中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-05-001

A Block-Based SNR Scalable Wavelet Video Codec with Sub-pixel Motion Vectors and R-D Optimization

Cho-Chun Cheng and Wen-Liang Hwang

January 2005 || Technical Report No. TR-IIS-05-001
http://www.iis.sinica.edu.tw/LIB/TechRept.htm

A Block-Based SNR Scalable Wavelet Video Codec with Sub-pixel Motion Vectors and R-D Optimization

Cho-Chun Cheng and Wen-Liang Hwang
Institute of Information Science, Academia Sinica, Taipei, Taiwan

Abstract

We propose a block-based wavelet codec in which motion vector estimation and motion residual encoding are performed within a wavelet domain. By interpolation on dyadic wavelet transform, we show that motion vector estimation in the wavelet domain can achieve sub-pixel precision. To improve PSNR performance at low bit rates, we propose a bit-plane-based rate-distortion (R-D) optimization algorithm. This algorithm allocates an optimal number of bit-planes to each macroblock. Our codec is SNR scalable and our bit-stream syntax is fully compatible with that of H.263. Experiments show that our block-based wavelet codec outperforms a frame-based wavelet codec. Also, compared with H.263 baseline results, our wavelet codec is competitive at low bit rates, and is superior at higher bit rates.

1 Introduction

Still image compression based on a wavelet transform achieves great results and is the foundation of the JPEG 2000 image compression standard [18]. However, a wavelet video codec, either coding based on a 3D wavelet transform [19] or wavelet coding of residual frames [2, 20], is still in its infancy stage. Although encoding a video using wavelets more easily achieves scalability and rate control for an embedded codec than traditional DCT-based methods [1, 15], its video coding properties have not been fully explored. We aim to explore these properties and evaluate the performance of a wavelet-based hybrid video coding system. We are particularly interested in a wavelet coding structure, as shown in Figure 1, in which both motion vector estimation and residual coding are performed within a wavelet domain. This structure provides more flexibility than traditional methods in that we can exploit a perceptual-relevant contrast sensitivity function [22], obtain a progressive motion vector estimation [12], and extract context information in the wavelet domain for motion vector estimations. We use *all wavelet codec* to denote a video codec in which motion vector estimation and residual encoding are performed within a wavelet domain. This is to differentiate this codec from the wavelet codec that uses 3D wavelet transform and the codec that uses wavelets only to encode residual frames.

Current video motion vector estimation module uses a block-based translation model in which macroblocks in adjacent frames are related by translation. It is shown in [9, 12] that a discrete wavelet transform (DWT) has hierarchical motion compensation, which gives a larger range in which to perform motion compensation. However, DWT is not a translation-invariant representation, and this hinders its large motion compensation range advantage. Many other wavelet representations have a translation-invariant property at the cost of a coefficient redundancy increase [7, 10, 21]. In [13], the relationship between dyadic wavelet transform coefficients and translations of block discrete wavelet coefficients is explored to develop an algorithm that quickly estimates motion vectors from block-based DWT coefficients; however, its motion vectors are restricted only to integer precision. We propose an interpolation on dyadic wavelet transform coefficients and show that this interpolation is equivalent to performing the same interpolation on an image. We can obtain sub-pixel accurate motion vectors by performing motion vector estimation in the wavelet domain.

Sub-pixel motion vector estimation is an important property for improving the performance of motion compensation of a hybrid video coding system. Another important property for performance improvement is using R-D optimization to encode residual frames [17]. Using R-D optimization on residual frames improves coding effectiveness, but it introduces excessive complexity for video coding, especially in low-delay applications [8]. An R-D approach for video coding must be simple and efficient, even at the cost of coding effectiveness. We propose a simple sub-optimal rate-distortion (R-D) optimization technique that can be easily implemented in any embedded block-based codec.

We propose a block-based all wavelet codec that has the following features. One of its features is asymmetry in that the encoder uses a dyadic wavelet transform and the decoder uses an inverse DWT. Another feature is a sub-pixel precision motion vector estimation in the wavelet domain. Its R-D feature optimizes the number of bit-planes assigned to each macroblock for block-based residual encoding. Another feature is an H.263 syntax compatible bit-stream, in which only a few Macroblock and Block layer semantics differ from H.263 baseline.

We evaluate the performance of our codec in a wide range of bit rates, and compare it with the results of a frame-based wavelet codec and that of H.263 baseline. The dyadic wavelet transform and its implementation is given in Section 2. In Section 3, we propose our all wavelet codec and its features. Implementation issues and performance comparisons are given in Section 4. The conclusion is given in Section 5.

2 Algorithm Á Trous

In order to estimate motion vectors in a wavelet domain, we need to compute the block DWT coefficients of any translation of a block within a search region. Directly computing these block DWT coefficients is time-consuming. These coefficients can be obtained by computing a dyadic wavelet transform and re-arranging its coefficients, thus reducing time cost. A way to implement dyadic wavelet transform is through the algorithm á trous, an undecimated filter bank [11, 16].

A two-dimensional orthogonal wavelet can be obtained by a tensor-product of two one-dimensional orthogonal wavelets, $\{\phi(x), \psi(x)\}$ and $\{\phi(y), \psi(y)\}$. The results are wavelets $\Psi^1(x,y) = \psi(x)\phi(y), \Psi^2(x,y) = \phi(x)\psi(y)$, and $\Psi^3(x,y) = \psi(x)\psi(x)$, and scaling function $\Phi(x,y) = \phi(x)\phi(y)$. We use $g_{2^j}(x,y)$ to represent $\frac{1}{2^{2j}} g(\frac{x}{2^j}, \frac{y}{2^j})$ for any function $g(x,y)$. The dyadic wavelet transform of image $f(x,y)$ yields the multi-resolution image representation

$$f(x,y) = \sum_{k=1,2,3} \sum_{j=1}^{J} \mathcal{W}^k_{2^j} f * \Psi^k_{2^j}(x,y) + \mathcal{A}_{2^J} f * \Phi_{2^J}(x,y),$$

where the dyadic wavelet coefficients are

$$\begin{aligned} \mathcal{W}^k_{2^j} f(x,y) &= \int\int f(p,q) \frac{1}{2^{2j}} \Psi^k(\frac{x-p}{2^j}, \frac{y-q}{2^j}) dpdq, \quad \text{and} \\ \mathcal{A}_{2^J} f(x,y) &= \int\int f(p,q) \frac{1}{2^{2J}} \Phi(\frac{x-p}{2^J}, \frac{y-q}{2^J}) dpdq. \end{aligned}$$

The algorithm á trous yields a redundant wavelet representation of an image because this algorithm does not decimate filter bank coefficients. The dyadic wavelet transform is a translation invariant representation, which is essential for precise motion estimation. That is,

$$\begin{aligned} f(x,y) &\rightarrow \mathcal{A}_{2^J} f(x,y) \ \{\mathcal{W}^k_{2^j} f(x,y) | k = 1,2,3; j = 1, \cdots, J\} \quad \textit{and} \\ f(x+m, y+n) &\rightarrow \mathcal{A}_{2^J} f(x+m, y+n) \ \{\mathcal{W}^k_{2^j} f(x+m, y+n) | k = 1,2,3; j = 1, \cdots, J\}. \end{aligned}$$

To obtain DWT coefficients of a block from dyadic wavelet transform coefficients, let (x_0, y_0) be the top-left corner of a block of size 2^J by 2^J in a 2^N by 2^N image f. Then, the DWT coefficient at (m, n) of scale 2^i of the block is the dyadic wavelet coefficient of image f at $(x_0 + 2^i m, y_0 + 2^i n)$ of scale 2^i. The proof of this statement is given in Appendix 1. Figure 3 shows an example of generating block-based DWT coefficients from dyadic wavelet coefficients.

3 An All Wavelet Video Codec

As shown in Figure 2, the input video sequence of the encoder is first transformed into a dyadic wavelet domain, then block-based motion compensation is performed on DWT coefficients. This algorithm does not alter the traditional motion compensation algorithm in an image domain, except to perform motion vector estimation on DWT coefficients. After the incoming bit-stream is decoded, an inverse DWT is performed to obtain a reconstructed frame at the decoder side.

3.1 Sub-Pixel Precision Wavelet Motion Vector Estimation

In recent video coding standards [3, 4, 5], interpolation for sub-pixel prediction is commonly used to increase motion estimation precision. In an image domain, sub-pixel prediction is easily achieved by an interpolation operation. We show that the result of applying this interpolation operation on an image is equivalent to applying the same interpolation operation on the dyadic wavelet coefficients of the image at each subband. Figure 4 gives a simple schematic proof that uses noble identities (as shown in Figure 5) for a one-dimensional signal. The proof can easily be extended to a two-dimensional image because our wavelets are obtained by a tensor-product of two one-dimensional wavelets. If we choose the positions of filter A at a multiple of L to be the dirac sequence

$$A(n) = \begin{cases} \delta[n], & n = Lm \in Z \\ any\ number, & otherwise, \end{cases}$$

then the dyadic wavelet coefficients at position mL of the interpolated image of f are the same as the dyadic wavelet coefficients at position m of input f. This property of wavelet transform interpolation is given in Figure 6. Because the DWT of a block can be obtained from the dyadic wavelet transform of an image, the sub-pixel DWT coefficients of a block can be obtained from the sub-pixel dyadic wavelet coefficients.

3.2 Block-based Embedded Residual Coding

Dividing a frame into several blocks and encoding each block individually prevents an error in one block from propagating to other blocks. It also provides a flexible means of grouping several blocks into a high-level group of block (GOB) structure for robust transmission. To encode the blocks, we combine the SPIHT algorithm with a block-based motion residual approach [14]. Because a motion residual frame usually has significant components located at unrelated frequency subbands and spatial positions, SPIHT does not perform as well on encoding motion residual frames as on normal images. However, using SPIHT on wavelet bit-planes is simple and it performs better than using a run-length code on a wavelet bit-plane. We can improve the performance of our block-based SPIHT using a sub-optimal R-D approach.

3.2.1 Bit-plane-based R-D Optimization

We improve the coding efficiency of our block-based system using an R-D optimization approach. Our approach assigns optimal bit-plane numbers to each macroblock, and is simple and easily implemented. The block-based R-D function for a fixed bit rate is written as the constrained optimization problem

$$\min_{\{R_i\}} \sum_{i=1}^{N} D_i(R_i),$$

subject to

$$\sum_{i=1}^{N} R_i = R,$$

where N is the number of blocks and R is the given bit rate [6]. The optimal solution of the above problem can be obtained by solving the following un-constrained Lagrangian problem

$$\text{Ł}(R_1, R_2, \ldots, R_N, \lambda) = \sum_{i=1}^{N} D_i(R_i) + \lambda(\sum_{i=1}^{N} R_i - R).$$

By taking the derivative of the above formula with respect to R_i and λ, we have

$$\frac{\partial \text{Ł}}{\partial R_i} = \frac{\partial D_i(R_i)}{\partial R_i} + \lambda = 0 \tag{1}$$

$$\frac{\partial \text{Ł}}{\partial \lambda} = \sum_{i=1}^{N} R_i - R = 0. \tag{2}$$

The solutions R_i^* and λ^* of Equations 1 and 2 are

$$\lambda^* = -\frac{\partial D_1(R_1^*)}{\partial R_1} = -\frac{\partial D_2(R_2^*)}{\partial R_2} = \cdots = -\frac{\partial D_N(R_N^*)}{\partial R_N} \quad \text{and}$$

$$\sum_{i=1}^{N} R_i^* = R.$$

For a fixed rate block-based coder, optimality is achieved when all blocks are coded with the same R-D slope, and total bits assigned to all blocks equal the given bit budget of a residual frame. The computation and transmission costs of this optimal R-D search can be too high for low-delay and low bit rate applications. Using results for the optimal solution of a block-based R-D model, we propose a simple and fast block-based R-D optimization algorithm to improve coding performance. Instead of allocating an optimal number of bits to each block, our bit-plane-based R-D optimization procedure allocates the optimal number of bit-planes to each block. We only record the slope and the number of bits immediately after the SPIHT algorithm finishes encoding a bit-plane of a block. We take the absolute values of R-D slopes and use $\lambda_{i,j}$ and $r_{i,j}$ to respectively represent the absolute R-D slope and the bits spent after encoding the j-th bit-plane of the i-th block.

We sort the absolute R-D slope $\lambda_{i,j}$ in a decreasing order. From the beginning of the sorted list, we encode the bit-plane of the block that has the current absolute R-D slope until the total number of bits to encode these bit-planes exceeds the bit budget. Figure 7 shows an example of three coding blocks whose R-D slopes of each bit-plane has been calculated. The proposed algorithm is given in Appendix 2.

Although our R-D optimization process improves coding quality, it incurs some overhead because we must tell the decoder the status of each block. We must provide the most significant bit-plane (MSB) of each block to the decoder. We also need to tell the decoder when to stop decoding the current block. We encode several consecutive bit-planes in each block and use an end of block symbol to inform the decoder whether to decode the next bit-plane of the current block or the MSB of the next block.

3.2.2 Syntax and Semantics

Video syntax is arranged into a hierarchical structure of layers, Picture, GOB, Macroblock, and Block [5]. The syntax of our codec is compatible with H.263 in that the semantics of our Picture layer, GOB layer, and INTER/INTRA are identical to those of H.263, and the semantics in our Macroblock and Block layers are only slightly different from H.263. Figure 8 shows the simplified syntax of the H.263 baseline coder and our proposed coder; the semantics differences are highlighted.

In the Macroblock layer, the semantics of our COD and MB type and those of the H.263 baseline are the same. The COD, which is a single bit indicator, shows whether the current macroblock is coded or not. If the current macroblock is not coded, then the decoder treats the macroblock as an inter-predicted macroblock with a zero-valued motion vector and with no coefficient data. Otherwise, the macroblock is coded with motion vectors, followed by the coefficient data. The MB type informs the decoder of the coding type (INTER or INTRA) of the current macroblock.

The Macroblock layer semantic difference occurs between H.263 CBPY and our MSB. The H.263 syntax CBPY represents the Y-channel coded block pattern, for which at least one non-INTRADC transform coefficient is transmitted. We replace this function with our MSB, which informs a decoder the starting bit-plane to be decoded of each block. In the Block layer, the run-length code of quantized DCT coefficients is replaced by our bit-plane-based DWT coefficients.

4 Implementation and Performance Evaluation

The video sequences used in our experiments are Y-luminance signals of YUV color space. We use MEPG 4 test sequences in our experiments. The size of each sequence is in QCIF format, and the testing frame rate is ten frames per second. The first frame of a video sequence is an intra-frame (I-frame) encoded by DCT and all other frames are inter-frames (P-frames).

4.1 Implementation

Some implementation issues, such as rate control, filter selection, and motion vector precision, are evaluated and discussed in this subsection. We use the same rate control strategy as that in H.263 TMN 2.0. The difference between our rate control and TMN 2.0's is that we use number of bit-planes instead of quantization step size to control the actual number of bits allocated to a frame. We use the same formula in H.263 TMN 2.0 to calculate the number of bits assigned to each frame. Let R be the total bit budgets of all frames, F be the frame number to be encoded, r be the number of bits used to encode all previous frames, and f be the number of encoded frames. According to TMN 2.0, the remaining bits are distributed uniformly to the remaining frames; hence, the current frame will have bits equal to $\frac{R-r}{F-f}$.

Figure 9 shows the PSNR performance of various wavelet filters for different video sequences. The results indicate that the Haar filter gets better PSNR performance than bi-orthogonal filters $9-7$ and $5-3$, so we perform three levels of dyadic wavelet transform using the Haar filter as our wavelet in all experiments. Filter $9-7$ has been used by many research groups to encode the entire residual frame. This filter is not appropriate for a block-based system using a 16 by 16 macroblock for estimating motion vectors because its taps are too long for such a small block size.

Motion compensation is implemented based on macroblock segmentation. Both the horizontal and vertical search range is ± 15 Y-luminance pixels. Because the H.263 baseline can reach half integer motion vector precision, our motion vectors are restricted to half integer precision, and their component range is $[-16, 15.5]$. Our motion vectors are entropy-encoded like those of H.263. The performance improvement of using a half integer precision motion vector over that the integer precision motion vector in Haar wavelet coefficients is shown in Figure 10. It also compares the performances of half integer precision motion vector estimation in Haar wavelet coefficients and in an image domain. Because the performance improvement of using a half integer precision is significant, we use it for motion estimation in all experiments.

4.2 Performance Evaluation

Comparison studies in terms of Y-PSNR are performed between our method and both H.263 baseline and the wavelet coding system proposed in [13]. H.263 baseline uses block matching in spatial domain to perform motion estimation, the residual frame is encoded first by DCT, then by quantization and run-length coding modules. The wavelet-based method [13] uses a low-band-shift (LBS) method to perform motion estimation in the wavelet domain, and encodes the residual frame by a frame-based SPIHT algorithm. Both our method and the LBS method apply the same motion estimation scheme in the wavelet domain. However, in the LBS method, motion vector estimation can only reach integer precision. We use our interpolation scheme to improve the LBS method (and call it sub-pixel LBS) so that it can reach half integer motion vector precision.

Experiments show that our block-based wavelet codec outperforms sub-pixel LBS at low bit rates for all video sequences. Figure 11 shows the average PSNR of various bit rates for the two-second Sean and Foreman sequences. Both results show that our bit-plane-based R-D coding system achieves the best performance over the entire bit rate range, with about a $0.3 - 0.8$ dB gain over sub-pixel LBS.

However, the performance of our codec without R-D optimization on Foreman is worse than sub-pixel LBS. This is because a block-based system allows us to use a single bit (or COD) to represent a low residual error block. Thus, a considerable number of bits is saved when encoding these low residual error blocks. For a sequence with a complex motion structure such as Foreman, poor motion prediction results in significant residual errors in almost every block, and the the advantages of COD disappear. The Sean sequence has less apparent movements, and benefits from the block-based coding structure. This is supported in Figure 11(a) where the coding gain of using a block-based system on Sean exceeds sub-pixel LBS.

Many wavelet video studies experiment on relatively high bit rates, and they do not compare the Y-PSNR performance of their codecs with that of H.263. Because of the popularity of H.263, we examine and compare the performance of our codec with that of the H.263 baseline over a wide range of bit rates. Figure 12 shows the performance of coding six video sequences at various bit rates by our bit-plane-based R-D optimization algorithm, sub-pixel LBS, and H.263 baseline. Our coding scheme attains a better Y-PSNR than H.263 baseline at a higher bit rate. At lower bit rates, most results show that our PSNR is comparable to the PSNR of H.263 baseline. For sequences with relatively little motion, such as Akiyo and Sean, our method obtains an average 0.2 to 0.6 dB gain over the H.263 baseline at bit rates between 24 and 100 kbits/sec. For moderate-movement sequences, such as Container and News, and for sequences with a lot of movement, such as Foreman and Stefan, the performance difference between our coding scheme and H.263 baseline varies between -0.2 dB and 0.4 dB at very low bit rates. For all test sequences, the performance of the H.263 baseline at low bit rates is better than sub-pixel LBS. At a bit rate higher than 200 kbits/sec, both wavelet-based codecs have a coding gain of about 1 to 2 dB over H.263 baseline.

5 Conclusions

We propose an all wavelet codec that estimates sub-pixel motion vector in wavelet domain, uses bit-plane-based R-D optimization algorithm for residual encoding, and uses a syntax compatible with H.263. The popular H.263 decoder needs only modify a few semantics to decode our bit-streams. The performance of our codec is competitive with H.263 baseline at low bit rates, and is superior at higher bit rates. The performance of our block-based codec is better than the frame-based codec LBS at low bit rates, and these codecs have comparable performance at higher bit rates. The performance of our codec can be further improved if a wavelet-adapted rate control is applied.

References

[1] P. Y. Cheng, J. Li, and C.-C. Jay Kuo, "Rate control for embedded wavelet video coder," *IEEE Transactions on Circuits and Systems Vidoe Technology*, vol. 7, no. 4, pp. 696-702, August 1997.

[2] R. E. Van Dyck, T. G. Marshall, M. Chin, and N. Moayeri, "Wavelet Video Coding with Ladder Structures and Entropy-Cionstrained Quantization," *IEEE Transactions on Circuits and Systems Vidoe Technology*, vol. 6, no. 5, pp. 483-495, 1996.

[3] ISO/IEC 11172-2, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s," *MPEG (Moving Pictures Expert Group), International Organization for Standardization*, 1993.

[4] ISO/IEC 13818-2, "Generic coding of moving pictures and associated audio information," *MPEG (Moving Pictures Expert Group), International Organization for Standardization*, 1994.

[5] ITU-T, "ITU-T Recommendation H.263: Video coding for low bitrate communication," *The International Telecommunication Union*, 1996.

[6] N. S. Jayant and P. Noll, "Digital Coding of Waveforms," *Prentice-Hall*, 1984.

[7] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Applied and Computational Harmonic Analysis*, 10, 234-253, 2001.

[8] K. K. Lin and R. M. Gray, "Video residual coding using SPIHT and dependent optimization," *Data Compression Conference*, pp. 113-122, 2001.

[9] J. C. Liu and W. L. Hwang, "Active contour model using wavelet modulus for object segmentation and tracking in video sequences," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 1, no. 1, pp. 93-113, March 2003.

[10] J. Magarey and N. Kingsbury, "Motion estimation using a complex-valued wavelet transform," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1069-1084, April 1998.

[11] S. Mallat, "A wavelet tour of signal processing," *Academic Press*, 1998.

[12] S.A. Martucci, I. Sodagar, T. Chiang, and Y.-Q. Zhang, “A zerotree wavelet video coder,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 109-118, February 1997.

[13] H.-W. Park and H.-S. Kim, “Motion estimation using low-band-shift method for wavelet-based moving-picture coding,” *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 577-587, April 2000.

[14] A. Said and W.A. Pearlman, “A new, fast, and efficient image codec based on set portioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250, June 1996.

[15] K. Shen and E.J. Delp, “Wavelet based rate scalable video compression,” *IEEE Transactions on Circuits and Systems for VideoTechnology*, vol. 9, no. 1, pp. 109-122, February 1999.

[16] M.J. Shensa, “The discrete wavelet transform: Wedding the á trous and Mallat Algorithms,” *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 109-118, October 1992.

[17] G. J. Sullivan and T. Wiegand, “Rate-Distortion optimization for video compression,” *IEEE Signal Processing Magazine*, pp. 74-90, November 1998.

[18] D. Taubman and M. W. Marcellin, “JPEG2000,” *Kluwer Academic Publishers*, 2002.

[19] D. Taubman and A. Zakhor, “Multirate 3-D subband coding of video,” *IEEE Transactions on Image Processing*, vol. 3, pp. 572-588, September 1994.

[20] Q. Wang and M. Ghanbari, “Scalable coding of very high resolution video using the virtual tree,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 719-727, October 1997.

[21] Q. Wang and L. Wu, “Translation invariance and sampling theorem of wavelet,” *IEEE Transactions on Signal Processing*, vol. 48, no. 5, pp. 1471-1474, May 2000.

[22] A. B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor, “Visibility of wavelet quantization noise,” *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1164-1174, August 1997.

Appendix 1

The discrete wavelet transform (DWT) coefficients of a block can be obtained from dyadic wavelet transform coefficients. Although this property is used in [13], we re-state it in the following theorem.

Theorem 1 Let (x_0, y_0) be the top left corner of block b whose size is 2^J by 2^J in a 2^N by 2^N image f $(N > J)$. The DWT coefficient of b at (m, n) at scale 2^i and orientation $k = 0, 1, 2, 3$ is the dyadic wavelet coefficient of image f at position $(x_0 + 2^i m, y_0 + 2^i n)$ of the same scale and orientation. We use $k = 0$ to denote the result of applying the scaling function and $k = 1, 2, 3$ on different wavelets. That is,

$$(DWT)^k_{2^i} b(x_0 + m, y_0 + n) = \mathcal{W}^k_{2^i} f(x_0 + 2^i m, y_0 + 2^i n)$$

for $0 \leq m, n < 2^{J-i}$ for $i = 1, \cdots, J$. We use (DWT) to stand for DWT, and $\mathcal{W}^k$ for dyadic wavelet transform with orientation k. By our notation, $\mathcal{W}^0_{2^i} f(x, y) = \mathcal{A}_{2^i} f(x, y)$.

Proof

Let f' be the translation of the image f such that $f'(0, 0) = f(x_0, y_0)$. Figure 13 shows a block diagram proof of the theorem in which the identities of the first subgraph, the second subgraph, and the bottom subgraph are obtained from applying nobel identities to Figure 5. The output of the DWT at scale 2^i and orientation $k = 1, 2, 3$ at branch $D_{i,k}$ is

$$(DWT)^k_{2^i} b(m, n) = \mathcal{W}^k_{2^i} f'(2^i m, 2^i n), \tag{3}$$

and the output at branch A_i is

$$(DWT)^0_{2^i} b(m, n) = \mathcal{W}^0_{2^i} f'(2^i m, 2^i n). \tag{4}$$

The DWT coefficients at (m, n) of block b and the dyadic wavelet coefficients of f' at $(2^i m, 2^i n)$ of the same orientation and the same scale are equivalent. The dyadic wavelet transform is translation invariant; thus, we have for all k,

$$\mathcal{W}^k_{2^i} f'(2^i m, 2^i n) = \mathcal{W}^k_{2^i} f(x_0 + 2^i m, y_0 + 2^i n). \tag{5}$$

From Equations (3), (4), and (5), we obtain

$$(DWT)^k_{2^i} b(m, n) = \mathcal{W}^k_{2^i} f(x_0 + 2^i m, y_0 + 2^i n).$$

According to this theorem, the DWT coefficients of any block of an image can be obtained from the dyadic wavelet coefficients of the image.

Appendix 2

Bit-plane-based R-D Encoder

cal_BP_number: maximum number of bit-planes that can be encoded for each block.

CBP_num(i): the current number of encoded bit-planes at the i_{th} block. Initial value is zero for each block.

BP(i): i_{th} block's highest bit-plane.

- Step1

```
$for(i = 1, i \leq block\_number, i++)$
    $for(j = 1, j \leq cal\_BP\_number, j++)$
        $rate(j, i)$ = bit rate needed to encode $i_{th}$ block's $j_{th}$ bit plane.
        $\lambda(j, i)$ = calculate $i_{th}$ block's $j_{th}$ bit-plane's R-D slope.
    $end$
$end$
$while(bit\_encoded < bit\_budget)$
    $(j, i)$ = check $max(\lambda)$'s block number $i$ and bit-plane number $j$.
    $bit\_encoded = bit\_encoded + rate(j, i)$.
    $CBP\_num(i) = CBP\_num(i) + 1$.
    $\lambda(j, i) = 0$.
$end$
```

- Step2

```
$for(i = 1, i \leq block\_number, i++)$
    $if(CBP\_num(i))$
        $bit(i, BP(i))$ = encode $i_{th}$ block with $CBP\_num(i)$ bit-planes by block-based SPIHT.
    $end$
$end$
```

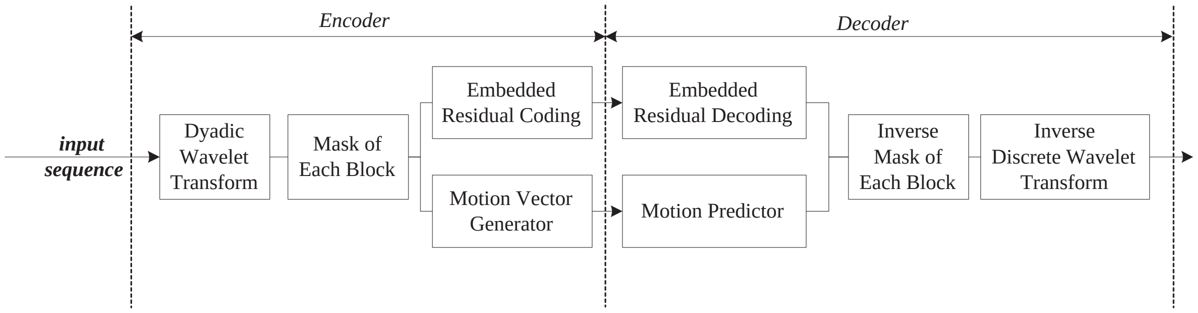


Figure 1: A mask can be used in our codec to modify wavelet coefficients. A sequence of masks can be inserted to generate a progressive motion vector by masking different subbands in different resolution scales.

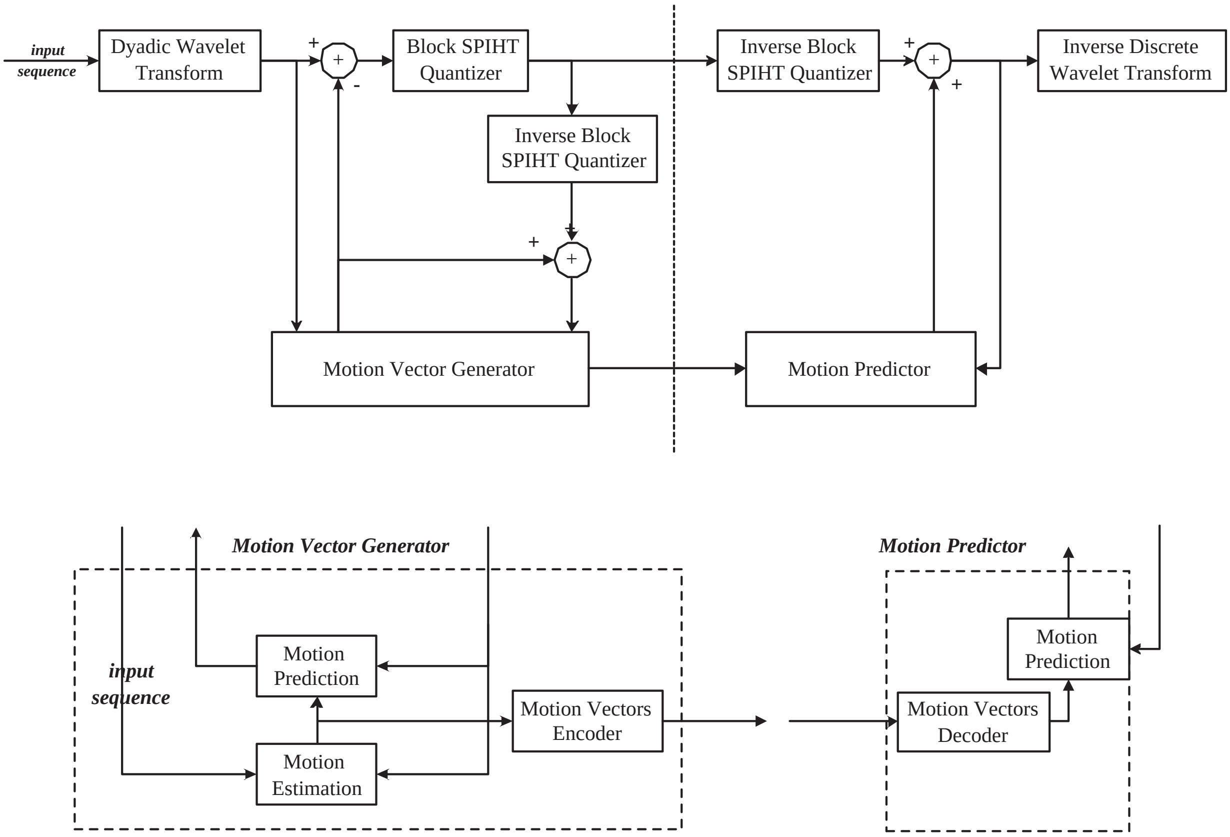


Figure 2: Block diagram of our proposed all wavelet video codec.

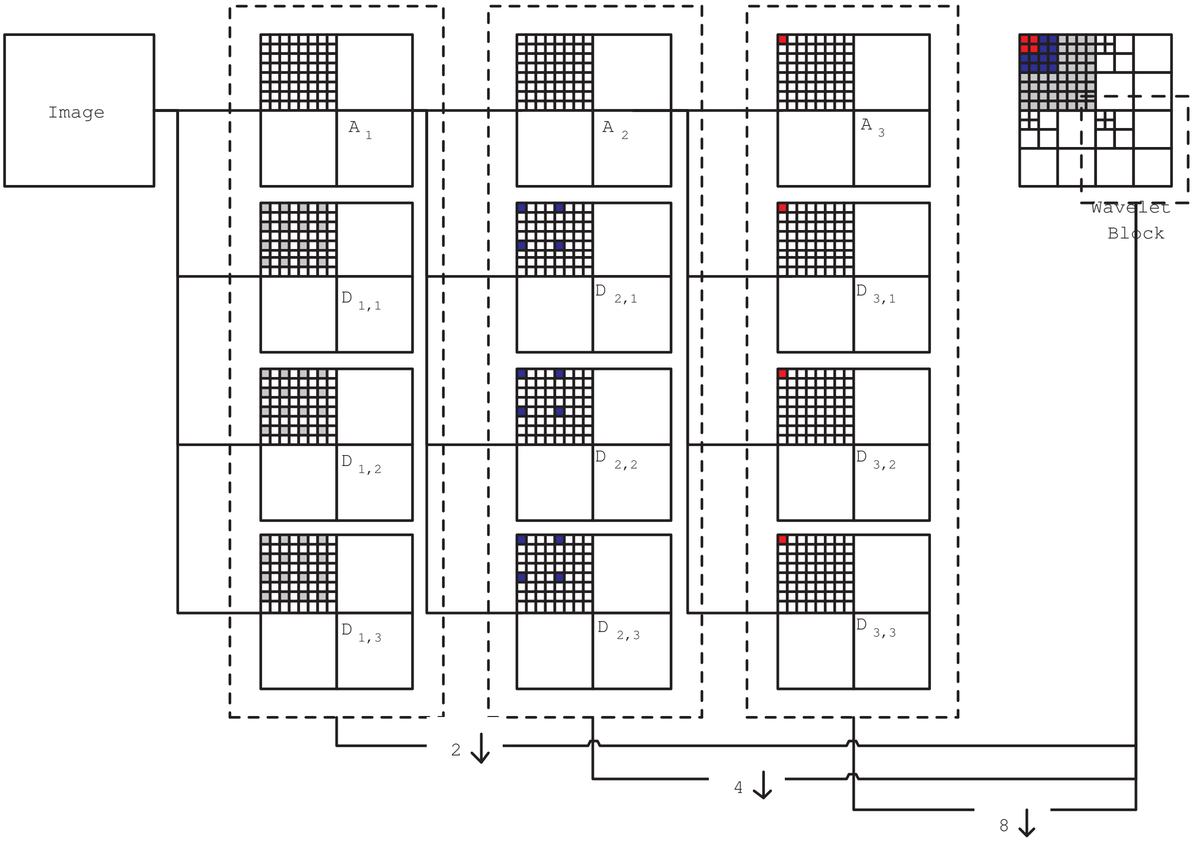


Figure 3: An example of generating a block-based DWT from dyadic wavelet coefficients.

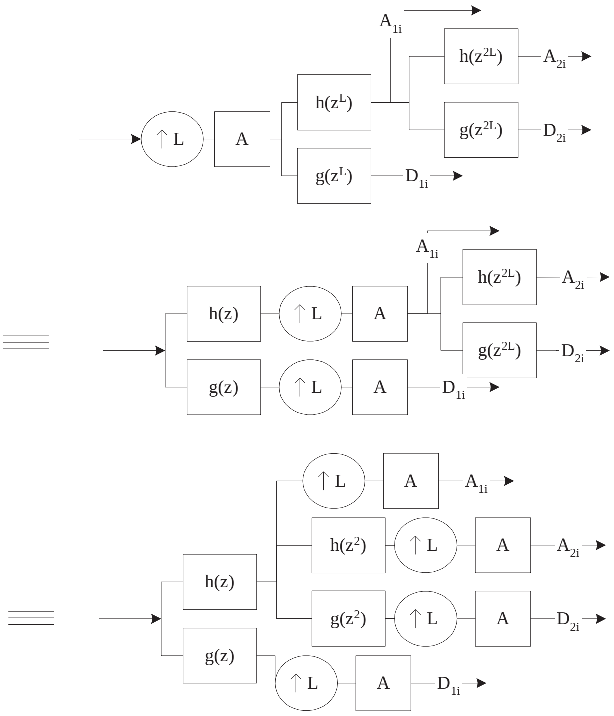


Figure 4: All three block diagrams are equivalent. Interpolation in an image domain with a filter A can be obtained by applying the same interpolation filter to dyadic wavelet coefficients in each subband. Our sub-pixel motion vectors are estimated using the bottom diagram.

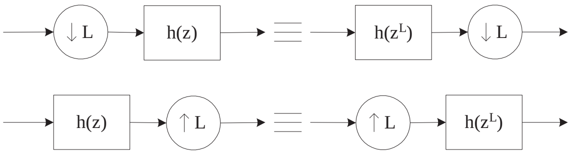


Figure 5: Noble identities used in Figures 4, 6, and 13.

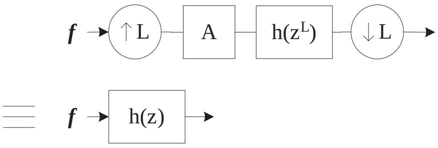


Figure 6: The dyadic wavelet coefficient at position mL of the interpolated image of f is equivalent to the dyadic wavelet coefficient at position m of the input image f.

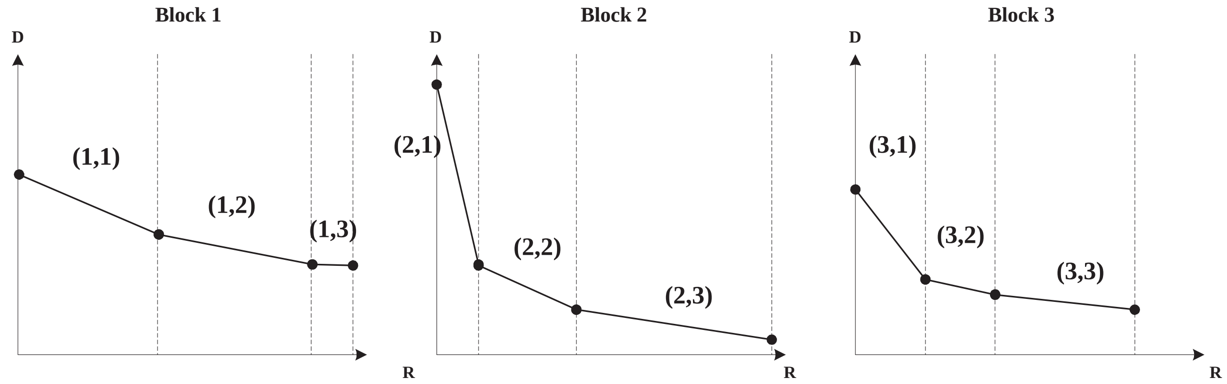


Figure 7: An example of the rate-distortion curves of three different blocks. The order of our bit-plane coding without R-D optimization is: $(2,1)$ $(1,1)$ $(3,1)$ $(1,2)$ $(1,3)$ $(2,2)$ $(3,2)$ $(3,3)$ $(2,3)$, and the order with our bit-plane-based R-D optimization is $(2,1)$ $(3,1)$ $(2,2)$ $(1,1)$ $(2,3)$ $(3,2)$ $(1,2)$ $(3,3)$ $(1,3)$.

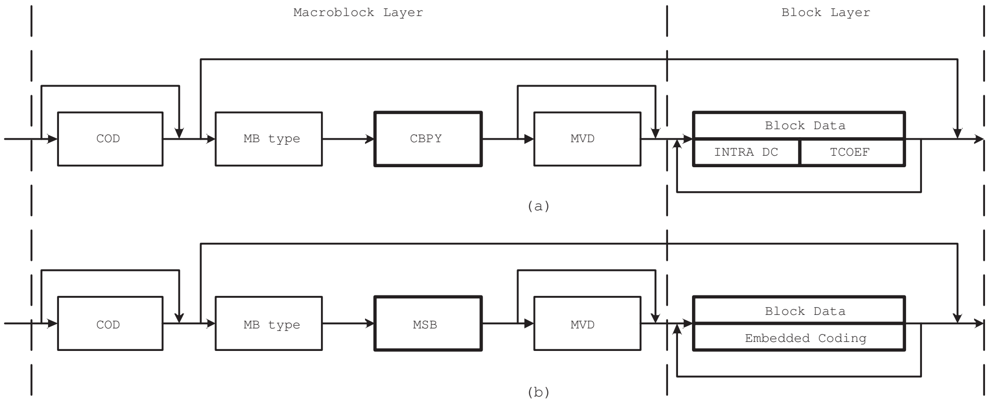
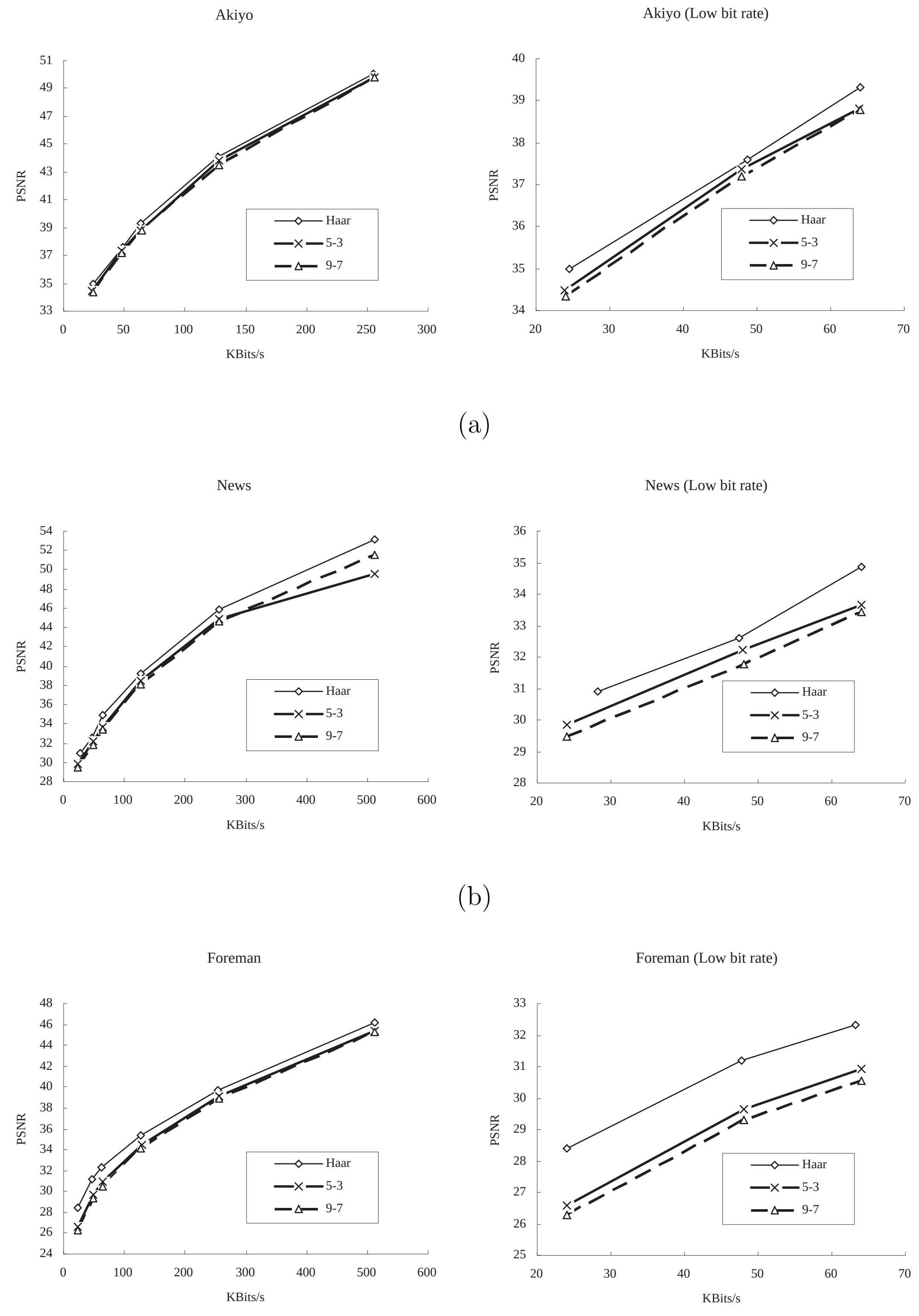


Figure 8: Syntax of the Macroblock and Block layers of (a) H.263 and (b) our proposed method. Bold boxes indicate the modules that have different semantics.

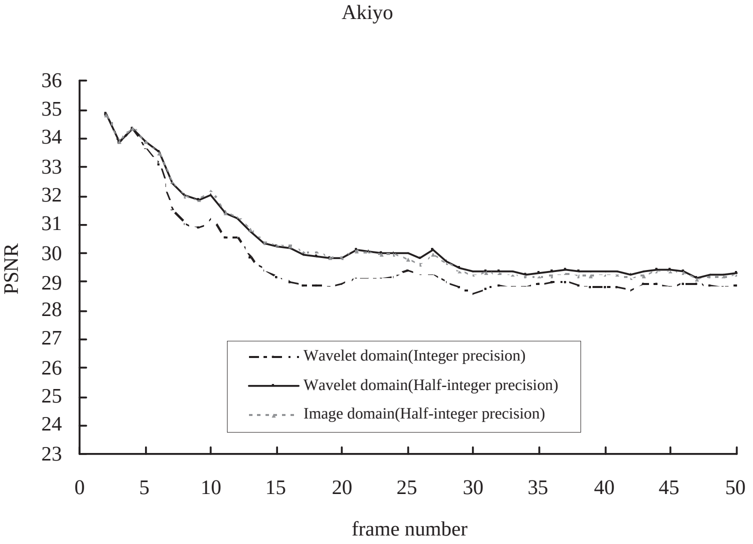


(a)

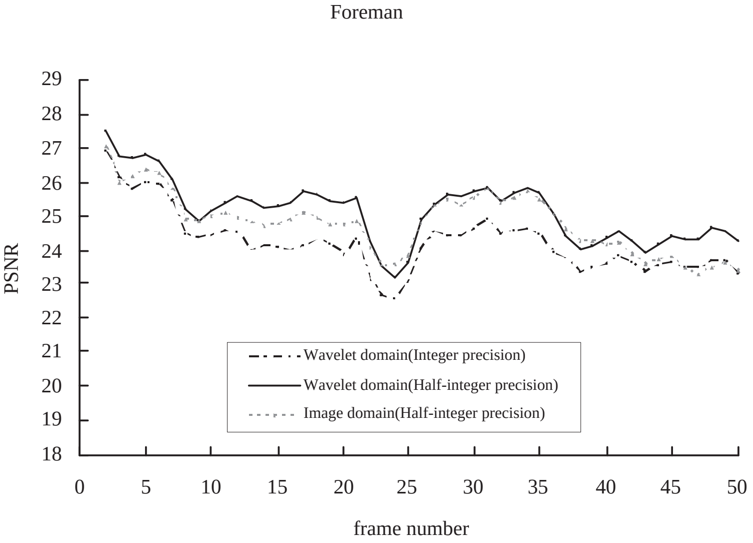
(b)

(c)

Figure 9: PSNR performance of various wavelet filters at appropriate bit rates are shown in the left column, and performance at low bit rates are shown in the right column. (a) Akiyo, (b) News, (c) Foreman. We use three levels of dyadic wavelet transform.

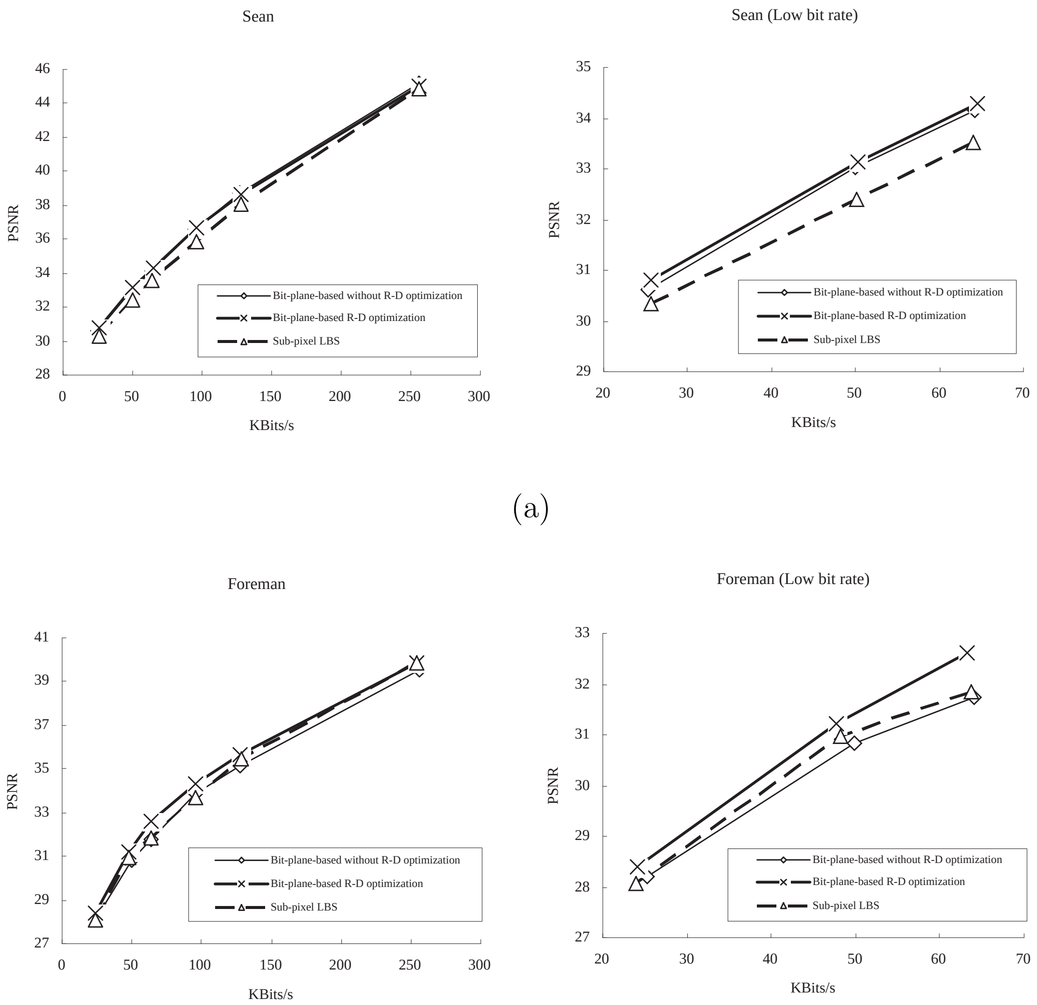


(a)



(b)

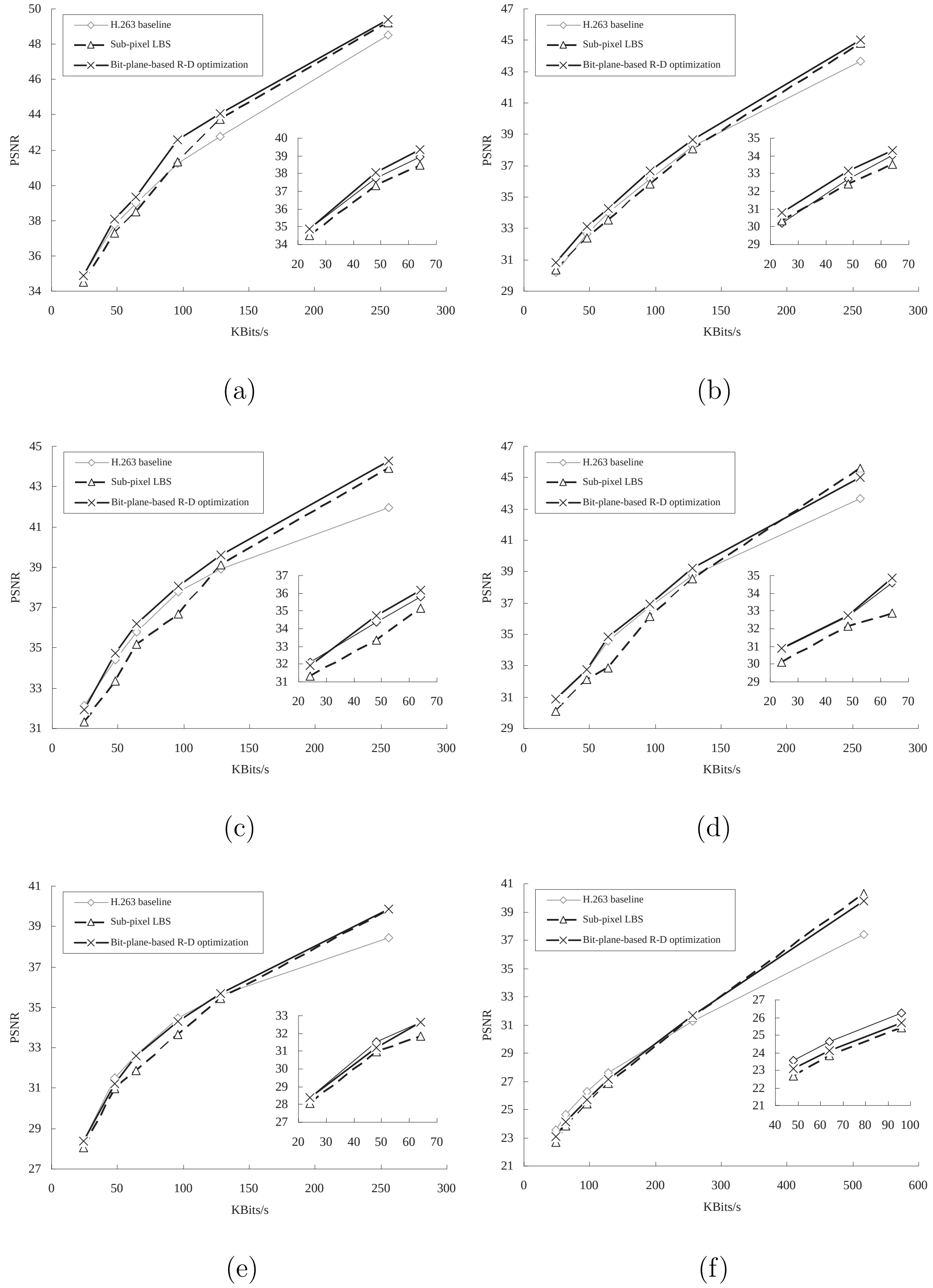
Figure 10: PSNR performance of motion vectors estimated from half integer precision of Haar wavelet coefficients, half integer precision of image domain, and integer precision of Haar wavelet coefficients. We use the bilinear interpolation for half integer motion vector estimations. (a) Akiyo, (b) Foreman.



(a)

(b)

Figure 11: Left: The average PSNR of various bit rates encoded by sub-pixel LBS, our bit-plane-based methods, one using and the other without using R-D optimization. Right: The average PSNR at low bit rates. (a) Sean, (b) Foreman.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 12: The average PSNR of various bit rates encoded by H.263 baseline, sub-pixel LBS, and our bit-plane-based R-D optimization algorithm. An insert of average PSNR at low bit rates is included in each subfigure. (a) Akiyo, (b) Sean, (c) Container, (d) News, (e) Foreman, (f) Stefan.

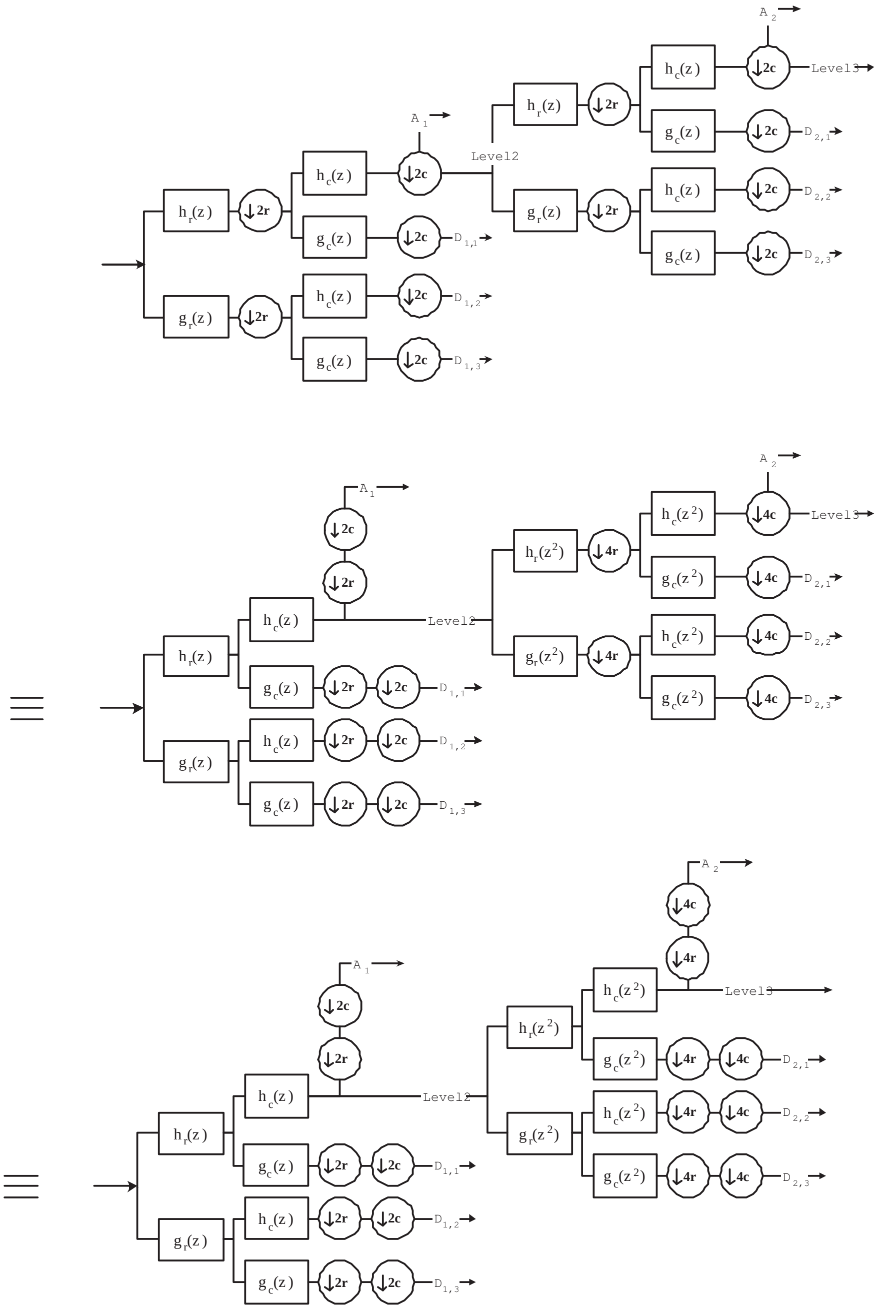


Figure 13: The schematic diagram proof of Theorem 1. A_i and $D_{i,k}$ are respectively the outputs at scale 2^i and orientation $k = 1, 2, 3$. The three block disgrams are equivalent. The bottom disgram is the filter bank structure of the algorithm á trous.