

Motion Compensation on DCT Domain

Ut-Va Koc

Lucent Technologies Bell Labs, 600 Mountain Avenue, Murray Hill, NJ 07974, USA
Email: koc@bell-labs.com

K. J. Ray Liu

*Department of Electrical and Computer Engineering and Institute for Systems Research,
University of Maryland, College Park, MD 20742, USA*
Email: kjrlui@eng.umd.edu

Received 21 May 2001 and in revised form 21 September 2001

Alternative fully DCT-based video codec architectures have been proposed in the past to address the shortcomings of the conventional hybrid motion compensated DCT video codec structures traditionally chosen as the basis of implementation of standard-compliant codecs. However, no prior effort has been made to ensure interoperability of these two drastically different architectures so that fully DCT-based video codecs are fully compatible with the existing video coding standards. In this paper, we establish the criteria for matching conventional codecs with fully DCT-based codecs. We find that the key to this interoperability lies in the heart of the implementation of motion compensation modules performed in the spatial and transform domains at both the encoder and the decoder. Specifically, if the spatial-domain motion compensation is compatible with the transform-domain motion compensation, then the states in both the coder and the decoder will keep track of each other even after a long series of P-frames. Otherwise, the states will diverge in proportion to the number of P-frames between two I-frames. This sets an important criterion for the development of any DCT-based motion compensation schemes. We also discuss and develop some DCT-based motion compensation schemes as important building blocks of fully DCT-based codecs. For the case of subpixel motion compensation, DCT-based approaches allow more accurate interpolation without any increase in computation. Furthermore, a sparse number of DCT coefficients after quantization significantly decreases the number of calculations required for motion compensation. Coupled with the DCT-based motion estimation algorithms, it is possible to realize fully DCT-based codecs to overcome the disadvantages of conventional hybrid codecs.

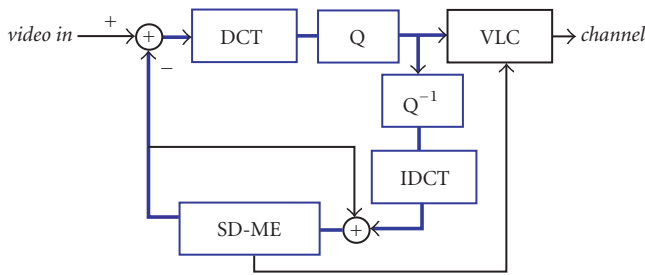
Keywords and phrases: video codec, video coding, MPEG compatibility, motion compensation, DCT.

1. INTRODUCTION

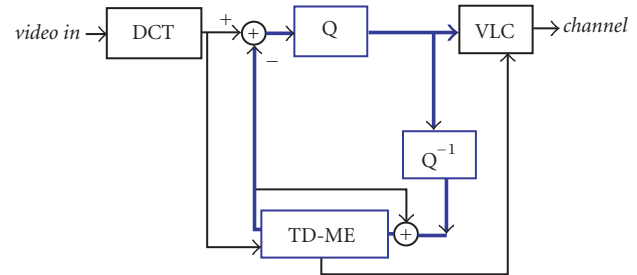
In most international video coding standards such as CCITT H.261 [1], MPEG-1 [2], MPEG-2 [3] as well as the proposed HDTV standard, Discrete Cosine Transform (DCT) and block-based motion estimation are the essential elements to achieve spatial and temporal compression, respectively. Most implementations of a standard-compliant codec adopt the conventional motion-compensated DCT video coder and decoder structures as shown in Figures 1a and 1b, respectively. The feedback loop in the coder for temporal prediction consists of a DCT, an Inverse DCT (IDCT), a spatial-domain motion compensator (SD-MC), and a spatial-domain motion estimator (SD-ME) which is usually the full search block matching approach (BKM). This is undesirable. In addition to the additional complexity added to the overall architecture, this feedback loop limits the throughput of the coder and becomes the bottleneck of a real-time high-end video codec.

A compromise is to remove the loop and perform open-loop motion estimation based upon original images instead of reconstructed images in sacrifice of the performance of the coder [4, 5]. The presence of the IDCT block inside the feedback loop of the conventional video coder design comes from the fact that currently available motion estimation algorithms can only estimate motion in the spatial domain rather than directly in the DCT domain. In the case of the conventional decoder, motion compensation is carried out in the spatial domain after conversion of compressed bit streams by the IDCT block back to the uncompressed reconstructed pixels. This implies the requirement for such a decoder to handle all the image pixels in real time even after being encoded at a very high compression rate.

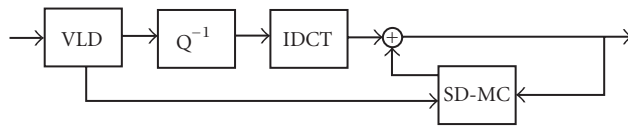
The alternative fully DCT-based video coder and decoder architectures, as shown in Figures 2a and 2b, respectively, have been proposed and explored in [5, 6, 7, 8] among others, to address the shortcomings of the conventional codec



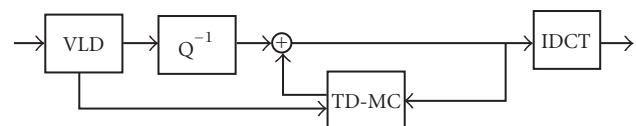
(a) Conventional hybrid motion-compensated DCT video coder.



(a) Fully DCT-based motion-compensated video coder.



(b) Conventional hybrid motion-compensated DCT video decoder.



(b) Fully DCT-based motion-compensated video decoder.

FIGURE 1: Conventional hybrid motion-compensated DCT video codec: motion estimation/compensation are performed in the spatial domain.

structures. In the fully DCT-based video codec, motion estimation and compensation are performed on the compressed bit stream in the transform (DCT) domain, and thus these functional blocks are called transform-domain motion estimator (TD-ME) and transform-domain motion compensator (TD-MC), respectively, to be distinguished from their spatial-domain counterparts. The resultant fully DCT-based motion compensated video codec structure enjoys several advantages over the conventional hybrid motion compensated DCT video codec structure:

- **Less coder components and complexity:** removal of the DCT-IDCT pair in the feedback loop of the fully DCT-based reduces the total number of components required in the feedback loop and thus the complexity of the complete coder.

- **Higher throughput rate:** the feedback loop of a video coder requires processing at the frame rate so that the previous frame data can be stored in the frame memory and need to be available for coding the next incoming frame. Traditionally, this loop has four components plus the spatial-domain motion estimation and compensation unit and thus creates the bottleneck for encoding large frame sizes in real time. In the conventional coder, the whole frame must be processed by both the DCT-IDCT pair and the Q - Q^{-1} pair before the next incoming frame. In the DCT-based structure, the whole frame must be processed by only the Q - Q^{-1} pair. This results in a less stringent requirement on the processing speed of the feedback loop components. Alternatively, this may increase the throughput rate of the coder and thus allow processing larger frame sizes when the technology keeps on improving the processing speed of these components. This high throughput advantage becomes increasingly important when the advances in optical networking

FIGURE 2: Fully DCT-based motion-compensated video codec: motion estimation/compensation are completed in the transform (DCT) domain.

technology permit transmission of high-quality production-grade video signals over broadband networks in real time at affordable costs.

- **Lower computational complexity of DCT-based motion estimation and compensation approaches:** due to the decorrelation of DCT exploited in most video standards, most energy tend to cluster in a few DCT coefficients, especially the DC terms, with the rest being zeros after quantization. This characteristic is particularly beneficial to the DCT-based approach since no computation is needed for the majority of DCT coefficients being zero [9].

- **Joint optimization of DCT-based components:** a fast lattice-structured DCT coder generates dual outputs (DCT and DST) which can be utilized by the DCT-based motion estimation algorithms.

- **Extendibility to a transcoder structure:** an optimal transcoder modifies the encoded video bit stream in the DCT domain directly to fit different usage requirements (such as frame rate conversion, frame size conversion, bit rate conversion, etc.) different from the usage requirement originally planned for. The fully DCT-based structure handles video data completely in the DCT domain and therefore can be easily extended to provide a transcoder function by cascading a DCT-based decoder with certain simplification and modification required by the end usage. For example, the DCT coder at the front of a DCT-based coder and an IDCT decoder of a DCT-based decoder can be removed.

- **Additional information processing:** DCT coefficients carry certain information which can be utilized, as an example, for image segmentation in the DCT domain [10]. The DCT-based codec structure facilitates such use of DCT coefficients.

- Compatibility with existing standards: one of the main objectives of most motion-compensated DCT video compression standards is to ensure proper interoperability between an encoder and a decoder supplied by two different vendors. The fully DCT-based structure encodes the intraframes and motion compensated residuals in DCT *fundamentally* in the same way as the hybrid structure does. The encoded bit stream can be made fully compatible with the existing video coding standards under certain conditions as discussed later in this paper.

For compatibility with existing standards, a decoder must be able to track closely the state of the encoder transmitting the encoded bit stream to ensure interoperability, because of the recursive nature of the motion-compensated standards, especially for systems having lots of interframes to achieve high compression. In this way, the decoder will be able to reproduce the images in as high fidelity as possible. However, despite the above advantages of fully DCT-based codecs, no effort has been spent on the important issue of how the conventional hybrid codecs and the fully DCT-based codecs can match with each other without causing degradation in video quality. In Section 2, we will demonstrate that

- The fully DCT-based codecs and the conventional hybrid codecs are mathematically compatible in the sense that the state of a DCT-based decoder can track the DCT-based encoder state in the same manner as the hybrid counterparts.

- If the coder-encoder pair of different architectural types meets the matching condition, then we can mix different types of coders and decoders without having the decoders reaching divergent states after a long series of P-frames.

Furthermore, we will also discuss DCT-based motion compensation schemes employed in these fully DCT-based codecs in Section 3.

2. MATCHING FULLY DCT-BASED CODECS WITH CONVENTIONAL HYBRID CODECS

In order to gain insight into the main differences between the conventional codec and the fully DCT-based one, we are going to derive the mathematical formulation for both architectures. The conventional hybrid coder is redrawn in Figure 3 with the SD-ME block splitted into two functional blocks (ME for motion estimation and MC for motion compensation) and the frame memory (Z) explicitly depicted for the storage of the previous reconstructed frame. The estimated motion vector field \hat{v}_t for the current frame is estimated from both the current frame x_t and the previous reconstructed frame y_{t-1} . The encoded bit stream for the compressed interframe residual is denoted as $\text{VLC}\{s_t\}$ where $\text{VLC}\{\cdot\}$ is the variable length coder and s_t is the quantized DCT coefficients of the interframe motion-compensated residual. Therefore, the conventional hybrid coder (also called Spatial Domain Encoder or SE for short) can be described mathematically as an iterative function:

- Interframe encoding (P-frame) for SE:

$$\begin{aligned} s_t &= Q\{\text{DCT}\{x_t - \text{MC}(y_{t-1}, \hat{v}_t)\}\}, \\ y_t &= \text{MC}(y_{t-1}, \hat{v}_t) + \text{IDCT}\{Q^{-1}\{s_t\}\}, \end{aligned} \quad (1)$$

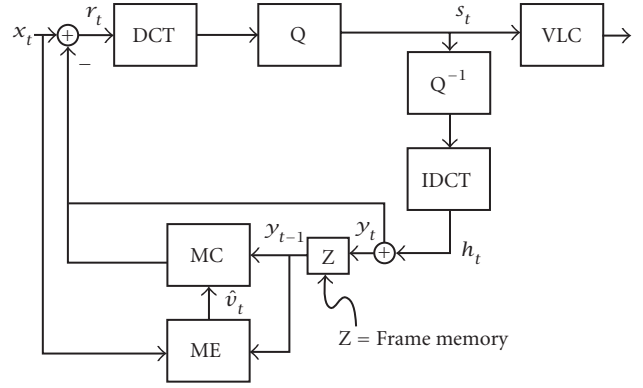


FIGURE 3: Modeling diagram for conventional hybrid motion-compensated DCT video coder.

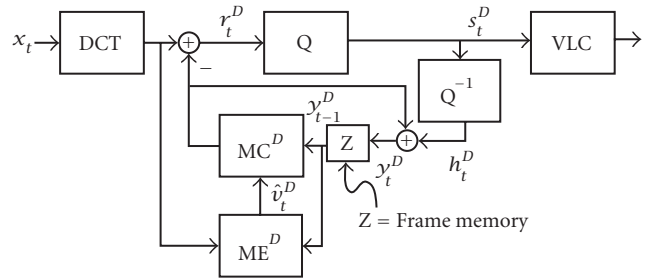


FIGURE 4: Modeling diagram for fully DCT-based motion-compensated video coder.

where y_t is the state of the coder stored in the frame memory, and the estimated motion vector

$$\hat{v}_t = \text{ME}(y_{t-1}, x_t). \quad (2)$$

- Intraframe encoding (I-frame) for SE:

$$\begin{aligned} s_0 &= Q\{\text{DCT}\{x_0\}\}, \\ y_0 &= h_0 = \text{IDCT}\{Q^{-1}\{Q\{\text{DCT}\{x_0\}\}\}\}. \end{aligned} \quad (3)$$

The above formulation considers only a group of pictures with the frame x_0 encoded as an I-frame and the rest of frames x_t as P-frames. Without loss of generality, B-frames are not considered here. Then s_t and \hat{v}_t are entropy coded into the bit stream sent to a decoder.

The fully DCT-based coder structure is redrawn in Figure 4 where the TD-ME block is also splitted, explicitly for formulating the encoding process, into two sections: the motion estimation (ME^D) and the motion compensation (MC^D). Note that the superscript D in Figure 4 and equations (4), (5), and (6) below are used to distinguish the fully DCT-based structure (called Transform Domain Encoder or TE) from the SE. Similar to the formulation of the SE, we consider only one group of pictures with the first frame coded as an I-frame and the rest as P-frames. The recursive function describing the TE structure is listed as follows:

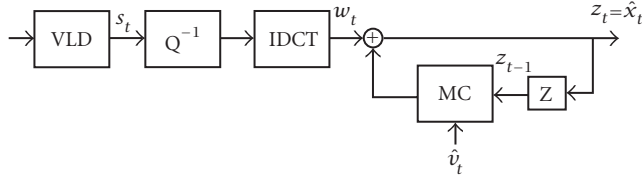


FIGURE 5: Modeling diagram for conventional hybrid motion-compensated DCT video decoder.

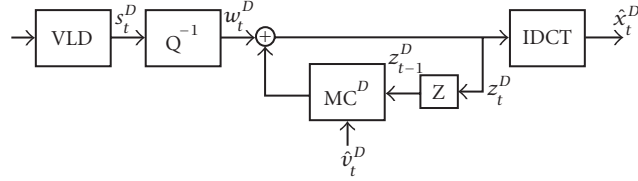


FIGURE 6: Modeling diagram for fully DCT-based motion-compensated video decoder.

- Interframe encoding (P-frame) for TE:

$$\begin{aligned} s_t^D &= Q \{ \text{DCT} \{ x_t \} - \text{MC}^D (y_{t-1}^D, \hat{v}_t^D) \}, \\ y_t^D &= \text{MC}^D (y_{t-1}^D, \hat{v}_t^D) + Q^{-1} \{ s_t^D \}, \end{aligned} \quad (4)$$

where s_t^D is the quantized DCT coefficients of the interframe motion-compensated residual, y_{t-1}^D is the state of the coder stored in the frame memory, and the estimated motion vector

$$\hat{v}_t^D = \text{ME}^D (y_{t-1}^D, \text{DCT} \{ x_t \}). \quad (5)$$

- Intraframe encoding (I-frame) for TE:

$$\begin{aligned} s_0^D &= Q \{ \text{DCT} \{ x_0 \} \}, \\ y_0^D &= h_0^D = Q^{-1} \{ Q \{ \text{DCT} \{ x_0 \} \} \}. \end{aligned} \quad (6)$$

In most motion-compensated video compression standards, the conventional hybrid decoder (Spatial Domain Decoder or SD for short) is usually cited for its conceptual simplicity. The SD is also depicted in Figure 5 where s_t reappears at the output of VLD (Variable Length Decoder) because variable length coding is considered as a reversible process, that is, $\text{VLD} \{ \text{VLC} \{ \alpha \} \} = \alpha$. The formulation for the SD is listed as follows:

- Interframe decoding (P-frame) for SD:

$$\begin{aligned} \hat{x}_t &= \text{IDCT} \{ Q^{-1} \{ s_t \} \} + \text{MC} (\hat{x}_{t-1}, \hat{v}_t), \\ z_t &= \hat{x}_t, \end{aligned} \quad (7)$$

where \hat{x}_t is the reconstructed frame image at the decoder and z_t will be stored in the frame memory as the state of the decoder.

- Intraframe decoding (I-frame) for SD:

$$z_0 = \hat{x}_0 = w_0 = \text{IDCT} \{ Q^{-1} \{ s_0 \} \}, \quad (8)$$

where $s_0 = Q \{ \text{DCT} \{ x_0 \} \}$.

Based on the DCT-based motion compensation schemes discussed above, the Transform Domain Decoder (TD), a fully DCT-based decoder, can be constructed [8, 11] as shown in Figure 6 with all the signals and components explicitly labeled. The TD can be modeled as follows:

- Interframe decoding (P-frame) for TD:

$$\begin{aligned} \hat{x}_t^D &= \text{IDCT} \{ Q^{-1} \{ s_t^D \} + \text{MC}^D (z_{t-1}^D, \hat{v}_t^D) \}, \\ z_t^D &= \text{MC}^D (z_{t-1}^D, \hat{v}_t^D) + Q^{-1} \{ s_t^D \}, \end{aligned} \quad (9)$$

where \hat{x}_t^D is the reconstructed frame image at the decoder and z_t^D will be stored in the frame memory as the state of the decoder.

- Intraframe decoding (I-frame) for TD:

$$\begin{aligned} \hat{x}_0^D &= \text{IDCT} \{ w_0^D \} = \text{IDCT} \{ Q^{-1} \{ s_0^D \} \}, \\ z_0^D &= w_0^D = Q^{-1} \{ s_0^D \}, \end{aligned} \quad (10)$$

where $s_0^D = Q \{ \text{DCT} \{ x_0 \} \}$.

No matter how we match the spatial- or transform-domain encoders (SE or TE) with the spatial- or transform-domain decoders (SD or TD), the reconstructed intraframes ($t = 0$) are related to the original frames as follows:

$$\hat{x}_0 = \text{IDCT} \{ Q^{-1} \{ Q \{ \text{DCT} \{ x_0 \} \} \} \}. \quad (11)$$

However, for intraframes, the reconstructed frames will have subtle difference with different matching pairs. As discussed below, the subtle difference in the states of the encoder and the decoder may have a divergent effect on the reconstructed images after encoding a long series of P-frames.

2.1. Matching SE with SD

If we send the encoded bit stream from an SE to an SD, then we can show that, as long as the encoder-decoder pair have matching DCT/IDCT, Q/Q^{-1} and MC implementation,

$$w_t = h_t, \quad z_t = y_t, \quad (12)$$

where $z_t (= \hat{x}_t)$ and y_t are the decoder and encoder states, respectively. In other words, the decoder can always track the state of the encoder even after a long series of P-frames. However, in practice, it is difficult to implement matching components at both the encoder and the decoder. As a result, a new intraframe is usually sent after a limited number of P-frames to reset the state of the codecs before diverging too far off. The reconstructed frames can be shown to be related to the original images as below:

$$\begin{aligned} \hat{x}_t &= \text{IDCT} \{ Q^{-1} \{ Q \{ \text{DCT} \{ x_t - \text{MC} (\hat{x}_{t-1}, \hat{v}_t) \} \} \} \} \\ &\quad + \text{MC} (\hat{x}_{t-1}, \hat{v}_t). \end{aligned} \quad (13)$$

2.2. Matching TE with TD

Similarly, if we send the encoded bit stream from a TE to a TD, then we can prove that, as long as the encoder-decoder pair have matching DCT/IDCT, Q/Q^{-1} and MC^D implementation,

$$w_t^D = h_t^D, \quad z_t^D = y_t^D, \quad (14)$$

where z_t^D and y_t^D are the decoder and encoder states, respectively. Therefore, the decoder can always track the state of the encoder even after a long series of P-frames in the same way as the SE-SD pair. It should be noted that, unlike the states of the SE-SD pair being the reconstructed frame images, the states of this TE-TD pair are the quantized DCT coefficients usually having many zeros in the high-frequency DCT domain and thus requiring less storage space than the SE-SD pair.

The reconstructed frames are related to the original images as below:

$$\hat{x}_t^D = \text{IDCT}\{z_t^D\}, \quad (15)$$

where

$$z_t^D = Q^{-1}\{Q\{\text{DCT}\{x_t\} - \text{MC}^D(z_{t-1}^D, \hat{v}_t^D)\} + \text{MC}^D(z_{t-1}^D, \hat{v}_t^D)\}. \quad (16)$$

2.3. Matching TE with SD

When the encoded bit stream from an SE is decoded by a TD, that is,

$$s_t = s_t^D, \quad \hat{v}_t = \hat{v}_t^D, \quad (17)$$

it can be shown that

$$w_t = \text{IDCT}\{h_t^D\}. \quad (18)$$

The reconstructed frames are related to the original images as below:

$$\hat{x}_t = \text{IDCT}\{Q^{-1}\{Q\{\text{DCT}\{x_t\} - \text{MC}^D(y_{t-1}^D, \hat{v}_t)\} + \text{MC}(\hat{x}_{t-1}, \hat{v}_t)\}\}. \quad (19)$$

However, the contents of the frame memories (z_t and y_t^D) of the TE encoder and the SD decoder are quite different: the frame memory of the TE holds the quantized DCT coefficients of the reconstructed frames whereas the SD frame memory stores the reconstructed images. Now we need to show the following theorem.

Theorem 1. *If the SD decoder and the TE encoder satisfy the matching MC- MC^D condition as described as follows:*

$$\begin{aligned} \text{MC}(\text{IDCT}\{y_{t-1}^D\}, \hat{v}_t^D) &= \text{IDCT}\{\text{MC}^D(y_{t-1}^D, \hat{v}_t^D)\}, \\ \text{IDCT}\{a + b\} &= \text{IDCT}\{a\} + \text{IDCT}\{b\}, \end{aligned} \quad (20)$$

then both frame memory contents can maintain a fixed relationship in order for the SD decoder to track the state of the TE encoder for accurately decoding a long series of P-frames in

high-fidelity, that is,

$$z_t = \text{IDCT}\{y_t^D\}. \quad (21)$$

Proof. For intraframes ($t = 0$),

$$\begin{aligned} z_0 &= \text{IDCT}\{Q^{-1}\{Q\{\text{DCT}\{x_0\}\}\}\} \\ &= \text{IDCT}\{h_0^D\} = \text{IDCT}\{y_0^D\}. \end{aligned} \quad (22)$$

For $t = 1$, $z_1 = \text{IDCT}\{Q^{-1}\{s_1^D\}\} + \text{MC}(z_0, \hat{v}_1^D)$, and

$$\begin{aligned} \text{IDCT}\{y_1^D\} &= \text{IDCT}\{Q^{-1}\{s_1^D\} + \text{MC}^D(y_0^D, \hat{v}_1^D)\} \\ &= \text{IDCT}\{Q^{-1}\{s_1^D\}\} + \text{IDCT}\{\text{MC}^D(y_0^D, \hat{v}_1^D)\} \\ &= z_1 + [\text{IDCT}\{\text{MC}^D(y_0^D, \hat{v}_1^D)\} - \text{MC}(z_0, \hat{v}_1^D)] \\ &= z_1 \end{aligned} \quad (23)$$

if $\text{IDCT}\{\text{MC}^D(y_0^D, \hat{v}_1^D)\} = \text{MC}(\text{IDCT}\{y_0^D\}, \hat{v}_1^D)$. The first equality holds for the distributive property of IDCT.

Now assuming that $z_t = \text{IDCT}\{y_t^D\}$,

$$\begin{aligned} \text{IDCT}\{y_{t+1}^D\} &= \text{IDCT}\{Q^{-1}\{s_{t+1}^D\} + \text{MC}^D(y_t^D, \hat{v}_{t+1}^D)\} \\ &= \text{IDCT}\{Q^{-1}\{s_{t+1}^D\}\} + \text{IDCT}\{\text{MC}^D(y_t^D, \hat{v}_{t+1}^D)\} \\ &= z_{t+1} + [\text{IDCT}\{\text{MC}^D(y_t^D, \hat{v}_{t+1}^D)\} - \text{MC}(z_t, \hat{v}_{t+1}^D)] \\ &= z_{t+1} \end{aligned} \quad (24)$$

if $\text{IDCT}\{\text{MC}^D(y_t^D, \hat{v}_{t+1}^D)\} = \text{MC}(\text{IDCT}\{y_t^D\}, \hat{v}_{t+1}^D)$. \square

The first requirement is the matching condition of MC- MC^D simply stating that the conventional motion compensated frame (MC) must be the same as the IDCT (image) of the DCT-based motion compensated frame (MC^D). The second condition says that the distributive property of IDCT must also be maintained true even in the finite-length and fixed-point implementation though it holds in theory.

When the first requirement cannot be satisfied, we find that the encoder and the decoder may have progressively divergent states as more P-frames are inserted between two I-frames. We may define the difference between the encoder and decoder states as δ_t for the t th P-frame as follows:

$$\delta_t \triangleq \text{IDCT}\{y_t^D\} - z_t. \quad (25)$$

In other words, δ_t is defined as the encoder state minus the decoder state. We further assume that the second condition holds and the spatial-domain and transform-domain motion compensators produce results differing by a fixed amount Δ for each pixel for the sake of simplicity in the analysis

$$\text{IDCT}\{\text{MC}^D(a, v)\} - \text{MC}(\text{DCT}\{a\}, v) = \Delta. \quad (26)$$

We know that $\delta_0 = \text{IDCT}\{y_0^D\} - z_0 = 0$. Also

$$\begin{aligned} \delta_1 &= \text{IDCT}\{y_1^D\} - z_1 \\ &= \text{IDCT}\{\text{MC}^D(y_0^D, \hat{v}_1^D)\} - \text{MC}(\text{IDCT}\{y_0^D\}, \hat{v}_1^D) \\ &= \Delta. \end{aligned} \quad (27)$$

Similarly,

$$\begin{aligned}\delta_2 &= \text{IDCT}\{\text{MC}^D(\gamma_1^D, \hat{v}_2^D)\} - \text{MC}(z_1, \hat{v}_2^D) \\ &= \text{IDCT}\{\text{MC}^D(\gamma_1^D, \hat{v}_2^D)\} - \text{MC}(\text{IDCT}\{\gamma_1^D\} + \delta_1, \hat{v}_2^D).\end{aligned}\quad (28)$$

Usually the commonly used spatial-domain motion compensation algorithms are distributive:

$$\text{MC}(a + b, v) = \text{MC}(a, v) + \text{MC}(b, v). \quad (29)$$

Since Δ is assumed to be fixed independent of its pixel position,

$$\begin{aligned}\delta_2 &= \text{IDCT}\{\text{MC}^D(\gamma_1^D, \hat{v}_2^D)\} - \text{MC}(\text{IDCT}\{\gamma_1^D\}, \hat{v}_2^D) \\ &\quad + \text{MC}(\delta_1, \hat{v}_2^D) \\ &= \Delta + \Delta = 2\Delta.\end{aligned}\quad (30)$$

In general,

$$\begin{aligned}\delta_t &= \text{IDCT}\{\text{MC}^D(\gamma_{t-1}^D, \hat{v}_t^D)\} - \text{MC}(z_{t-1}, \hat{v}_t^D) \\ &= \Delta + (t-1)\Delta = t\Delta.\end{aligned}\quad (31)$$

This suggests that the state difference will diverge in proportion to the number of P-frames between two I-frames. If Δ is positive, then the decoded picture will become dimmer and dimmer; otherwise, the decoded picture will become brighter and brighter.

2.4. Matching SE with TD

A number of papers (see [8, 11, 12]) discuss the implementation of a DCT-based decoder taking an SE-encoded bit stream but none of them has addressed the issue of divergent states if their DCT-based decoder is not matched properly with the conventional hybrid encoder.

Consider the case when the encoded bit stream from an SE is decoded by a TD, that is,

$$s_t^D = s_t, \quad \hat{v}_t^D = \hat{v}_t. \quad (32)$$

We can show that

$$h_t = \text{IDCT}\{w_t^D\}. \quad (33)$$

The reconstructed frames are constructed from the original images accordingly:

$$\begin{aligned}\hat{x}_t^D &= \text{IDCT}\left\{\text{Q}^{-1}\left\{\text{Q}\left\{\text{DCT}\{x_t - \text{MC}(\gamma_{t-1}, \hat{v}_t)\}\right\}\right\}\right. \\ &\quad \left. + \text{MC}^D(z_{t-1}^D, \hat{v}_t^D)\right\}.\end{aligned}\quad (34)$$

In the same way as in the case of the TE-SD pair, we are going to show the following theorem.

Theorem 2. *If the TD decoder and the SE encoder satisfy the matching MC-MC^D condition as described in the following:*

$$\begin{aligned}\text{MC}(\text{IDCT}\{z_{t-1}^D\}, \hat{v}_t) &= \text{IDCT}\{\text{MC}^D(z_{t-1}^D, \hat{v}_t)\}, \\ \text{IDCT}\{a + b\} &= \text{IDCT}\{a\} + \text{IDCT}\{b\},\end{aligned}\quad (35)$$

then both frame memory contents can maintain a fixed relationship in order for the TD decoder to track the state of the SE encoder and accurately decode P-frames continuously in high-fidelity:

$$\gamma_t = \text{IDCT}\{z_t^D\}. \quad (36)$$

Proof. For intraframes ($t = 0$), $\gamma_0 = \text{IDCT}\{\text{Q}^{-1}\{\text{Q}\{\text{DCT}\{x_0\}\}\}\} = \hat{x}_0 = \text{IDCT}\{z_0^D\}$. For $t = 1$, $\gamma_1 = h_1 + \text{MC}(\text{IDCT}\{z_0^D\}, \hat{v}_1)$, and

$$\begin{aligned}\text{IDCT}\{z_1^D\} &= \text{IDCT}\{w_1^D + \text{MC}^D(z_0^D, \hat{v}_1)\} \\ &= \text{IDCT}\{w_1^D\} + \text{IDCT}\{\text{MC}^D(z_0^D, \hat{v}_1)\} \\ &= \gamma_1 + [\text{IDCT}\{\text{MC}^D(z_0^D, \hat{v}_1)\} - \text{MC}(\gamma_0, \hat{v}_1)] \\ &= \gamma_1\end{aligned}\quad (37)$$

if $\text{IDCT}\{\text{MC}^D(z_0^D, \hat{v}_1)\} = \text{MC}(\text{IDCT}\{z_0^D\}, \hat{v}_1)$.

Now assuming that $\gamma_t = \text{IDCT}\{z_t^D\}$,

$$\begin{aligned}\text{IDCT}\{z_{t+1}^D\} &= \text{IDCT}\{w_{t+1}^D + \text{MC}^D(z_t^D, \hat{v}_{t+1})\} \\ &= \text{IDCT}\{w_{t+1}^D\} + \text{IDCT}\{\text{MC}^D(z_t^D, \hat{v}_{t+1})\} \\ &= \gamma_{t+1} + [\text{IDCT}\{\text{MC}^D(z_t^D, \hat{v}_{t+1})\} - \text{MC}(\gamma_t, \hat{v}_{t+1})] \\ &= \gamma_{t+1}\end{aligned}\quad (38)$$

if $\text{IDCT}\{\text{MC}^D(z_t^D, \hat{v}_{t+1})\} = \text{MC}(\text{IDCT}\{z_t^D\}, \hat{v}_{t+1})$. \square

Notice that the first requirement of a matching MC-MC^D pair is the same as the matching condition for the TE-SD pair. This implies that if we can build a matching MC-MC^D pair satisfying the first requirement, then we can build a TE and TD so that we can mix the encoder-decoder pair in any combination without reaching the divergent states.

Similar to the case of matching TE with SD, we can derive how much the decoder state diverges from the encoder state in terms of the number of P-frames between two I-frames, if the first condition cannot be met. Define the state difference, δ'_t for the t th P-frame as follows:

$$\delta'_t \triangleq -(\gamma_t - \text{IDCT}\{z_t^D\}). \quad (39)$$

Please note that δ'_t is defined as the decoder state minus the encoder state, opposite to δ_t in the case of matching TE-SD. Similarly, we assume that the second condition holds and the spatial-domain and transform-domain motion compensators produce results differing by a fixed amount Δ for each pixel for the sake of simplicity in the analysis as in the TE-SD case:

$$\text{IDCT}\{\text{MC}^D(a, v)\} - \text{MC}(\text{DCT}\{a\}, v) = \Delta. \quad (40)$$

Therefore, $\delta'_0 = \text{IDCT}\{z_0^D\} - \gamma_0 = 0$. Also

$$\begin{aligned}\delta'_1 &= \text{IDCT}\{z_1^D\} - \gamma_1 \\ &= \text{IDCT}\{\text{MC}^D(z_0^D, \hat{v}_1)\} - \text{MC}(\text{IDCT}\{z_0^D\}, \hat{v}_1) \\ &= \Delta.\end{aligned}\quad (41)$$

The equality $\delta' = \delta = \Delta$ means that we can expect the same divergent amount in the codec states for the SE-TD case as for the TE-SD case. Similarly,

$$\begin{aligned}\delta'_2 &= \text{IDCT}\{\text{MC}^D(z_1^D, \hat{v}_2)\} - \text{MC}(\gamma_1, \hat{v}_2) \\ &= \text{IDCT}\{\text{MC}^D(z_1^D, \hat{v}_2)\} - \text{MC}(\text{IDCT}\{z_1^D\} - \Delta, \hat{v}_2).\end{aligned}\quad (42)$$

Because of the distributive property of spatial-domain motion compensation ($\text{MC}(a+b, v) = \text{MC}(a, v) + \text{MC}(b, v)$), and the assumption of Δ being fixed independent of its pixel position,

$$\begin{aligned}\delta'_2 &= \text{IDCT}\{\text{MC}^D(z_1^D, \hat{v}_2)\} - \text{MC}(\text{IDCT}\{z_1^D\}, \hat{v}_2) + \text{MC}(\Delta, \hat{v}_2) \\ &= \Delta + \Delta = 2\Delta.\end{aligned}\quad (43)$$

In general,

$$\begin{aligned}\delta'_t &= \text{IDCT}\{\text{MC}^D(z_{t-1}^D, \hat{v}_t)\} - \text{MC}(\gamma_{t-1}, \hat{v}_t) \\ &= \Delta + (t-1)\Delta = t\Delta.\end{aligned}\quad (44)$$

This suggests that the state difference will diverge in proportion to the number of P-frames between two I-frames in the opposite direction to the diverging state difference for the case of matching TE with SD. If Δ is positive, then the decoded picture will become brighter and brighter.

2.5. Simulation results

We implement two types of encoders (SE and TE) and two types of decoders (SD and TD). In order to make fair comparison, we use the full search block matching approach for both encoders to find the estimate of motion vectors. Furthermore, in order to verify (31) and (44), we *deliberately* add a constant Δ to the transform-domain motion compensation process so that the motion-compensated frames from both the spatial-domain and transform-domain motion compensators differ by that constant Δ . Simulation is performed on the “Flower Garden” sequence. The simulation results for the cases of matching TE-SD and SE-TD are plotted in Figures 7 and 8, respectively. Both figures show that the codec state differences for both cases grow linearly with respect to the number of P-frames as predicted by (31) and (44). For the case of TE-TD, the TD decoder can keep track of the TE encoder state, as shown in Figure 9.

3. DCT-BASED MOTION COMPENSATION SCHEMES

Manipulation of compressed video data in DCT domain has been recognized as an important component in many advanced video applications (cf. [8, 11, 13, 14, 15, 16, 17]). In a video bridge where multiple sources of compressed video are combined and retransmitted in a network, techniques of manipulation and retransmission of compressed video streams entirely in DCT domain eliminate the need to build a decoding/encoding pair. Furthermore, manipulation in DCT domain provides flexibility to match heterogeneous Quality of Service requirements with different network or user resources

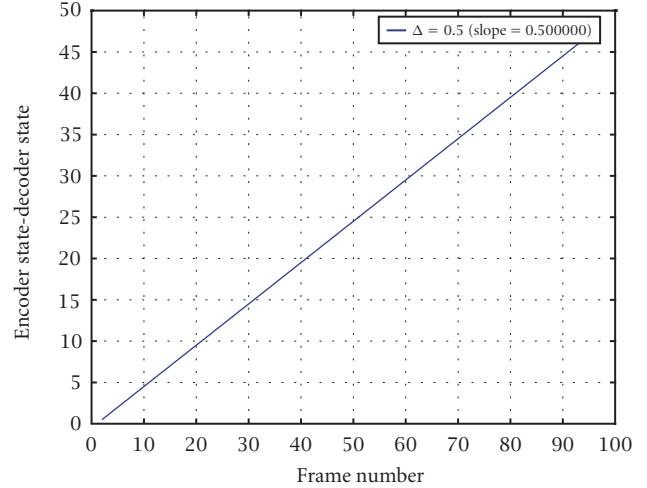


FIGURE 7: Simulation result of the “Flower Garden” sequence for matching TE-SD with $\Delta = 0.5$.

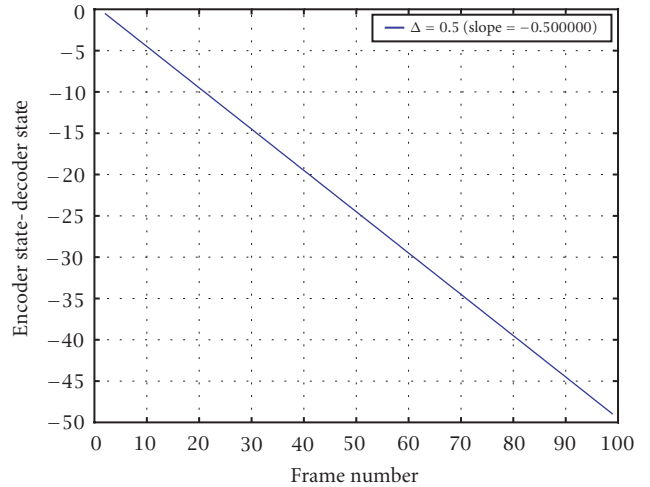


FIGURE 8: Simulation result of the “Flower Garden” sequence for matching SE-TD with $\Delta = 0.5$.

such as prioritization of signal components from low order DCT coefficients to fit low-end communication resources. Finally, many manipulation functions can be performed in the DCT domain more efficiently than in the spatial domain [14] due to a much lower data rate and removal of the decoding/encoding pair. However, all the earlier works have been focused mainly on manipulation at the decoder side.

To serve the purpose of building a fully DCT-based motion compensated video coder, our aim is to develop the techniques of motion compensation in the DCT domain without converting back to the spatial domain before motion compensation. In [14], the method of pixelwise (integer-pel) translation in the DCT domain is proposed for extracting a DCT block out of four neighboring DCT blocks at an arbitrary position. Though addressing a different scenario, this method can be applied after modification to integer-pel

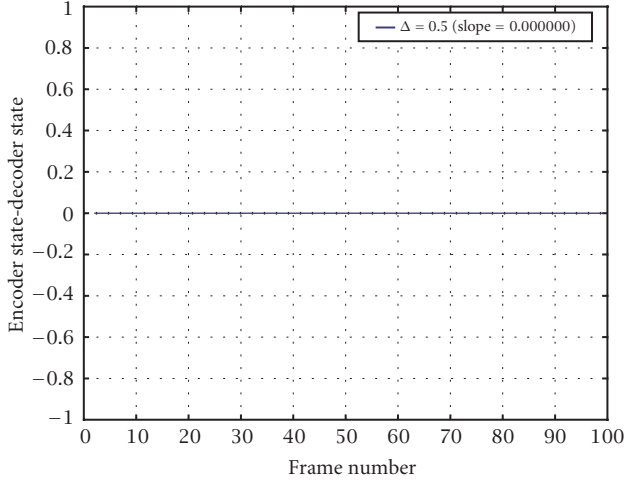


FIGURE 9: Simulation result of the “Flower Garden” sequence for matching TE-TD with $\Delta = 0.5$.

motion compensation in the DCT domain. For subpel motion compensation, we derive an equivalent form of bilinear interpolation in the DCT domain and then show that it is possible to perform other interpolation functions for achieving more accurate and visually better approximation in the DCT domain without increasing the complexity.

3.1. Integer-pel DCT-based motion compensation

As illustrated in Figure 10a, after motion estimation, the current block \mathbf{C} of size $N \times N$ in the current frame I_t can be best predicted from the block displaced from the current block position by the estimated motion vector (d_u, d_v) in the spatial domain. This motion estimate determines which four contiguous predefined DCT blocks are chosen for the prediction of the current block out of eight surrounding DCT blocks and the block at the current block position. To extract the displaced DCT block in the DCT domain, a direct method is used to obtain separately from these four contiguous blocks four subblocks which can be combined together to form the final displaced DCT block as shown in Figure 10b with the upper-left, lower-left, upper-right, and lower-right blocks from the previous frame I_{t-1} labeled as \mathbf{B}_1 , \mathbf{B}_2 , \mathbf{B}_3 , and \mathbf{B}_4 , respectively [14]. Subblocks \mathbf{S}_i are extracted in the spatial domain from these four blocks by pre-multiplication and post-multiplication of the windowing/shifting matrices, \mathbf{H}_i and \mathbf{V}_i :

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{B}_k \mathbf{V}_k, \quad \text{for } k = 1, \dots, 4, \quad (45)$$

where \mathbf{H}_k and \mathbf{V}_k are the $N \times N$ windowing/shifting matrices defined as

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{h_1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{V}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{v_1} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{H}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{h_2} & \mathbf{0} \end{bmatrix}, \quad \mathbf{V}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{v_2} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{H}_3 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{h_3} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{V}_3 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{v_3} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{H}_4 = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{h_4} & \mathbf{0} \end{bmatrix}, \quad \mathbf{V}_4 = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{v_4} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (46)$$

Here \mathbf{I}_n is the $n \times n$ identity matrix, that is, $\mathbf{I}_n = \text{diag}\{1, \dots, 1\}$ and n is determined by the height/width of the corresponding subblock. These pre-multiplication and post-multiplication matrix operations can be visualized in Figure 10c where the overlapped grey areas represent the extracted subblock. Then these four subblocks are summed to form the desired translated block $\hat{\mathbf{B}}_{ref}$.

If we define the DCT operation on an $N \times N$ matrix \mathbf{B} as

$$\text{DCT}\{\mathbf{B}\} = \mathbf{D}\mathbf{B}\mathbf{D}^T, \quad (47)$$

where the (k, m) element of \mathbf{D} is the DCT-II kernel:

$$\mathbf{D}(k, m) = \frac{2}{N} C(k) \cos \frac{k\pi}{N} \left(m + \frac{1}{2}\right), \quad \text{for } k, m = 0, \dots, N-1. \quad (48)$$

Therefore, $\mathbf{D}^T \mathbf{D} = (2/N) \mathbf{I}_N$. The formation of the DCT of $\hat{\mathbf{B}}_{ref}$ in the DCT domain can be described in this equation:

$$\text{DCT}\{\hat{\mathbf{B}}_{ref}\} = \left(\frac{N}{2}\right)^2 \sum_{k=1}^4 \text{DCT}\{\mathbf{H}_k\} \text{DCT}\{\mathbf{B}_k\} \text{DCT}\{\mathbf{V}_k\}. \quad (49)$$

This corresponds to pre- and post-multiplication of the DCT transformed \mathbf{H}_k and \mathbf{V}_k with the DCT of \mathbf{B}_k since DCT is a unitary orthogonal transformation and is guaranteed to be distributive to matrix multiplications. The DCT of the motion-compensated residual (displaced frame difference or DFD) for the current block \mathbf{C} is, therefore,

$$\text{DCT}\{\text{DFD}\} = \text{DCT}\{\hat{\mathbf{B}}_{ref}\} - \text{DCT}\{\mathbf{C}\}. \quad (50)$$

$\text{DCT}\{\mathbf{H}_k\}$ and $\text{DCT}\{\mathbf{V}_k\}$ can be precomputed and stored in the memory. Furthermore, many high-frequency coefficients of $\text{DCT}\{\mathbf{B}_k\}$ or displacement estimates are zeros (i.e., sparse and block aligned reference blocks), making the actual number of computations in (49) small. In [14], simulation results show that the DCT-domain approach is faster than the spatial-domain approach by about 10% to 30%. Further simplification is also possible as seen from Figure 10b that

$$\mathbf{H}_U = \mathbf{H}_1 = \mathbf{H}_3, \quad \mathbf{H}_L = \mathbf{H}_2 = \mathbf{H}_4, \quad (51)$$

$$\mathbf{V}_L = \mathbf{V}_1 = \mathbf{V}_2, \quad \mathbf{V}_R = \mathbf{V}_3 = \mathbf{V}_4.$$

Therefore, only four windowing/shifting matrices need to be accessed from the memory instead of eight.

In [8, 18], further savings in the computation of the windowing/shifting matrices is made by using fast DCT. It is reported that 47% reduction in computational complexity with fast DCT over the brute-force method without the assumption of sparseness and 68% with only the top-left 4×4 subblocks being nonzero can be achieved with the use of fast DCT.

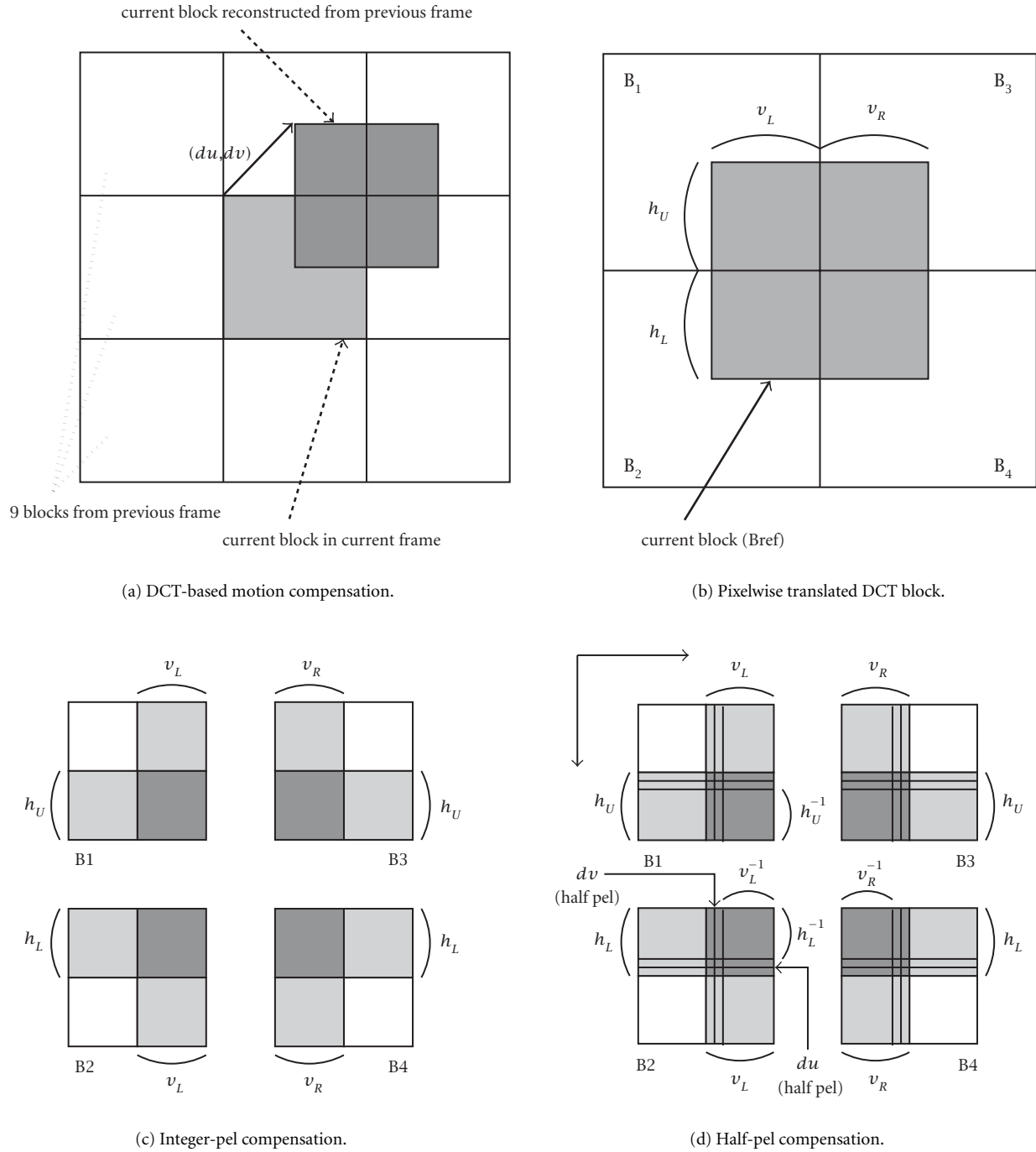


FIGURE 10: (a) Prediction of current block in current frame from four contiguous DCT blocks selected among nine neighboring blocks in previous frame based upon the estimated displacement vector for current block. (b) Schematic diagram of how a pixelwise translated DCT block is extracted from four contiguous DCT blocks. (c) Decomposition of integer-pel DCT-based translation as four matrix multiplication operations. (d) Decomposition of half-pel DCT-based translation as four matrix multiplication operations.

3.2. Subpixel DCT-based motion compensation

For the case of subpixel motion, interpolation is used to predict interpixel values. According to the MPEG standards, bilinear interpolation is recommended for its simplicity in

implementation and effectiveness in prediction [2, 3], though it is well known that a range of other interpolation functions, such as cubic, spline, Gaussian, and Lagrange interpolations, can provide better approximation accuracy and more pleasant visual quality (see [19, 20, 21, 22]). The complexity

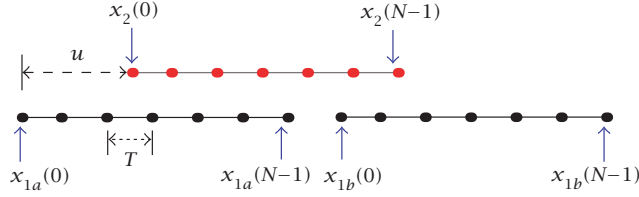


FIGURE 11: Illustration of extraction of the subpel displaced block $x_2(n)$ from two adjacent 1-D blocks $x_{1a}(n)$ and $x_{1b}(n)$ with bilinear interpolation.

argument is true if the interpolation operation is performed in the spatial domain, but in the DCT domain, it is possible to employ better interpolation functions than the bilinear interpolation without any additional computational load increase.

3.2.1 Interpolation filter

For simplicity of derivations, we start with the one-dimensional half-pel bilinear interpolation and then proceed to the two-dimensional case of quarter-pel accuracy with other interpolation functions. Consider two one-dimensional adjacent blocks, $x_{1a}(n)$ and $x_{1b}(n)$ for $n = 0, \dots, N-1$, as shown in Figure 11. We want to extract a block $\{x_2(n)\}_{n=0}^{N-1}$ that displaced u pixels to the right of $x_{1a}(0)$ where u is supposed to be an odd multiple of 0.5 (i.e., half-pel motion). Therefore, we can show that

$$x_2(n) = \begin{cases} \frac{1}{2} [x_{1a}(N+n-i) + x_{1a}(N+n-i+1)], & 0 \leq n \leq i-2, \\ \frac{1}{2} [x_{1a}(N-1) + x_{1b}(0)], & n = i-1, \\ \frac{1}{2} [x_{1b}(n-i) + x_{1b}(n-i+1)], & N-1 \geq n \geq i, \end{cases} \quad (52)$$

where $i = \lceil u \rceil$. In the matrix form,

$$\bar{\mathbf{x}}_2 = \mathbf{G}_{BL}(i)\bar{\mathbf{x}}_{1a} + \mathbf{G}_{BR}(i)\bar{\mathbf{x}}_{1b}, \quad (53)$$

where $\bar{\mathbf{x}}_2$, $\bar{\mathbf{x}}_{1a}$, and $\bar{\mathbf{x}}_{1b}$ are the column vectors of $x_2(n)$, $x_{1a}(n)$, and $x_{1b}(n)$, respectively, and $\mathbf{G}_{BL}(i)$ and $\mathbf{G}_{BR}(i)$ are defined as follows:

$$\mathbf{G}_{BL}(i) = \frac{1}{2} \left\{ \begin{bmatrix} \mathbf{0} & \mathbf{I}_i \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{I}_{i-1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right\}, \quad (54)$$

$$\mathbf{G}_{BR}(i) = \frac{1}{2} \left\{ \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{N-i} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I}_{N-i+1} & \mathbf{0} \end{bmatrix} \right\}.$$

In the DCT domain,

$$\text{DCT}\{\bar{\mathbf{x}}_2\} = \text{DCT}\{\mathbf{G}_{BL}(i)\}\text{DCT}\{\bar{\mathbf{x}}_{1a}\} + \text{DCT}\{\mathbf{G}_{BR}(i)\}\text{DCT}\{\bar{\mathbf{x}}_{1b}\}. \quad (55)$$

Here, $\mathbf{G}_{BL}(i)$ and $\mathbf{G}_{BR}(i)$ can be regarded as bilinear inter-

polation filter matrices which act as a linear filter or transform. Therefore, $\mathbf{G}_{BL}(i)$ and $\mathbf{G}_{BR}(i)$ can be replaced by any FIR filter or interpolation function of finite duration (preferably with the length much smaller than the block size N).

3.2.2 Bilinear interpolated subpixel motion compensation

For the 2-D case, if (u, v) is the displacement of the reconstructed block $\hat{\mathbf{B}}_{ref}$ measured from the upper left corner of the block \mathbf{B}_1 , then for $h_U = \lceil u \rceil$ and $v_L = \lceil v \rceil$,

$$\text{DCT}\{\hat{\mathbf{B}}_{ref}\} = \sum_{k=1}^4 \text{DCT}\{\mathbf{H}_k\}\text{DCT}\{\mathbf{B}_k\}\text{DCT}\{\mathbf{V}_k\}, \quad (56)$$

where

$$\begin{aligned} \mathbf{H}_1 &= \mathbf{H}_3 = \mathbf{H}_U = \mathbf{G}_{BL}(h_U), \\ \mathbf{H}_2 &= \mathbf{H}_4 = \mathbf{H}_L = \mathbf{G}_{BR}(h_U), \\ \mathbf{V}_1 &= \mathbf{V}_2 = \mathbf{V}_L = \mathbf{G}_{BL}^T(v_L), \\ \mathbf{V}_3 &= \mathbf{V}_4 = \mathbf{V}_R = \mathbf{G}_{BR}^T(v_L). \end{aligned} \quad (57)$$

Here

$$[\mathbf{G}_{BL}(h_U) \mathbf{G}_{BR}(h_U)] = \begin{bmatrix} 0 & \cdots & 0 & 0.5 & 0.5 & 0 & \cdots \\ \vdots & \ddots & \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & 0.5 & 0.5 \end{bmatrix}. \quad (58)$$

Once again, $\mathbf{G}_{BL}(\cdot)$ and $\mathbf{G}_{BR}(\cdot)$ can be precomputed and stored in the memory as in the case of integer-pel motion compensation and thus the extra computational load for doing bilinear interpolation is eliminated.

3.2.3 Cubic interpolated subpixel motion compensation

Three different interpolation functions, namely cubic, cubic spline, and bilinear interpolations, are plotted in Figure 12a. As can be seen, the bilinear interpolation has the shortest filter length and the cubic spline has a longest ripple but the cubic spline has the smallest approximation error among these three (cf. [22]). To compromise between filter length and approximation accuracy, we choose the cubic interpolation in the simulation. By choosing the resolution of the filter as half a pixel length, the bilinear interpolation $f_{hb}(n) = [0.5, 1, 0.5]$ and the cubic interpolation $f_{hc}(n) = [-0.0625, 0, 0.5625, 1.0000, 0.5625, 0, -0.0625]$. From Figure 12b, it is clear that the contributions at the half-pel position from all the pixel values are summed up and give rise to the bilinear filter matrices $\mathbf{G}_{BL}(\cdot)$ and $\mathbf{G}_{BR}(\cdot)$. In a similar way, as in Figure 12c, the cubic filter matrices $\mathbf{G}_{CL}(\cdot)$ and $\mathbf{G}_{CR}(\cdot)$ can be defined as

$$\begin{aligned} \mathbf{G}_{CL}(i) &= \begin{bmatrix} \mathbf{0} & -0.0625\mathbf{I}_{i+1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & 0.5625\mathbf{I}_i \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &+ \begin{bmatrix} \mathbf{0} & 0.5625\mathbf{I}_{i-1} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & -0.0625\mathbf{I}_{i-2} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \end{aligned}$$

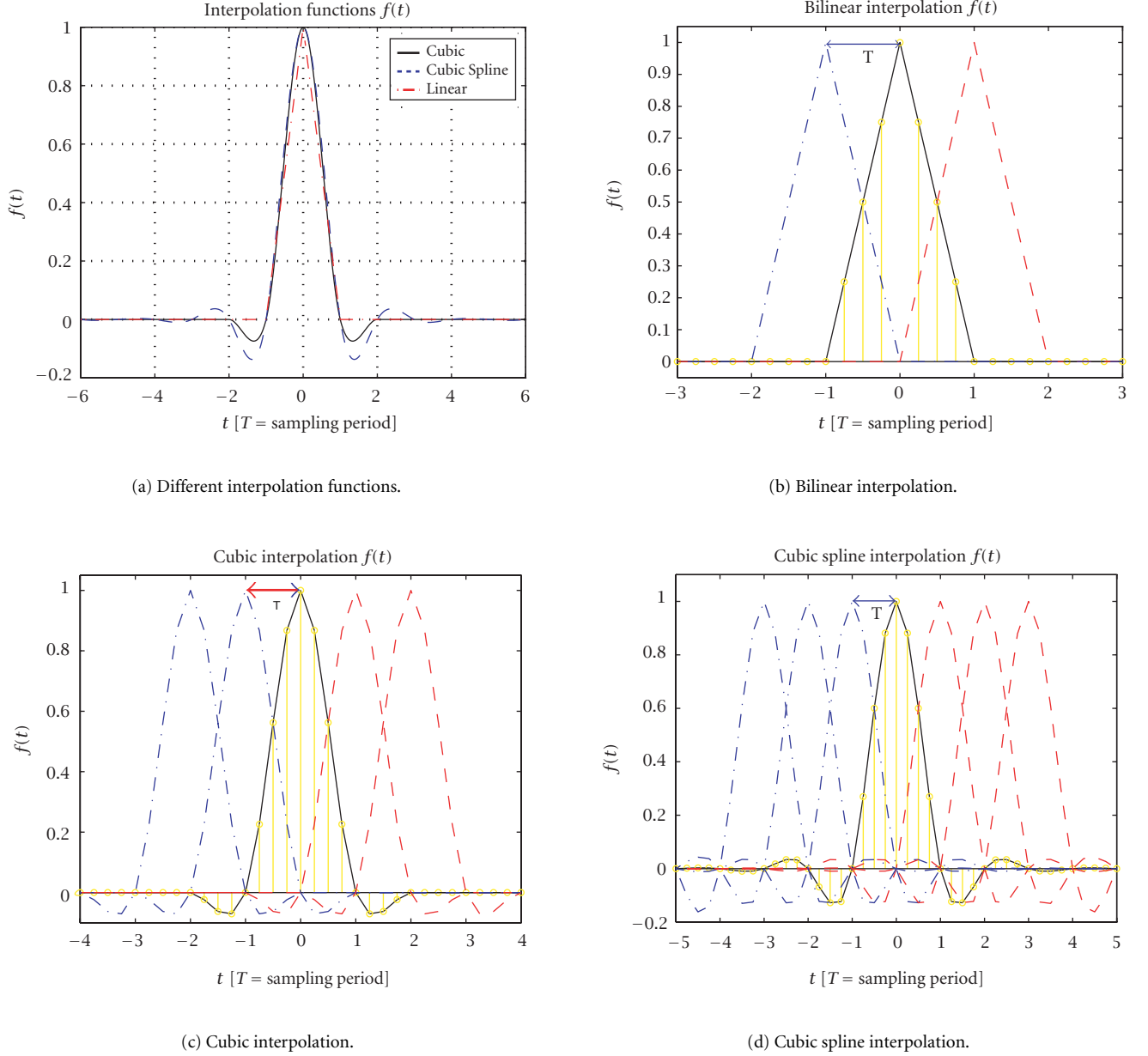


FIGURE 12: (a) plots of different interpolation functions. (b), (c), and (d) depict how to form a pre- or post-multiplication matrix for half-pel or even quarter-pel DCT-based motion compensation.

$$\begin{aligned}
 \mathbf{G}_{CR}(i) &= \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -0.0625\mathbf{I}_{N-i-1} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ 0.5625\mathbf{I}_{N-i} & \mathbf{0} \end{bmatrix} \\
 &+ \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ 0.5625\mathbf{I}_{N-i+1} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -0.0625\mathbf{I}_{N-i+2} & \mathbf{0} \end{bmatrix}.
 \end{aligned} \tag{59}$$

Here $\mathbf{G}_{CL}(\cdot)$ and $\mathbf{G}_{CR}(\cdot)$ can be precomputed and stored. Therefore, its computational complexity remains the same as both integer-pel and half-pel bilinear interpolated DCT-based motion compensation methods. The reconstructed DCT block and the corresponding motion-compensated

residual can be obtained in a similar fashion:

$$\begin{aligned}
 \text{DCT}\{\hat{\mathbf{B}}_{ref}\} &= \sum_{k=1}^4 \text{DCT}\{\mathbf{H}_k\} \text{DCT}\{\mathbf{B}_k\} \text{DCT}\{\mathbf{V}_k\}, \\
 \text{DCT}\{\text{DFD}\} &= \text{DCT}\{\hat{\mathbf{B}}_{ref}\} - \text{DCT}\{\mathbf{C}\},
 \end{aligned} \tag{60}$$

where

$$\begin{aligned}
 \mathbf{H}_1 = \mathbf{H}_3 = \mathbf{H}_U = \mathbf{G}_{CL}(h_U), & \quad \mathbf{H}_2 = \mathbf{H}_4 = \mathbf{H}_L = \mathbf{G}_{CR}(h_U), \\
 \mathbf{V}_1 = \mathbf{V}_2 = \mathbf{V}_L = \mathbf{G}_{CL}^T(v_L), & \quad \mathbf{V}_3 = \mathbf{V}_4 = \mathbf{V}_R = \mathbf{G}_{CR}^T(v_L).
 \end{aligned} \tag{61}$$

This idea can be extended to other interpolation functions such as sharpened Gaussian [20] and quarter-pel accuracy.

3.3. Interpolation by DCT/DST

Discrete Cosine Transform of type I (DCT-I) or type II (DCT-II) have successfully been applied to discrete interpolation applications (cf. [23, 24, 25]). Interpolation using DST or the more general W transform has also been studied over the years (cf. [26, 27]). Interpolation using DCT/DST is found to surpass the usual DFT interpolation method, especially for sinusoidal inputs, and, by taking into account of the boundary condition, very accurate interpolated values can be generated [25]. In the following, we will relate DCT-I interpolation with the Nyquist sampling theorem and show that, by means of DCT-I interpolation, the DCT-II coefficients of a half-pel shifted block can be directly obtained from the DCT-I coefficients of the original block. In this way, we can build a DCT-based motion compensation block without the need of converting the DCT coefficients back to pixels before motion compensation as required in the conventional approach.

3.3.1 DCT-I interpolated sequence

The procedure of interpolation using DCT-I is given as follows:

- (1) Compute the modified DCT-I defined as

$$Y(m) = \text{DCT-I}\{\gamma(n)\} \\ = \frac{2}{N} K_m \sum_{n=0}^N K_n \gamma(n) \cos\left(\frac{mn\pi}{N}\right); \quad \text{for } m=0, \dots, N, \quad (62)$$

where

$$K_m = \begin{cases} \frac{1}{2}, & m=0, \text{ or } N, \\ 1, & \text{otherwise.} \end{cases} \quad (63)$$

- (2) Append zeros to the DCT-I sequence $Y(m)$ to form another DCT-I sequence $Z(m)$:

$$Z(m) = \begin{cases} Y(m), & m=0, \dots, N, \\ 0, & m=N+1, \dots, MN. \end{cases} \quad (64)$$

- (3) Obtain the interpolated sequence $z(n)$ by calculating the modified inverse DCT-I transform of the zero-padded $Z(m)$ as below:

$$z(n) = \text{DCT-I}^{-1}\{Z(m)\} \\ = \sum_{m=0}^{MN} Z(m) \cos\left(\frac{mn\pi}{MN}\right); \quad \text{for } n=0, \dots, MN. \quad (65)$$

The above interpolation procedure of using DCT-I can be shown to be equivalent to upsampling of the reconstructed

bandlimited signal from the sequence $\gamma(n)$ by a pair of sinc-like η functions. By defining

$$Q_c(n, \nu) \triangleq \frac{2}{N} K_n \sum_{m=0}^N K_m \cos\left(\frac{mn\pi}{N}\right) \cos\left(\frac{m\nu\pi}{N}\right), \quad (66)$$

the interpolated sequence is

$$z(n) = \sum_{n'=0}^N \gamma(n') Q_c\left(n', \frac{n}{M}\right). \quad (67)$$

Notice that

$$Q_c(n, \nu) = \frac{K_n}{N} \left[\sum_{m=0}^N K_m \cos\left(\frac{m\pi}{N}(n-\nu)\right) + \sum_{m=0}^N K_m \cos\left(\frac{m\pi}{N}(n+\nu)\right) \right] \\ = \frac{K_n}{N} \{\eta(n-\nu) + \eta(n+\nu)\}, \quad (68)$$

where

$$\eta(x) \triangleq \sum_{m=0}^N K_m \cos\left(\frac{m\pi}{N}(x)\right). \quad (69)$$

It can be shown that

$$\eta(x) = \frac{1}{2} \sin \pi x \cdot \frac{\cos(\pi x/2N)}{\sin(\pi x/2N)} \approx N \text{sinc}(x) \quad \text{if } \pi x \ll 2N. \quad (70)$$

Since the orthogonal equation is

$$\frac{2}{N} K_m \sum_{n=0}^N K_n \cos\left(\frac{m'n\pi}{N}\right) \cos\left(\frac{mn\pi}{N}\right) = \delta(m' - m), \quad (71)$$

we can show that

$$Q_c(n, \nu) = \delta(n - \nu) \quad \text{if } \nu \text{ is an integer.} \quad (72)$$

Therefore, $z(nM) = \gamma(n)$ for $n=0, \dots, N$. This satisfies the requirement of being an interpolation function.

From the Nyquist sampling theorem, a continuous bandlimited signal, $f(t)$, can be perfectly reconstructed from its sampled sequence, $f(nT)$ for the sampling interval T by a series of sinc functions (see [28, 29])

$$f(t) = \sum_{n=-\infty}^{\infty} f(nT) \text{sinc}\left(n - \frac{t}{T}\right). \quad (73)$$

The reconstructed $f(t)$ can then be resampled at a different sampling rate, $T_1 = T/M$, to generate an interpolated sequence $f(mT_1)$.

$$f(mT_1) = f(mT/M) = \sum_{n=-\infty}^{\infty} f(nT) \text{sinc}\left(n - \frac{m}{M}\right). \quad (74)$$

Therefore, interpolation using DCT-I expressed in (67) is the truncated version of (74) for a symmetric sequence $f(nT)$.

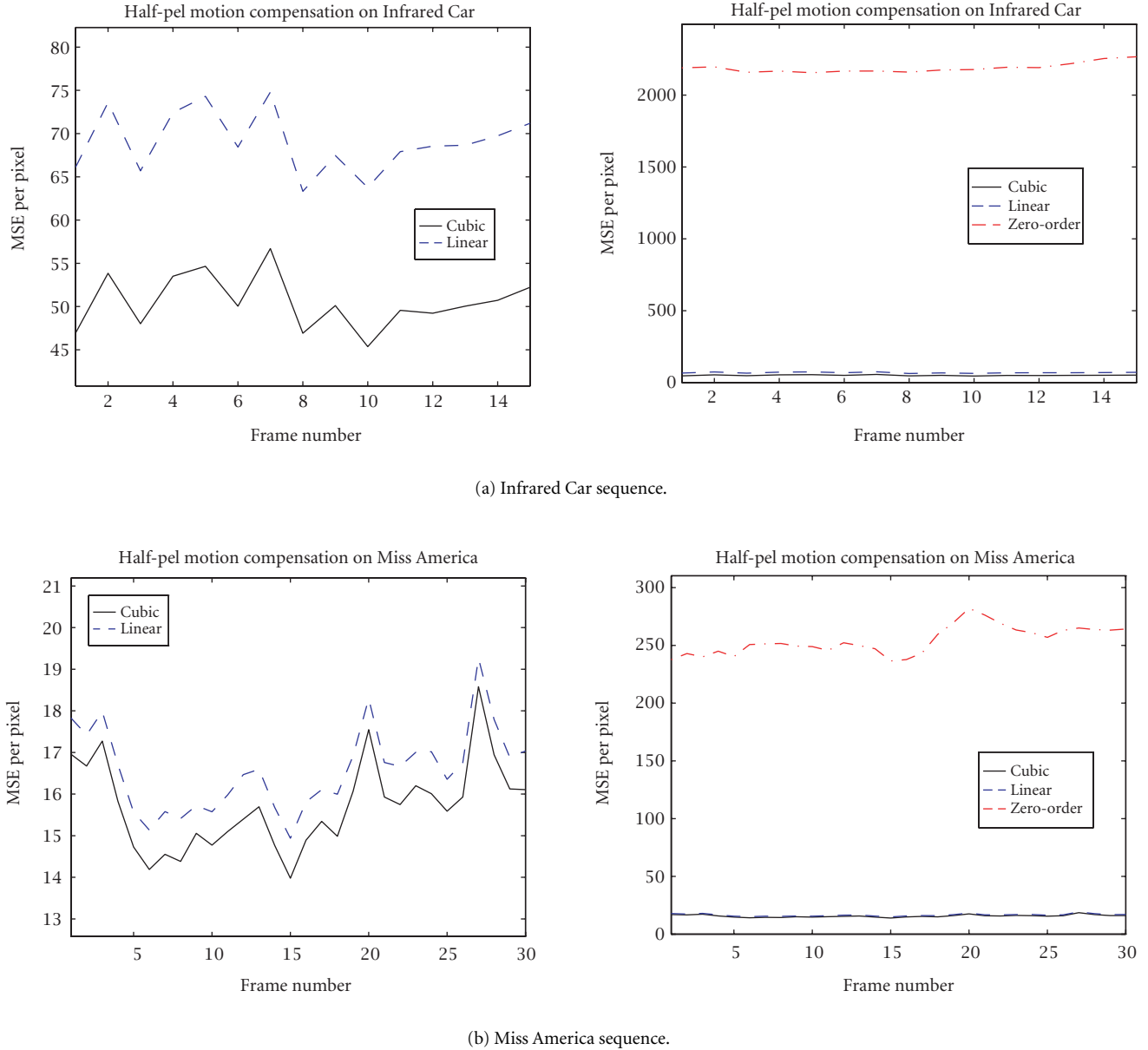


FIGURE 13: Pictures from the Infrared Car and Miss America sequences are subsampled and displaced by a half-pel motion vector with different motion compensation methods. The MSE-per-pixel values are obtained by comparing the original unsampled pixel values with the predicted pixel values of the motion compensated residuals. Zero-order interpolation means replication of sampled pixels as the predicted pixel values.

3.3.2 DCT-II of DCT-I interpolated half-pel motion compensated block

Given a sequence of length $N + 1$, $\{y(n); n = 0, \dots, N\}$, the half-pel shifted sequence $\{w(n); n = 0, \dots, N - 1\}$ is generated by sampling the DCT-I interpolated sequence of length $2N + 1$, $z(n)$ such that

$$w(i) = z(2i + 1) \quad \text{for } i = 0, \dots, N - 1. \quad (75)$$

The DCT-II coefficients $W(k)$ of $w(i)$ are found to have simple relationship with modified DCT-I coefficients $Y(m)$ of

the original sequence $y(n)$ as follows:

$$\begin{aligned} W(k) &\triangleq \text{DCT-II}\{w(i)\} \\ &= \frac{2}{N} C(k) \sum_{i=0}^{N-1} z(i) \cos \frac{k\pi}{N} \left(i + \frac{1}{2}\right) \\ &= C(k) \sum_{m=0}^{N-1} Y(m) [\delta(m+k) + \delta(m-k)] \end{aligned} \quad \text{for } k = 0, \dots, N - 1. \quad (76)$$

Therefore,

$$W(0) = \sqrt{2}Y(0); \quad W(k) = Y(k) \quad \text{for } k = 1, \dots, N-1. \quad (77)$$

With this simple relationship, once the modified DCT-I coefficients of an $N+1$ sequence are obtained, the DCT-II of the DCT-I interpolated half-pel shifted block will easily be obtained via (77).

3.3.3 Simulation results

Simulation is performed on the Infrared Car and Miss America sequences to demonstrate the effectiveness of our bilinear and cubic motion compensation methods.

The first set of simulations subsamples each picture $I_t(i, j)$ from the sequences (i.e., $y(i, j) = I_t(2 * i, 2 * j)$) and then this shrunk picture $y(i, j)$ is displaced by a half-pel motion vector (arbitrarily chosen as (2.5, 1.5)) with both bilinear and cubic interpolated motion compensation methods. The mean square errors per pixel (MSE) are computed as $MSE = (\sum_{i,j} [\hat{x}(i, j) - x(i, j)]^2) / N^2$ by treating the original unsampled pixels $I_t(2 * i + 1, 2 * j + 1)$ as the reference picture $x(i, j) = I_t(2 * i + 1, 2 * j + 1)$ where $\hat{x}(i, j)$ is the predicted pixel value from $y(i, j)$. As shown in Figure 13, the zero-order interpolation is also simulated for comparison. The zero-order interpolation, also called sample-and-hold interpolation, simply takes the original pixel value as the predicted half-pel pixel value [21]. As can be seen in Figure 13, both the bilinear and cubic methods have much lower MSE values than the zero-order method and also the cubic method performs much better than the bilinear counterpart without increased computational load.

Figures 14 and 15 show the results of another set of simulations in which the subpixel DCT-based motion compensation algorithms generate motion compensated residuals of the sequences “Infrared Car” and “Miss America,” respectively, based on the displacement estimates of the full search block matching algorithm, where the residuals are used to compute the MSE and BPS values for comparison. It can be seen that the cubic interpolation approach achieves lower MSE and BPS values than the bilinear interpolation.

4. CONCLUSION

Despite the conceptual simplicity in the conventional hybrid DCT motion compensated codec structures adopted for implementation of most DCT-based video coding standards, an alternative DCT-based video codec structure has been proposed in the past to address a number of shortcomings inherent in the conventional hybrid architecture. However, no prior research has been directed at what criteria can allow these two drastically different architectures to interoperate with each other in accordance with the standards.

In this paper, we establish the criteria for matching conventional codecs with fully DCT-based codecs such that compatibility with existing standards and interoperability with each other are guaranteed. We find that the key to this interoperability lies in the heart of the implementation of motion compensation modules performed in the spatial and

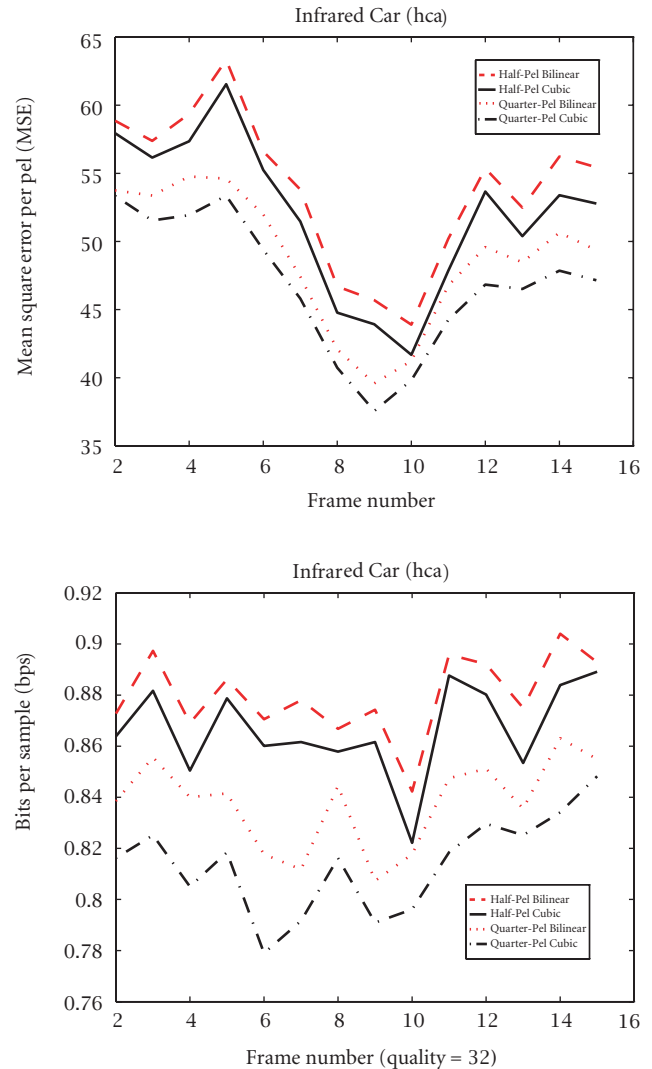


FIGURE 14: Pictures from the Infrared Car sequence are subsampled and displaced by a half-pel motion vector with different motion compensation methods. The MSE-per-pixel values are obtained by comparing the original unsampled pixel values with the predicted pixel values of the motion compensated residuals. Zero-order interpolation means replication of sampled pixels as the predicted pixel values.

transform domains at both the encoder and the decoder. Specifically, if the spatial-domain motion compensation is compatible with the transform-domain motion compensation, then the states in both the coder and the decoder will keep track of each other even after a long series of P-frames. Otherwise, the states will diverge in proportion to the number of P-frames between two I-frames. This sets an important criterion for the development of any DCT-based motion compensation schemes.

Following this criterion, we also discuss and develop some DCT-based motion compensation schemes as important building blocks of fully DCT-based codecs. For the case of subpixel motion compensation, DCT-based approaches

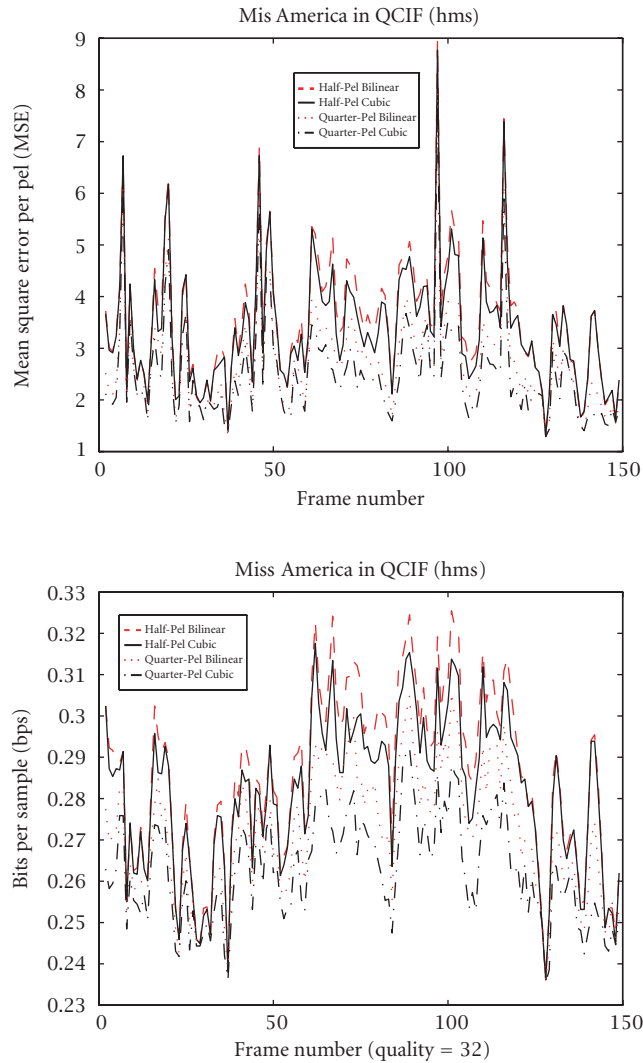


FIGURE 15: Pictures from the Miss America sequence are subsampled and displaced by a half-pel motion vector with different motion compensation methods. The MSE-per-pixel values are obtained by comparing the original unsampled pixel values with the predicted pixel values of the motion compensated residuals. Zero-order interpolation means replication of sampled pixels as the predicted pixel values.

allow more accurate interpolation without any increase in computation. Furthermore, a sparse number of DCT coefficients after quantization significantly decreases the number of calculations required for motion compensation. Coupled with the DCT-based motion estimation algorithms, it is possible to realize fully DCT-based codecs to overcome the disadvantages of conventional hybrid codecs.

REFERENCES

- [1] CCITT Recommendation H.261, *Video Codec for Audiovisual Services at $p \times 64$ kbit/s*, CCITT, August 1990.
- [2] CCITT Recommendation MPEG-1, *Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s*, ISO/IEC 11172, Geneva, Switzerland, 1993.
- [3] CCITT Recommendation MPEG-2, *Generic Coding of Moving Pictures and Associated Audio*, ISO/IEC 13818, Geneva, Switzerland, 1994 H.262.
- [4] J. S. McVeigh and S.-W. Wu, "Comparative study of partial closed-loop versus open-loop motion estimation for coding of HDTV," in *Proc. IEEE Workshop on Visual Signal Processing and Communications*, New Brunswick, September 1994, pp. 63–68.
- [5] H. Li, A. Lundmark, and R. Forchheimer, "Image sequence coding at very low bitrates: a review," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 589–609, 1994.
- [6] U. V. Koc and K. J. R. Liu, "DCT-based motion estimation," *IEEE Trans. Image Processing*, vol. 7, no. 7, pp. 948–965, 1998.
- [7] U. V. Koc and K. J. R. Liu, "Interpolation-free subpixel motion estimation techniques in DCT domain," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 460–487, 1998.
- [8] N. Merhav and V. Bhaskaran, "Fast algorithms for DCT-domain image downsampling and for inverse motion compensation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 468–476, 1997.
- [9] M. Song, A. Cai, and J. A. Sun, "Motion estimation in DCT domain," in *Proceedings of International Conference on Communication Technology (ICCT)*, 1996, vol. 2, pp. 670–674.
- [10] S. Ji and H. W. Park, "Moving object segmentation in DCT-based compressed video," *Electronics Letters*, vol. 36, no. 21, pp. 1769–1770, 2000.
- [11] J. Song and B.-L. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 10, no. 5, pp. 767–775, 2000.
- [12] R. Kleihorst and F. Cabrera, "Implementation of DCT-domain motion estimation and compensation," in *IEEE Workshop on Signal Processing Systems*, October 1998, pp. 53–62.
- [13] S. F. Chang and D. G. Messerschmitt, "A new approach to decoding and compositing motion-compensated DCT-based images," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, April 1993, vol. V, pp. 421–424.
- [14] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 1, pp. 1–11, 1995.
- [15] Y. Y. Lee and J. W. Woods, "Video post-production with compressed images," *SMPTE Journal*, vol. 103, pp. 76–84, February 1994.
- [16] B. C. Smith and L. Rowe, "Algorithms for manipulating compressed images," *IEEE Computer Graphics and Applications*, vol. 13, no. 5, pp. 34–42, 1993.
- [17] J.-B. Lee and B. G. Lee, "Transform domain filtering based on pipelining structure," *IEEE Trans. Signal Processing*, vol. 40, pp. 2061–2064, August 1992.
- [18] N. Merhav and V. Bhaskaran, "A fast algorithm for DCT-domain inverse motion compensation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, May 1996, vol. IV, pp. 2307–2310.
- [19] R. W. Schafer and L. R. Rabiner, "A digital signal processing approach to interpolation," *Proceedings of the IEEE*, vol. 61, no. 6, pp. 692–702, 1973.
- [20] W. F. Schreiber, *Fundamentals of Electronic Imaging Systems—Some Aspects of Image Processing*, Springer-Verlag, 3rd edition, 1993.
- [21] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.
- [22] H. Hou and H. C. Andrews, "Cubic splines for image interpolation and digital filtering," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 508–517, 1978.

- [23] Z. Wang, "Interpolation using type I discrete cosine transform," *Electronics Letters*, vol. 26, no. 15, pp. 1170, 1990.
- [24] J. I. Agbinya, "Interpolation using the discrete cosine transform," *Electronics Letters*, vol. 28, no. 20, pp. 1927, 1992.
- [25] C.-Y. Hsu and S.-M. Chen, "Discrete interpolation using the discrete cosine transform with the mapping of the boundary conditions," *IEEE Signal Processing Letters*, vol. 2, no. 10, pp. 185, 1995.
- [26] Z. Wang and L. Wang, "Interpolation using the fast sine transform," *Signal Processing*, vol. 26, no. 2, pp. 131–137, 1992.
- [27] Z. Wang, G. A. Jullien, and W. C. Miller, "The generalized discrete W transform and its application to interpolation," *Signal Processing*, vol. 36, no. 1, pp. 99–109, 1994.
- [28] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, Signal Processing. Prentice Hall, New Jersey, 1989.
- [29] S. P. Kim and W.-Y. Su, "Direct image resampling using block transform coefficients," *Signal Processing: Image Communication*, vol. 5, no. 3, pp. 259–272, 1993.

Ut-Va Koc is a Distinguished Member of Technical Staff (DMTS) at Bell Labs Research, Lucent Technologies, Murray Hill, New Jersey. He has published numerous peer-reviewed papers and book chapters on signal processing in multimedia and communications. Dr Koc has been active in serving as reviewer of various journals and conferences, editor for EURASIP Journal of Applied Signal Processing and guest co-chair in various international conferences. He received the Ph.D. degree from the University of Maryland, College Park and the B.S. degree from the National Chiao Tung University.



K. J. Ray Liu received the B.S. degree from the National Taiwan University, and the Ph.D. degree from UCLA, both in electrical engineering. He is a Professor of Electrical and Computer Engineering Department and Institute for Systems Research of University of Maryland, College Park. His research interests span broad aspects of signal processing algorithms and architectures; image/video compression, coding, and processing; wireless communications; security; and medical and biomedical technology in which he has published over 200 papers. Dr Liu received numerous awards including the 1994 National Science Foundation Young Investigator, the IEEE Signal Processing Society's 1993 Senior Award (Best Paper Award), IEEE Vehicular Technology Conference Best Paper Award, Amsterdam, 1999, and the George Corcoran Award in 1994 for outstanding contributions to electrical engineering education and the Outstanding Systems Engineering Faculty Award in 1996 in recognition of outstanding contributions in interdisciplinary research, both from the University of Maryland. Dr Liu is Editor-in-Chief of EURASIP Journal on Applied Signal Processing, and has been an Associate Editor of IEEE Transactions on Signal Processing, a Guest Editor of special issues on Multimedia Signal Processing of Proceedings of the IEEE, a Guest Editor of special issue on Signal Processing for Wireless Communications of IEEE Journal of Selected Areas in Communications, a Guest Editor of special issue on Multimedia Communications over Networks of IEEE Signal Processing Magazine, a Guest Editor of special issue on Multimedia over IP of IEEE Trans. on Multimedia, and an editor of Journal of VLSI Signal Processing Systems. He currently serves as the Chair of Multimedia Signal Processing Technical Committee of IEEE Signal Processing Society and the Series Editor of Marcel Dekker series on signal processing and communications.

