Since the elapsed time between two consecutive frames in a video stream is generally very short, the consecutive frames are very similar in content and contain a lot of redundancy. In most cases however, the differences between the consecutive frames can be best expressed by considering the displacements of the objects and encoding the motions vectors and differences. This strategy reduces the data size considerably but at the same time it adds the complexity of motion detection in the video. To simplify the procedure the image is divided into small fixed ( H.261, MPEG-1[8]) or variable sized blocks (H.263[12], MPEG4[8], H.264[13]). Motion detection then is performed by finding the best match for the block in a window in the first frame centered at the block's position in the second frame. The criteria for a match is generally either a (mean) sum of squared difference (Mean Squared Error ***MSE***, Sum of Squared Difference ***SSD*** ) or Mean Absolute Error ***MAE***. Equations 1 , 2 and 3 define these criteria formally.

$$MSE(s,t) = \frac{1}{mn}\sum_{j=1}^{m}\sum_{k=1}^{n}(f_n(j,k) - f_{n-1}(j+s,k+t))^2 \tag{1}$$

$$SSD(s,t) = \sum_{j=1}^{m}\sum_{k=1}^{n}(f_n(j,k) - f_{n-1}(j+s,k+t))^2 \tag{2}$$

$$MAE(s,t) = \frac{1}{mn}\sum_{j=1}^{m}\sum_{k=1}^{n}|f_n(j,k) - f_{n-1}(j+s,k+t)| \tag{3}$$

To find the optimum or near optimum match for a block, one of the matching criteria given above should be applied in many different positions. Generally the search region is restricted to $(2d+1)\times(2d+1)$ positions where $d$ is the block side length as a full search is very expensive. Some methods have been introduced to decrease the computation complexity such as 2D logarithmic search [18], three step search [17] and multiresolution block matching [16]. An important problem in block wise matching and segmentation of the video frames is that artifacts are generated at the boundaries of the blocks which are visible by human viewers. A solution to this problem is proposed in the Overlapped Block Matching algorithm[15]. The idea is allowing the the blocks to overlap. Each block is extended in all directions to twice of its original size. Each pixel therefore will be covered by four extended blocks as can be seen in figure 2. The estimation is carried out in the same manner. A window function which decays towards the boundaries of the block is applied for the error prediction. Equation 4 shows the error prediction using mean absolute difference after applying window function

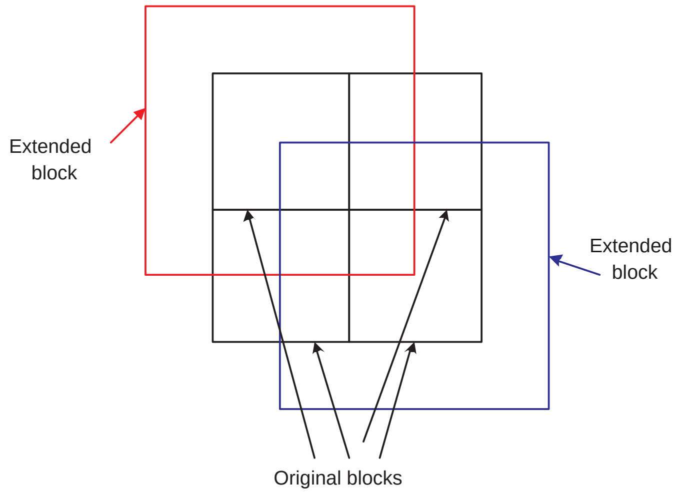


Fig. 2. Overlapped extended blocks

$W$.

$$MAD = \frac{1}{2d} \sum_{x=1}^{m} \sum_{y=1}^{m} |W(x,y)E(x,y)| \quad (4)$$

## *B. Psychological Redundancy*

In most of video streaming applications, the intended recipient is a human being so it is reasonable to consider the capabilities of the human visual system before encoding. A frame is divided into macro blocks which are further divided into three layers each having one of the color components of the pixels. The color model used in almost all video coding standards is YCbCr because of its compatibility with YUV standard used in television systems and its ability to separate intensity from chroma. The macro block layer for Y component is a 16x16 block (or 4 8x8 blocks). The Cb and Cr components can be sub-sampled to 8x8 blocks. The point behind the sub-sampling in Cb and Cr components is the fact that the human visual system is more sensitive to illumination changes than changes in color. The format obtained in this way is referred to as 4:2:0. The sub-samplings in Cb and Cr components create a 50% reduction in the data size. A macro block converted into 6 blocks of 8x8 each using 4:2:0 mode is shown in figure 3.

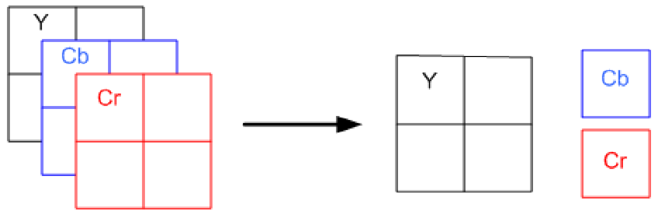


Fig. 3. Macro block after 4:2:0 reduction

### *C. Spatial Redundancy*

Each image frame in a video sequence may contain many details which do not transfer important information and their elimination will not affect the image or video quality. To distinguish unimportant data from important ones a special transform is necessary. Each block of the video frames is tansformed using Discrete Cosine Transform DCT or Discrete Wavelet Transform DWT and then quantized. The quantized blocks of values are converted into 1-D vectors by reading their values in zig-zag order. The main point in these transforms is that the high frequency values having a small energy content can be safely ignored. Also the quality of the image will be directly related to this elimination which means a trade off between quality and compression ratio can be achieved considering the specifications of the available transmission medium.

### *D. Statistical Redundancy*

The quantized values are then compressed by Run Length Encoding (RLE) and the resulted values are coded using a Huffman encoder. The zig-zag order of reading blocks gives the most compressed vectors after RLE since most of the non-zero values after quantization are concentrated in upper left corner. The value at left upper left corner (DC value) is generally not coded. AC components are converted into (skip, value) format for RLE encoding, then the composite symbols (skip,value) are encoded using Huffman coding. Huffman Tables can be custom (sent in header) or default.

*E. Other Considerations*

In temporal redundancy section it was mentioned that the difference between the current image and the previous one may be encoded instead of the entire frame. This however may result in accumulation of error if a frame experience corruption while transmission. A second issue is the problem of freely navigation in the video stream. An encoding based on the differences only creates a serially accessible data which means to view a frame all previous frames should be read and decompressed. The solution to these problems is categorizing the frames as I-frames, P-frames and B-frames. An I-frame is an independent frame which is encoded without considering the previous frames. A P-frame is a uni-directional difference frame which is obtained from the previous I or P frame. A B-frame encodes the difference between its previous and next P or I frames. A B-frame includes the motions vectors in both forward and backward directions and makes navigation in both directions possible. Figure 4 shows the sequence of I, P and B frames in a video stream. Audiovisual scenes in MPEG-4

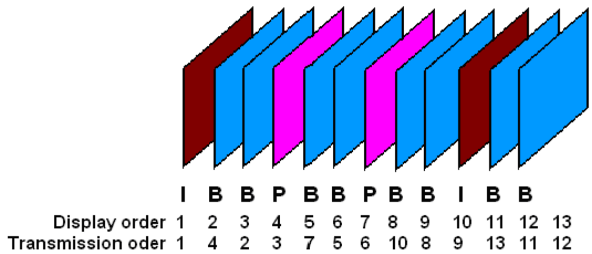


Fig. 4. The sequence of I, P and B frames

are composed of audiovisual objects (AVO) which are organized in a hierarchical fashion[4]. Primitive AVOs such as a fixed background, image of a person, a telephone set, the ringing sound associated with the phone, are found at the lowest level of the hierarchy. Figure 5 shows an image with several primitives. Audiovisual objects are combined in a hierarchical structure to define audiovisual scenes. The tree like structure created in this way can be

Fig. 5. Sample Audio Visual Objects

changed through user interaction. [3].

### *F. Recent Improvements: H264/AVC*

As an improved standard, H.264/AVC incorporates a set of new coding features to achieve a coding efficiency higher than that of all other existing standards [13]. A summary of the features that are used in the proposed transcoding method are as follows.

1. Multiple macroblock partitions for motion-compensated prediction

   The luminance component of each 16×16 macroblock can be partitioned into 16×16, 16×8, 8×16 , and 8×8 samples, and when necessary, each 8×8 block of samples can be further partitioned into 8×4, 4×8, and 4×4 samples, resulting in a combination of seven motion-compensated prediction (MCP) modes. Consequently, up to sixteen motion vectors can be specified for prediction of an inter-coded macroblock. The large variety of partitions allow more accurate inter-frame prediction in regions with fine motion and spatial details, yielding less amount of prediction residue to code.

2. Motion vectors with quarter-pixel accuracy

   The motion vectors are specified in quarter-pixel accuracy, enabling more precise motion compensation in areas with fine or complex motion. To perform compensation at sub-pixel accuracy, the frame values at half- or quarter-pixel positions are interpolated from values at nearby integer-pixel positions.

3. Multiple frame prediction

   Up to five previously coded frames can be used as references for inter-frame macroblock prediction. This is useful when an object or scene is absent from the immediately preceding reference frame but present in other previous reference frames.

## III. Transcoding

Despite the fact that a video stream is compressed by eliminating all redundancies, many network channels may not have the necessary capabilities to handle these streams. For example, in a wireless network, which normally has less than 20 Kb/s bandwidth, the quality degradation due to the low bitrate is significant at 25 or 30 frames/s). The solution used here is reducing data size by changing video format. This change can be in bits per pixel, pixels per frame, frame per second, content of the video, or coding standard. For instance to transport video over low bandwidth channels, such as the public switched telephone network (PSTN) or wireless network, a high transcoding ratio is required. However, the high transcoding ratio may result in unacceptable picture quality when the video is transcoded with the full frame-rate or full spatial resolution. Frame-rate reduction is often used as an efficient scheme to allocate more bits to the remaining frames, so that acceptable quality for each frame can be maintained. In addition, the frame-rate conversion is also needed when an end-system supports only a lower frame-rate. In this section, we discuss motion vector refinement for transcoding involving the frame-rate conversion. A common solution to this problem is adapting the video streams to the channel properties. This adaptation which is referred to as video transcoding converts a video stream from one format into another[14]. This conversion may involve bit rate (bits per pixel), resolution (spatial transcoding), frame rate (temporal transcoding) or encoding standard[31]. All these format conversions are readily achievable if an encoded video is decoded and then re-encoded using the new requirements. However, a large fraction of the processing time is used in motion estimation and data transformation. This means that the procedure should be carried out in compressed data to be relevant in real time applications. In the following sections video transcoding methods are discussed in compressed streams.

### *A. Bit Rate Transcoding*

Bit rate transcoding reduces the number of bits transmitted for the total video without reducing the spatial resolution or number of frames per second[20]. This reduction is achieved by reducing the quality of the video at each frame. The main idea being reduction of lengthy and intensive computations, DCT/IDCT and motion detection are avoided as far as possible[11]. The main method which is also referred to as open loop transcoding discards the high frequency components of DCT transform and re-quantizes the coefficients. The method however, suffers from the drift problem. As mentioned in the above sections, a P frame is obtained by encoding the difference of the current frame and the previous P or I frame. Discarding some of the components and requantizing the coefficients changes the base image with which the current image has been compared and the decoder will estimate the frame with an error which accumulates till the next I frame[19]. The drift problem as described above is the major problem in all bit rate transcoding algorithms. A proposed solution to the drift problem is repeating IDCT/DCT computations with discarding high frequency components and re-quantization but using the same motion vectors as available in the original stream. The main disadvantage of this method is the intensive computation of IDCT/DCT for the total stream.

*B. Spatial Transcoding*

Many end users in a heterogeneous network may have a lower displaying capability than the transmitted video stream. This together with the bandwidth restriction demands a transcoding from a higher resolution to a lower one by down-sampling[29]. The most intuitive way to perform down-sampling in the DCT domain is to only retain the low-frequency coefficients of each block and recompose the new motion block using the compositing techniques proposed in [33]. Specifically, for conversion by a factor of 2, only the $4 \times 4$ DCT coefficients of each $8 \times 8$ block in a macro block are retained. These low frequency coefficients from each block are then used to form the output macro block. 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 [30] and [31] may also be considered. Due to the high computational cost of the motion estimation, in the down-scaled frames motion estimation is carried out using the motion vectors from the original data. The challenging problem here is finding the new motion vectors[22]. The simplest case in down-scaling however, is a 2:1 ratio in which four macro blocks are converted to only one macro block. This case has been shown in figure 6. To estimate the motion vector different ways of combining the original vectors have been used. Among these methods, median, mean and random selection are more common. The more complicated case of arbitrary ratios is obtained by considering unequal contribution for each motion vector. In [27] a resizing ratio R, which is defined as the ratio of original dimensionality to the desired dimensionality is considered. For each macro block in the resized frame, a supporting area as shown in figure 7 is defined. A weighted average scheme has been used to find out the predicted motion vector by using all motion vectors related to the underlying macro blocks in the original reference frame. Equation 5 shows how the weighted average of the motion

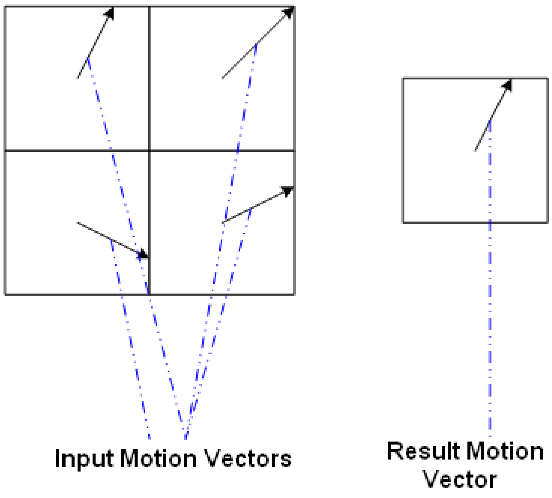


Fig. 6. Downsampling in 2:1 ratio.

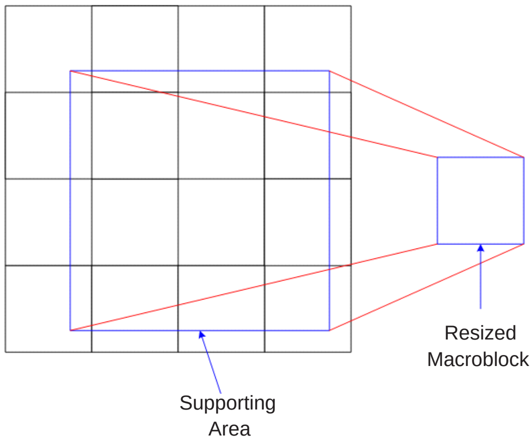


Fig. 7. Definition of a supporting area.

vectors is computed.

$$mv = \frac{1}{R} \frac{\sum_{i=1}^{N} mv_i A_i}{\sum_{i=1}^{N} A_i} \tag{5}$$

where N is the number of macro blocks in the supporting area, R is the resizing ratio, and $A_i$ is the overlapping area of macro block i with the supporting area. Another scheme for reducing drift error effect is given in [26]. In this scheme, output macro blocks are subject to a DCT-domain down-sampling, requantization, and variable-length coding. Output macro blocks 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 intra-blocks; there is no picture drift associated with these blocks. The decision to code an intra-block from the frame store depends on the macro block coding modes and picture statistics. In the first case, based on the coding mode, an output macro block 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 intra-coded block in its place. Of course, the increase in the number of intra-blocks 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 intra-blocks usually require more bits to code.

### *C. Temporal Transcoding*

In temporal transcoding one or a few frames are skipped between any two frames of a stream[21]. Similar to spatial transcoding a drift problem is the main source of the quality degradation. Also since each frame is compared to the previous frame and the difference is encoded and transmitted, removing a frame leaves the next frame with no base to compare and accumulates the error in the following frames too. As the main time consuming part of the encoding algorithms is motion compensation and DCT/IDCT transforms, main challenges have been focussed on performing the transcoding in DCT domain without recomputing the motion vectors. Some authors have proposed a frame control scheme to dynamically adjust the number of skipped frames according to the accumulated magnitude of motion vectors such that the transcoded sequence can present a much smoother motion [28]. Two different cases have been considered here. Case one considers dropping a frame when the motion vector of the blocks in the next frame is zero. This case is illustrated in figure 8. The motion

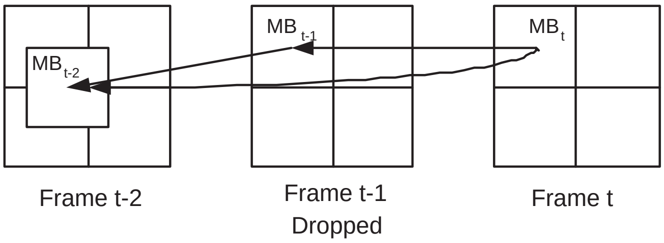


Fig. 8. Dropping frame with zero Motion Vector.

vector of the macro block in the next frame will be updated by replacing it with the motion vector of the dropped frame's macro block. The DCT coefficients are obtained by adding the prediction error coefficients of the dropped macro block and the next frame's macro block. Re-quantization is then carried out using the sum of prediction error DCT coefficients as shown in equations 6 and 7.

Assuming $MB_i = MB_{i-1} + \epsilon_i$ where $MB_i$ is the $i_{th}$ macro block and $\epsilon_i$ is the difference between $MB_i$ and $MB_{i-1}$ we have:

$$DCT(\hat{\epsilon}_t) = DCT(\epsilon_t) + DCT(\epsilon_{t-1}) \quad (6)$$

$$Q[DCT(\hat{\epsilon}_t)] = Q[DCT(\epsilon_t) + DCT(\epsilon_{t-1})] \quad (7)$$

where Q[.] shows the quantization process. Case two is when the motion vectors are not zero. In this case IDCT/DCT is performed for the macro blocks and the motion vectors are obtained from interpolating the original ones. Also as described in [23], [24], and [25], 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 [23] and [32] 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 [25], 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 [24], [25].

### *D. Standard Transcoding*

Standard transcoding considers the conversion from one encoding standard to the other. The challenges here are the features available in the source standard which are not supported by the destination standard. Some of these problems are summarized below. First group of problems is related to the quality of the properties supported by the standards. The solution for these problems are the same as the solutions for other transcodings. As an example, some of the standards such as MPEG-4-SP support a low resolution for video frames. A transcoding from MPEG-2 with frame resolution of $720 \times 480$ to MPEG-4-SP with a resolution of $176 \times 144$ needs a reduction of resolution and is converted to a spatial transcoding problem[8]. The second group of problems are related to representation or optimization features. MPEG-4 for instance supports a Global Motion Compensation (GMC) vector which optimizes the encoding stage when the video includes a global motion. This feature is not supported by lower versions of MPEG such as MPEG-1 and MPEG-2. MPEG-4-SP also does not support **B** frames so a transcoding into this standard needs the conversion of **B** frames into **P** frames.

### *E. Conclusion*

In this paper we reviewed the basic challenges and methods in video streaming. Most of the video coders use data compression methods for reducing the size of the transmitted data especially for real time streaming applications. This makes video processing in compressed space a necessity. On the other hand heterogeneity of the underlying technology and improvements of the video coding algorithms need transcoding from one set of video properties to another. Format incompatibility also can be resolved using a transcoder. The challenge here is performing the transcoding in real time and compressed space.

## References


[1] R. Mohan, J.R. Smith, S. Li, "Adapting multimedia Internet content for universal access", IEEE Transaction on Multimedia Vol 1, no 1, pp 104–114, 1999

[2] Y. Wang, S.F. Chang, A. C. Loui, "Subjective Preference of Spatio- Temporal Rate in Video Adaptation Using Multi-Dimensional Scalable Coding," Proceedings of IEEE International Conference on Multimedia and Expo, Taipei, Taiwan, June 2004.

[3] M. A. Smith and T. Kanade, "Video Skimming for Quick Browsing Based on Audio and Image Characterization", Carnegie Mellon University, Technical Report, July 1995.

[4] H. Sundaram, L. Xie, S.F. Chang, "A Utility Framework for the Automatic Generation of Audio-Visual Skims", Proceedings of ACM Multimedia, Juan Les Pins, France, December 2002.

[5] E.C. Reed and J.S. Lim, "Optimal multidimensional bit-rate control for video communications", IEEE Transaction on Image Processing, vol. 11, no. 8, pp. 873–885, August 2002.

[6] J. Lee and B. W. Dickinson, "Temporally adaptive motion interpolation exploiting temporal masking in visual perception", IEEE Transactions on Image Processing, vol. 3, pp. 513-526, September 1994

[7] F. Pereira and I. Burnett, "Universal Multimedia Experiences for Tomorrow", IEEE Signal Processing Magazine, vol. 20, pp. 63–73, March 2003.

[8] Public MPEG documents, $http://www.chiariglione.org/mpeg/working_documents.htm$

[9] S. F. Chang and A. Vetro, "Video adaptation: Concepts, technologies, and open issues", Proceedings of IEEE, vol. 93, no. 1, pp. 148-158, January 2005.

[10] A. Vetro, H. Sun, and Y. Wang, "Object-based transcoding for adaptable video content delivery", IEEE Transaction on Circuits and Systems for Video Technology", vol. 11, no. 3, pp. 387-401, March 2001.

[11] Y. P. Tan, Y. Q. Liang, and J. Yu, "Video transcoding for fast forward/ reverse video playback", Proceedings of IEEE International Conference on Image Processing, vol. 1, 2002, pp. 713-716.

[12] "Video coding for low bit rate communication", International Telecommunication Union-Telecommunication (ITU-T), Geneva, Switzerland, Recommendation H.263, 1998.

[13] "Draft text of final draft international standard for advanced video coding", International Telecommunication Union-Telecommunication (ITU-T), Geneva, Switzerland, Recommendation H.264 (draft), March 2003.

[14] A. Vetro, C. Christopulos, and H. Sun, "Video transcoding architectures and techniques: An overview", IEEE Signal Processing Magazine, vol. 20, no. 2, pp. 18-29, March 2003.

[15] S. Nogaki, M. Ohta, "An overlapped block motion compensation for high quality motion picture coding", Proceedings of IEEE International Symposium on Circuits and Systems vol. 1, pp. 184–187, 1992

[16] D. Tzovaras, M. Strintzis, H. Sahinolou, "Evaluation of multiresolution block matching tchniques for motion and display estimation", Signal Processing and Image Communication, 6, pp. 56–67, 1994

[17] T. Koga, K. Linuma, A. Hirano, Y. Iijima, T. Ishiguro, "Motion compensated interfrae coding for video conferencing", Proceedings of NTC'81, G5.3.1–G5.3.5, New Orleans, 1981

[18] J.R. Jain, A.K. Jain, "Displacement measurement and its application in interframe image coding", IEEE Transaction on Communication, COM-23(12), pp. 1799–1808, 1981

[19] Y. Nakajima, H. Hori, and T. Kanoh, "Rate conversion of MPEG coded video by re-quantization process", in Proceedings of IEEE International Conference on Image Processing, vol. 3, pp. 408-411, 1995

[20] P. A. A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bitrate reduction of MPEG-2 bit streams", IEEE Transaction on Circuits and Systems for Video Technology, vol. 8, no. 8, pp. 953-967, December 1998.

[21] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatial-temporal resolutions and different encoding formats", IEEE Transaction on Multimedia, vol. 2, no. 2, pp. 101-110, June 2000.

[22] J. Song and B. L. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock", IEEE Transaction on Circuits and Systems for Video Technology, vol. 10, no. 5, pp. 767-775, August 2000.

[23] J. Youn, M.T. Sun, and C.W. Lin, "Motion vector refinement for high performance transcoding", IEEE Transaction on Multimedia, vol. 1, pp. 30-40, March 1999.

[24] A. Lan and J.N. Hwang, "Context dependent reference frame placement for MPEG video coding", in Proceedings of IEEE International Conference on Acoustics, Speech

and Signal Processing, Munich, Germany, pp. 2997-3000, April 1997.

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

[26] P. Yin, A. Vetro, B. Lui, and H. Sun, "Drift compensation for reduced spatial resolution transcoding", IEEE Transaction on Circuits and Systems for Video Technology, vol. 12, pp. 1009-1020, November 2002.

[27] G. Shen, B. Zeng, Y.Q. Zhang, and M.L. Liou, "Transcoder with arbitrarily resizing capability", in Proceedings of IEEE International Symposium on Circuits and Systems, vol. 5, pp. 25-28, 2001.

[28] N. Bjork and C. Christopoulos, "Transcoder architecture for video coding", IEEE Transaction on Consumer Electronics, vol. 44, no. 1, pp. 8898, February 1998.

[29] R. Dugad and N. Ahuja, "A fast scheme for image size change in the compressed domain", IEEE Transaction on Circuit and Systems for Video Technology, vol. 11, no. 4, pp. 461–474, April 2001.

[30] A.N. Skodras and C. Christopoulos, "Down-sampling of compressed images in the DCT domain", in Proceedings of European Signal Processing Conference (EUSIPCO), Rhodes, Greece, pp. 1713-1716, September 1998.

[31] A. Vetro, H. Sun, P. DaGraca, and T. Poon, "Minimum drift architectures for three-layer scalable DTV decoding", IEEE Transaction on Consumer Electronics, vol. 44, pp. 527-536, August 1998.

[32] A. Vetro, P. Yin, B. Liu, and H. Sun, "Reduced spatio-temporal transcoding using an intra-refresh technique", in Proceedings of IEEE International Symposium on Circuits and Systems, Scottsdale, AZ, vol. 4, pp. 723-726, May 2002.

[33] S.F. Chang and D.G. Messerschmidt, Manipulation and compositing of MC-DCT compressed video", IEEE Journal of Selective Areas Communications, vol. 13, pp. 1-11, January 1995.

[34] Lujun Yuan, Feng Wu, Qi Chen, Shipeng Li, Wen Gao,"The fast close-loop video transcoder with limited drifting error", Proceedings of the 2004 International Symposium on Circuits and Systems, Vol. 3, pp. 769-772, May 2004.