Significance–Linked Connected Component Analysis for Very Low Bit Rate Wavelet Video Coding

Jozef Vass *Bing-Bing Chai*[†] *Kannappan Palaniappan* *Xinhua Zhuang*

Multimedia Communications and Visualization Laboratory
Department of Computer Engineering & Computer Science
University of Missouri-Columbia
Columbia, MO 65211
{vass,zhuang}@cecs.missouri.edu

[†] *Sarnoff Corporation*
Princeton, NJ 08543
bchai@sarnoff.com

Abstract

In recent years, a tremendous success in wavelet image coding has been achieved. It is mainly attributed to innovative strategies for data organization and representation of wavelet-transformed images. However, there have been only a few successful attempts in wavelet video coding. The most successful one is perhaps Sarnoff Corporation's zerotree entropy (ZTE) video coder. In the paper, a novel hybrid wavelet video coding algorithm termed video significance-linked connected component analysis (VSLCCA) is developed for very low bit rate applications. It has also been empirically evidenced that wavelet transform combined with those innovative data organization and representation strategies can be an invaluable asset in very low bit rate video coding as long as motion-compensated error frames are ensured to be blocking-effect-free or coherent.

In the proposed VSLCCA codec, first, fine-tuned motion estimation based on H.263 Recommendation is developed to reduce temporal redundancy and exhaustive overlapped block motion compensation is utilized to ensure the coherency in motion-compensated error frames, second, wavelet transform is applied to each coherent motion-compensated error frame to attain global energy compaction, third, significant fields of wavelet-transformed error frame are organized and represented as significance-linked connected components so that both the within-subband clustering and cross-scale dependency are exploited, finally, the horizontal and vertical components of motion vectors are encoded separately using adaptive arithmetic coding while significant wavelet coefficients are encoded in bit-plane order by using high order Markov source modeling and adaptive arithmetic coding.

Experimental results on eight standard MPEG-4 test sequences show that for intraframe coding, on average the proposed codec exceeds H.263 and ZTE in PSNR by as much as 2.07 dB and 1.38 dB at 28 kbits, respectively. For entire sequence coding, VSLCCA is superior to H.263 and ZTE by 0.35 dB and 0.71 dB on average, respectively.

Submitted: IEEE Trans. on Circuits and Systems for Video Technology

Correspondence author: Dr. Xinhua Zhuang
Address: 201 Engineering Building West
Department of Computer Engineering & Computer Science
University of Missouri-Columbia
Columbia, MO 65211
Phone: (573) 882-2382
Fax: (573) 882-8318
E-mail: zhuang@cecs.missouri.edu
URL: http://meru.cecs.missouri.edu

1 Introduction

Very low bit rate video coding has triggered intensive research in both academia and industry. The adopted ITU-T H.263 Recommendation [1] offering a solution for very low bit rate videophony applications is the first standard to break the 64k bits-per-second (bps) barrier in audio-visual communications. It can be viewed as a modified and enhanced version of previous block-based video coding standards such as H.261 [2], MPEG-1 [3], and MPEG-2 [4] but specifically tailored to very low bit rate applications. The recently adopted MPEG-4 standard covers very low bit rate to medium bit rate multimedia communications. One of the functionalities of the emerging MPEG-4 standard is improved coding efficiency [5].

At very low bit rates, discrete cosine transform (DCT) based image coders suffer from blocking effect and mosquito noise. Subband coding schemes, also popularly used for progressive image transmission and browsing, offer a possible alternative to the block-based DCT. Not even mentioning more advanced subband coding schemes, the conventional ones have already yielded comparable objective performance to block-based coders and showed superior visual quality by eliminating the disturbing blocking artifacts. As for video coding, there have been three conceptually different ways of using wavelet transform reported in the literature:

- Using 3-dimensional (3-D) wavelet transform;
- wavelet transform of the original frames followed by motion estimation and motion compensation of wavelet coefficients; and
- traditional time domain motion estimation and motion compensation followed by wavelet transform of motion-compensated error frames.

The extension of 2-D subband image coding to include the time domain naturally leads to 3-D (temporal × spatial × spatial) subband video coding algorithms originally proposed in [6]. The advantages of 3-D wavelet video coding schemes include their low computational complexity and prevention of error propagation; the former is due to that computationally expensive time domain motion estimation and motion compensation are replaced by temporal filtering for which the Haar wavelet is usually used, i.e., in the time domain two subbands are obtained as the sum and difference of two consecutive frames, respectively, and the latter is due to that there is no recursive loop in the coder architecture as is the case with hybrid coders. These features make 3-D subband video coders an attractive tool for mobile communications [7]. Nevertheless, temporal filtering has not been as successful as time domain block-based motion estimation and motion compensation algorithms for exploiting the temporal redundancy inherent in video sequences in general, as evidenced by the reduction of coding gain for high motion sequences at low frame rates. Although this problem can be alleviated by using motion adaptation schemes [8, 9], 3-D subband video coding algorithms are mainly applicable for medium to high bit rate applications [8, 9, 10].

A subband domain multiresolution motion estimation and motion compensation scheme was introduced in [11]. After wavelet transform of each original frame, hierarchical block-based motion estimation and motion compensation are carried out, followed by encoding of significant motion-compensated wavelet coefficients. The proposed coder is well suited for medium and high bit rate applications such as HDTV and provides an easy conversion between different video coding standards. But subband decomposition is a *space variant* process [12]. Thus a translational motion between two consecutive frames may not be translated into a translational motion between two consecutive wavelet transformed frames. It was observed that the performance of multiresolution subband domain motion estimation and motion compensation deteriorated dramatically as the bit rate decreased.

In the third type of subband video coding algorithms, after time domain motion estimation and motion compensation, motion-compensated error frames are encoded in the wavelet domain. Superficially, the difference between this type of video coding algorithms and the traditional hybrid DCT video coding scheme is seen that wavelet transform replaces DCT in encoding of motion-compensated error frames. However, the replacement seems not to be working adequately with error frames if generated by nonoverlapped block motion compensation methods. It is understandable, that a global transform such as wavelet transform by no means tolerates localized blocking artifacts and thus its strength in terms of energy compaction can be severely degraded. Fortunately, the inconsistency could be largely alleviated by using overlapped block motion compensation (OBMC) technique [13, 14]. As was reported, OBMC not only mitigated the blocking effect but also reduced the overall energy of motion-compensated error frames. The majority of recently proposed very low bit rate wavelet video coding algorithms [15, 16, 17] were of this type and used OBMC.

In recent years, an impressive success of wavelet image coding has been achieved due to the use of innovative strategies for data organization and representation of wavelet-transformed images. There were four such wavelet image coders published in the literature. Shapiro's embedded zerotree wavelet (EZW) [18] coder and Said and Pearlman's set partitioning in hierarchical trees (SPIHT) [19] use the regular tree structure and the set-partitioned tree structure to approximate *insignificant* wavelet coefficients across subbands, respectively. Servetto *et al.*'s morphological representation of wavelet data (MRWD) [20] finds irregular-shaped clusters of *significant* coefficients within subbands. Chai *et al.*'s significance-linked connected component analysis (SLCCA) [21, 22, 23, 24, 25, 26] extends MRWD by exploiting both within-subband clustering of significant coefficients and cross-scale dependency of significant clusters. Among the above four wavelet image coding algorithms, SLCCA delivers the highest performance in general.

Despite success in still image coding, there have been only a few successful attempts in wavelet video coding. Bhutani and Pearlman proposed to use Shapiro's EZW algorithm to encode error frames obtained by recursive motion compensation [15]. Their coder showed superior performance when compared to the MPEG-1 standard. Kim and Pearlman proposed to extend SPIHT to 3-D subband video coding [27] and superior results to MPEG-2 were reported. Recently, Vass *et al.* applied the SLCCA data organization and representation strategy for low computational complexity highly scalable video coding [28]. In Sarnoff Corporation's zerotree entropy (ZTE) [16] video coder, after time domain block-based motion estimation and motion compensation similar to that of H.263, an EZW variant algorithm was proposed for the representation and encoding of motion-compensated error frames.

In the paper, a high performance hybrid wavelet video coding algorithm termed video significance-linked connected component analysis (VSLCCA) (partially presented in [29]) is developed for very low bit rate applications. As is empirically evidenced, the wavelet transform with the aids of those innovative data organization and representation methods can be an invaluable asset in very low bit rate video coding if motion-compensated error frames are ensured to be blocking-effect-free or coherent. In VSLCCA codec, first, *fine-tuned* time domain motion estimation based on H.263 Recommendation [1] is used to reduce temporal redundancy and *exhaustive* overlapped block motion compensation is utilized to ensure the coherency in motion-compensated error frames. Second, wavelet transform [30] is applied to each coherent motion-compensated error frame to attain global energy compaction. While the within-subband clustering property of wavelet decomposition is exploited by organizing and representing significant wavelet coefficients as *connected components* [31] obtained by morphological conditioned dilation operation [32], the cross-scale dependency of significant wavelet coefficients is exploited by the *significance-linkage* between clusters at different scales. Finally, motion vectors are encoded directly and significant wavelet coefficients are encoded in *bit-plane* order, both by using adaptive arithmetic coder [33] with space-variant high order Markov source modeling. Performance evaluation on several MPEG-4 test sequences shows that for intraframe coding, the proposed VSLCCA codec exceeds H.263 and ZTE in peak signal-to-noise ratio (PSNR) by as much as 2.07 dB and 1.38 dB at 28k bits (kb) on average, respectively. For entire sequence coding, VSLCCA is superior to H.263 and ZTE by 0.35 dB and 0.71 dB on average, respectively. The subjective advantage of VSLCCA over H.263 is also distinctive in that the disturbing blocking effects are entirely eliminated.

The rest of the paper is organized as follows. In the next section, the entire VSLCCA coding algorithm is presented in detail. Section 3 gives thorough performance evaluation in comparison to other state-of-the-art video coders. Finally, the last section concludes the paper.

2 VSLCCA Video Coding Algorithm

In this section, after reviewing the SLCCA data organization and representation strategy, the VSLCCA video coding technique is described, which in addition includes fine-tuned motion estimation, exhaustive overlapped block motion compensation, and adaptive arithmetic coding of motion information.

2.1 Image Coding

The main building blocks of the SLCCA image coding technique include:

- Multiresolution discrete wavelet transform and quantization;
- connected component analysis within subbands;
- posptocessing of significance map;
- significance-link registration across scales; and
- adaptive arithmetic coding with space-variant high order Markov source modeling.

2.1.1 Wavelet Transform and Quantization

Wavelet transform decomposes a signal into different frequency components, and then investigates each component with a resolution matched to its scale [34, 30]. The wavelet transform of a signal evolving in time depends on two variables: scale (or frequency) and time; wavelets provide a tool for time-frequency localization. Thus the wavelet transform represents an excellent alternative to Short Time Fourier Transform (STFT) [35], well suited to the analysis of non-stationary signals. Due to the use of short time windows at high frequencies and long time windows at low frequencies, the wavelet transform is able to maintain a constant relative bandwidth analysis. Since its extension to multidimensional signal analysis [36], it found significant applications in image and video coding recently [11, 16, 18, 19, 20, 25, 27]. The original "Lena" image and its corresponding three-scale wavelet decomposition are shown in Figs. 1a and 1b, respectively.

A wavelet coefficient c is called *significant* with respect to a predefined threshold T if its magnitude is larger than or equal to T, i.e., $|c| \geq T$; otherwise, it is deemed *insignificant*. An insignificant coefficient is also known as zero coefficient.

In SLCCA, all the wavelet coefficients are quantized with a single uniform scalar quantizer. The quantizer step size q is specified by the user and used to control the bit rate. All the wavelet coefficients that are quantized to non-zero are significant and to be transmitted. Since a uniform mid-step quantizer with double-spaced dead-zone is applied $T = q$.

This quantization choice might seem oversimplified, but as was evidenced by our experiments and also stated in [37], using more sophisticated quantization and optimization schemes such as optimal bit allocation among bands [38], optimal non-uniform scalar quantization [39], or vector quantization [40, 41] was not justifiable when the aforementioned advanced data representation and organization strategies were used since the performance gain, if any, was only marginal.

2.1.2 Connected Component Analysis

For natural images, the majority of pixels belong to either homogeneous regions or texture regions. Most of the energy of homogeneous and texture regions are compacted into the low frequency subband by the wavelet transform. By contrast, highly condensed energy around edge regions are compacted into high frequency subbands distributed over their small spatial neighborhoods. This signifies that most of the high frequency coefficients are clustered (around discontinuities), a phenomenon called *within-subband clustering* [20]. This within-subband clustering property of wavelet transformation is exploited by organizing wavelet coefficients into irregular-shaped clusters or *connected components* implemented by morphological conditioned dilation operation.

First, some basic binary morphological operations relevant to our application are reviewed. More detailed discussion of mathematical morphology can be found in [42, 31]. A binary image can be thought of a subset of $E \times E$, where E denotes the set of numbers used to index a row or column position on a binary image. Clearly, pixels are in this subset if and only if they have the binary value one on the image. The dilation of set $A \subseteq E^2$ with set $B \subseteq E^2$ is defined by $A \oplus B = \cup_{b \in B} A_b$, where A_b denotes the translation of A by b. For a structuring element that contains the origin, the dilation operation produces an enlarged set entirely containing the original set A. The *conditioned dilation* operation [32] will be used to recursively find a cluster in a set. Let $A \subseteq E^2$ denote the set wherein a cluster is to be sought. Let $M \subseteq A \subseteq E^2$ represent a subset of A to be used as a seed to be grown into a cluster in A. Then the dilation of M by the structuring element B with the origin being included conditioned on A defines the conditioned dilation:

$$D(M, B|A) = (M \oplus B) \cap A,$$

where the structuring element B also plays a role in controlling the density and geometric shape of the cluster. If the conditioned dilation is recursively applied, i.e.,

$$D^n(M, B|A) = D(D^{n-1}(M, B|A), B|A), \tag{1}$$

a cluster is formed as the recursion terminates, i.e., $D^n(M, B|A) = D^{n-1}(M, B|A)$.

After quantization of wavelet coefficients, a *significance map* which has the same size as the original image is defined:

$$A(x, y) = \begin{cases} 1 & \text{if the wavelet coefficient } c \text{ at location } (x, y) \text{ is significant,} \\ 0 & \text{otherwise.} \end{cases}$$

The conditioned dilation can be progressively used for segmentation of the *significance map* into within-subband significant clusters.

The segments generated by the conditioned dilation seem to fall into a more restrictive category of clusters named connected components which have been popularly used in *machine vision* for segmentation of a binary image over decades. A connected component is defined based on one of the three types of connectivity: 4-connected, 8-connected, or 6-connected geometric adjacency. Since the significant wavelet fields are only loosely clustered, the conventional definition of connected component using a strict geometric adjacency may produce too many components, affecting the coding efficiency. Thus, in SLCCA, we use symmetric structuring elements with a size larger than 3×3 square, but the segments generated by the conditioned dilation are still called *connected components* even if they may not be geometrically connected. Some structuring elements tested in our experiments are shown in Fig. 2. The ones in Figs. 2a and 2b generate 4- and 8-connectivity, respectively. The structuring elements in Figs. 2c and 2d represent a diamond of size 13 and a 5×5 square, respectively. These latter two may not preserve geometric connectivity but perform better than the former two in terms of coding efficiency.

To delineate a significant cluster, all zero coefficients within the neighborhood B of each significant coefficient in the cluster are labeled as the boundary of the cluster. The boundary information needs to be transmitted to the decoder. As the size of the structuring element increases, the number of connected components decreases and the number of boundary zero coefficients increases. The optimal choice of structuring element is determined by the comparative costs of encoding boundary zero coefficients vs. encoding the positioning information (seed position) of connected components.

Both the encoder and decoder must use the same structuring element, which is fixed during the entire coding process.

The conditioned dilation operation is used to recursively detect and transmit the significance map. At each step of the recursive conditioned dilation operation, the newly discovered significant and insignificant (boundary zero) coefficients are to be transmitted to the decoder after adaptive arithmetic coding as described in Section 2.1.5. Since both the encoder and decoder use the same structuring element and seed positions, the decoder can exactly replicate the operation of the encoder and thus construct the significance map.

2.1.3 Postprocessing of Significance Map

As extremely small clusters likely do not produce discernible visual effects but render a higher boundary-to-area ratio than large clusters, they are eliminated by using area thresholding to avoid their more expensive coding cost. As evidenced by several experiments, this area thresholding is quite practical to reach a higher coding gain without sacrificing the perceived image quality.

The connected component analysis and postprocessing are illustrated for the "Lena" image at 0.25 bits-per-pixel (bpp) in Fig. 3. The significance map is obtained by quantizing the wavelet coefficients with $q = 20.98$ uniform scalar quantizer. The 11797 wavelet coefficients are organized in 930 clusters by using diamond structuring element of Fig. 2c. After removing isolated wavelet coefficients (clusters with only one coefficient), the 11340 wavelet coefficients constitute 473 clusters (Fig. 3a). Fig. 3b shows the *transmitted* significance map which also includes boundary zero coefficients for delineating significant clusters. It is clear that only a small fraction of zero coefficients are to be transmitted.

2.1.4 Significance-Link Registration

Naturally, the seed position of each connected component must be available at the decoder. In the following, we will show how the cross-scale dependency property of wavelet transform can be exploited to reduce the cost caused by explicit transmission of the cluster positioning information. Naturally, seed positions that cannot be predicted must be explicitly encoded and transmitted.

By definition of [18, 43], relative to a given wavelet coefficient, all coefficients at finer scales of similar orientation which correspond to the same spatial location are called its *descendents*; accordingly, the given coefficient is called their *ancestor*. Specifically, the coefficient at the coarse scale is called the *parent* and all four coefficients corresponding to the same spatial location at the next finer scale of similar orientation are called *children*. This parent-child dependency for a three-scale wavelet decomposition is illustrated in Fig. 1c. Although the linear correlation between the values of parent and child wavelet coefficients has been empirically found to be extremely small as expected, there is likely additional dependency between the magnitudes of parent and children. Experiments showed that the correlation coefficient between the squared magnitude of a child and its parent tends to be between 0.2 and 0.6 with a strong concentration around 0.35 [18]. These properties of wavelet transformed images can be seen in Figs. 1b and 3. The cross-subband similarity among *insignificant coefficients* in wavelet pyramid has been exploited in EZW and SPIHT that greatly improves the coding efficiency. On the other hand, it is found that the spatial similarity in wavelet pyramid is not satisfied strictly, i.e., an insignificant parent does not warrant all four children to be insignificant. The "isolated zero" used in EZW indicates the failure of such a dependency. In SLCCA, as opposed to EZW and SPIHT, the spatial similarity among *significant coefficients* is exploited. However, SLCCA does not seek for the very strong parent-child dependency for each and every significant coefficient. Instead, it predicts the existence of clusters at finer scales. The fact that statistically the magnitudes of wavelet coefficients decay from a parent to its children [44] implies that in a cluster formed within a fine subband there likely exists a significant child whose parent at the coarser subband is also significant. In other words, a significant child can likely be traced back to its parent through this *significance linkage*. It is crucial to note that this significance linkage relies on a much looser spatial similarity.

Now, we define *significance-link* formally. Two connected components or clusters are called *significance-linked* if the significant parent belongs to one component, and at least one of its four children is significant and lies in another component. If the positioning information of the significant parent in the first component is available, the positioning information of the second component can be inferred through labeling the parent as having a significance-link. Since there are generally many significant coefficients in a connected component, the likelihood of finding significance-link between two connected components is fairly high. Apparently, labeling the significance-link costs much less than directly encoding the position, and a significant saving on encoding cluster positions is thus achieved.

The efficiency of the significance-linkage technique is evidenced by the significant reduction of explicit seed positions. For the "Lena" image at 0.25 bpp, the seed position of only 7 out of 473 clusters need to be explicitly transmitted which renders significant saving of the required bit budget.

2.1.5 Adaptive Arithmetic Coding of Significance Map and Significant Magnitudes

Usually, the last step of a coding algorithm is the entropy coding. The entropy coding techniques attempt to exploit the source statistics in order to generate an average codeword length closer to the source entropy for which, in SLCCA, adaptive arithmetic coding [33] is used. In contrast to a fixed arithmetic coder which works well for a stationary Markov source, an adaptive arithmetic coder updates the corresponding conditional probability estimation every time when the coder visits a particular context.

In SLCCA, both the significance map and the magnitudes of the significant coefficients in each subband are encoded by adaptive arithmetic coding. It is known that for the data stream generated by a nonstationary source such as natural images, the conditional probabilities may vary substantially from one section to another. The knowledge of the local probability distributions acquired by an adaptive model is more robust than the global estimates and follows well the local statistical variation. To exploit the full strength of adaptive arithmetic coding, it is preferable to organize the outcomes of a nonstationary Markov source into such a data stream that each local probability distribution is in favor of one symbol. The well known lossless *bit-plane* encoding is built upon the above idea. In SLCCA, the magnitude of a significant coefficient in each cluster in a subband is converted into a fixed-length binary representation and encoded in bit-plane order, where the length is determined by the maximum magnitude in the subband. Generally, most of magnitudes in any cluster in a subband are smaller than the maximum magnitude in the subband, implying that the more significant bit-planes would contain significantly more 0's than 1's. Accordingly, the adaptive arithmetic coder would generate more accurate local probability distributions in which the conditional probabilities for "0" symbols are close to one for more significant bit-planes. In the following, the intersection of a cluster in the subband with a bit-plane is called a *cluster-section*.

The context used to define the conditional probability models at each pixel is related to the status of significance of its eight neighbors and its parent as well. As is shown in Fig. 4a, the number of significant coefficients in the 8-connected neighborhood of a given pixel yields nine possible models. As is shown in Fig. 4b, the significance status of the parent is also used in determining the final context resulting in a total of $9 \times 2 = 18$ possible models. Those 18 contexts are used to define conditional probabilities needed for adaptive arithmetic coding of both the significance map in the subband and the cluster-sections in every bit-plane in the subband.

First, four symbols are used to encode the significance map in the subband. POS and NEG symbols are used to represent the sign of positive and negative significant coefficients, respectively, ZERO symbol is used to label insignificant boundary pixels, and a special SL symbol is used to indicate a significant pixel in a cluster has been assigned a significance-link. All four children of the SL pixel belong to a new cluster or its boundary at finer scale and at least one child must be inside the new cluster. To determine the context around a pixel in the significance map or its boundary, only the number of those significant neighbors which are already transmitted is counted.

Second, the significant magnitudes in the subband are encoded in bit-plane order. In each bit-plane, cluster-sections are encoded following the same order that the clusters in the subband are detected by the previously described conditioned dilation operation. Apparently, only two symbols, i.e., "0" and "1", are needed. This time with no change the originally defined 18 contexts are used to adaptively calculate the conditional probabilities to be used in adaptive arithmetic coding.

The distribution of the bit budget for the "Lena" image at 0.25 bpp is as follows. As is shown in Table 1, 11 bytes are required to specify the 7 seed positions needed to be transmitted explicitly. The majority of the bitstream (4989 bytes) is spent on transmitting the significance map, which includes implicit seed positions (SL symbol), the sign of significant coefficients (POS and NEG symbols), and the boundary zero coefficients (ZERO symbol). Finally, 3192 bytes are spent on specifying the magnitude of significant wavelet coefficients.

Timing results of both the encoder and decoder of SLCCA for the "Lena" image at 0.25 bpp executing on one 195 MHz R10000 CPU of an SGI Octane workstation are shown in Table 2. As there is no optimization involved, both the decoder and encoder has approximately equal low computational complexity being comparable with that of zerotree algorithms.

2.2 Video Coding

The block diagram of the proposed VSLCCA video coding algorithm is shown in Fig. 5. As is seen, the SLCCA data organization and representation technique is embedded in the VSLCCA video coding scheme. In addition, VSLCCA include:

- Motion estimation;
- motion compensation; and
- adaptive arithmetic coding of motion information.

Fine-tuned block-based motion estimation following the spirit of H.263 Recommendation is used to reduce temporal redundancy. Zero, one, or four motion vectors per macroblock are determined by using full search block matching algorithm with half pixel refinement. Then, *exhaustive* overlapped block motion compensation [13] is used to reduce the artificial blocking effect caused by block-based motion estimation. Each predicted block in the current frame is formed as a weighted sum of as many as nine blocks from the previous reconstructed frame which are determined by translating the current block using the motion vectors associated with the current block and its eight neighboring blocks. This fine-tuned motion estimation followed by exhaustive overlapped block motion compensation results in a coherent motion-compensated error frame without artificial block boundaries, for which the wavelet transform can be efficiently applied to compact the frame energy into few significant coefficients.

After wavelet transform of motion-compensated error frame, all the wavelet coefficients are scalarly quantized. SLCCA algorithm is then utilized to organize and represent significant wavelet coefficients as significance-linked connected components to exploit both the within-subband clustering and cross-scale dependency.

Finally, adaptive arithmetic coding is used for the direct encoding of motion vectors and bit-plane encoding of significant coefficients modeled as a space-variant high order Markov source.

2.2.1 Motion Estimation

The original frame is divided into non-overlapping 16×16 *macroblocks*. As in H.263 Recommendation [1], each macroblock may have zero, one or four motion vectors. Full search block matching algorithm with integer pixel resolution is used on the luminance component to determine one motion vector per macroblock using *mean-squared error* (MSE) criterion. In the current implementation, the previous reconstructed (instead of original) frame is used as reference since it provides better performance. The search range is ±15 pixels in both vertical and horizontal directions. As is well known, the performance of block matching can be substantially improved by using subpixel resolution. As was specified in the H.263 Recommendation, the initial integer motion vectors can be refined to half pixel resolution in both directions in terms of computationally inexpensive bilinear interpolation and, again, the previous reconstructed frame is used here as reference. Each macroblock is then split into four 8×8 *blocks* and one motion vector per block is searched and refined to half pixel resolution by using just the same procedure. As is seen, the four block motion vectors are determined independent (as opposed to H.263 test model [45]) of the motion vector of the macroblock.

Afterwards, zero motion vector per macroblock is decided when $\mathrm{MSE}_{\mathrm{zero}} < \mathrm{MSE}_{\mathrm{one}} + T_{null}$, where $\mathrm{MSE}_{\mathrm{zero}}$ and $\mathrm{MSE}_{\mathrm{one}}$ are the resulting MSE of the macroblock by using zero motion vector and one motion vector per macroblock, respectively, and T_{null} is a specified *null margin*. When this condition is not satisfied and $\mathrm{MSE}_{\mathrm{four}} + T_{split} < \mathrm{MSE}_{\mathrm{one}}$, four motion vectors per macroblock are used, where $\mathrm{MSE}_{\mathrm{four}}$ denotes the macroblock MSE when four motion vectors per macroblock are used, and T_{split} is a predefined *split margin*. Otherwise, one motion vector per macroblock is used. Naturally, the number of motion vectors used for each macroblock needs to be transmitted to the decoder as side information

By increasing the null margin T_{null}, the number of macroblocks with zero motion vector increases. This results in a reduction of bandwidth spent on transmission of motion vectors and an increase of motion-compensated error frame energy. By reducing the value of split margin T_{split}, four motion vectors over one motion vector per macroblock are favored resulting in an increase of motion vector information and a decrease of motion prediction error energy. By using several test image sequences, it has been experimentally found that $T_{split} \approx 6 \cdot T_{null}$ matches well with the *exhaustive* OBMC algorithm to produce coherent motion-compensated error frames with a significant reduction of MSE.

2.2.2 Motion Compensation

Blocking-effect-free coherent motion compensation is crucial to the success of VSLCCA. In DCT-based hybrid video coding algorithms, the effectiveness of DCT is not significantly degraded by artificial blocking effect introduced by block-based motion estimation and motion compensation due to the fact that the motion block boundaries are well aligned with the DCT block boundaries. However, in the case of a global transform such as wavelet transform, the introduced artificial blocking effect (discontinuities) may generate quite a lot spurious high frequency components and thus the effectiveness of wavelet transform in terms of its energy compaction is significantly degraded. As a remedy, in VSLCCA *exhaustive* overlapped block motion compensation is used to alleviate the blocking effect of motion-compensated error frames. OBMC not only provides a coherent motion-compensated error frame but also decreases the motion-compensated prediction error. Therefore, a simple OBMC algorithm is included as part of H.263 Recommendation.

The operation of the *exhaustive* OBMC algorithm as applied in VSLCCA is illustrated in Fig. 6. The frame is divided into non-overlapping blocks of 8×8 pixels and one motion vector per block is assigned; that is, when one motion vector per macroblock is decided in motion estimation, that motion vector is replicated for each of the four constituent blocks. Each predicted block is composed as the weighted sum of as many as nine blocks from the previous reconstructed frame determined by translating the current block by using the nine motion vectors assigned to the current block and its eight neighboring blocks. Performance evaluation on several test image sequences shows that *raised cosine* window with 4 pixels overlap (Fig. 7) is a good choice for weighting.

2.2.3 Effectiveness of Proposed Motion Estimation and Compensation Techniques

The effectiveness of the applied motion estimation and motion compensation schemes is evidenced by a significant reduction of MSE of the motion-compensated error frames and a thorough elimination of the blocking effect as follows. For the "Foreman" sequence at 48 kbps, H.263 test model spends 982 bits-per-frame (bpf) on average for motion vectors while the averaged MSE of the motion-compensated error frames is 90.06. On the other hand, the applied motion estimation and motion compensation techniques spend 1704 bpf for motion vector information but with a significant reduction of the averaged MSE of the motion-compensated error frames to 72.76. As is clearly shown in Fig. 8, the 156th motion-compensated error frame of the "Foreman" sequence produced by H.263 test model still suffers from blocking effect while the error frame by the adopted technique is free of blocking artifacts.

This bit budget increase is due to two reasons. First, as is shown in Fig. 9, in VSLCCA, larger portion of the macroblocks is coded by using four motion vectors than in H.263 test model. Second, unlike in H.263, in VSLCCA, the four motion vectors of each macroblock are independently determined from the initial one motion vector of the macroblock with the same search range, which requires more bits for the encoding. However, this motion vector bandwidth increase is inevitable in order to ensure the coherency of motion-compensated error frames and well compensated by the reduction of bandwidth spent on encoding of significant wavelet coefficients of motion-compensated error frames.

The coding mode selection (zero, one, or four motion vectors per macroblock) for VSLCCA and H.263 for the "Foreman" sequence sampled at 5 frame-per-second (fps) is shown in Fig. 9. As is seen, there are two major differences between VSLCCA and H.263. First, in VSLCCA, the frequency of macroblocks with four motion vectors is approximately two-to-three times higher than in H.263. Second, as opposed to H.263, the probability of macroblocks with four motion vectors decreases as the bit rate increases. The explanation of this latter phenomena is the followings. In H.263 and its optimized mode selection algorithm [46], as the bit rate decreases more bits are spent on the encoding of motion-compensated error frames than motion information, which increases the objective performance. However, as is well-known, as the bit rate decreases more blocking artifacts are introduced, which cannot be tolerated in VSLCCA, where due to the global wavelet transform maintaining the coherency of motion-compensated error frames is of key importance. Thus at low bit rates, more accurate motion estimation is required in order to prevent the more apparent blocking effects of the block-based motion estimation and motion compensation schemes. At higher bit rates, where the difference between the reference and motion-compensated error frame decreases, one motion vector per macroblock yields satisfactory performance.

The proposed motion estimation and motion compensation technique moderately increases the computational complexity of both the encoder and decoder. Since H.263, ZTE, and VSLCCA all use full search block matching algorithm with subpixel refinement, the increase of computational complexity of motion estimation is due to that in VSLCCA the four motion vectors per macroblock are determined with ±15 pixels search range, where in H.263 test model and ZTE only few pixel refinement is used. Since the computational complexity of full search block matching algorithm is quadraticly proportional to both the search range and block size, the computational complexity of VSLCCA motion estimation scheme is at most twice as much as that of H.263. The increase of computational complexity of motion compensation is due to that in VSLCCA nine blocks are used to determine each predicted block, where in H.263 and ZTE only three blocks are used. Furthermore, VSLCCA uses raised cosine window weighting function implemented with floating point arithmetic compared to the integer implementation of H.263. Computer experiments show that while H.263 spends 99.7 ms for motion compensation per frame on average, VSLCCA requires 208.9 ms.

2.2.4 Adaptive Arithmetic Coding of Motion Vector Information

As was mentioned before, each macroblock may have zero, one, or four motion vectors being associated. The number of motion vectors per macroblock is encoded by using adaptive arithmetic coding with a single model of three symbols and transmitted to the decoder as side information. As in H.263, motion vector components are encoded separately, i.e., a different adaptive model is used for the vertical and horizontal components, respectively. In each model, each possible value of components is represented using a different symbol. Since the motion vector range is ±15 pixels with half pixel resolution, a total of 64 symbols are needed to encode each component of the motion vector.

Note, that as opposed to H.263, motion vector prediction is not applied in VSLCCA. In H.263, each motion vector component is separately predicted by the median of already transmitted components of neighboring (left, above, and right) macroblocks. This is necessary due to the *non-adaptive* variable length coding used for motion vector coding in H.263 (without Annex E). In VSLCCA, however, the adaptive arithmetic coder well exploits the local statistics of motion vector components, thereby making motion vector prediction unnecessary.

2.3 Justification of VSLCCA Algorithm

All the previously mentioned four top wavelet image coding algorithms can be applied for video coding. In this section, we show that SLCCA is more applicable for the encoding of motion-compensated error frames generated by the proposed fine-tuned motion estimation and overlapped block motion compensation algorithms than zerotree-like algorithms.

Ideally, motion compensation should result in zero residual error. However, lack of a good match from the previous frame usually results in large error magnitudes. This includes [47] 1) new object is coming into the scene, i.e., pixels belonging to the new object do not have a good match in the previous frame; 2) uncovered background areas, i.e., pixels belonging to these areas do not have a good match in the previous frame, where they were covered by the object; and 3) moving texture areas since texture areas have a high intensity variance and an even small deviation from the true motion can yield large magnitudes.

As a result, motion-compensated error frames have very different statistics than natural images. Their histogram can be well-modeled by a generalized Gaussian distribution, and the correlation between pixels is very low in comparison to natural images [48]. Furthermore, motion-compensated error frames show both line structure belonging to the edges of moving components [49] and texture structure belonging to moving texture regions. This is clearly visible in Figs. 10a and 10b depicting the 114th original and motion-compensated error frames of the "Foreman" sequence, respectively. The line structure and textureness of the motion-compensated error frame are also clearly recognizable in the wavelet-transformed error frame (Fig. 10c).

When texture- and edge-rich frames are encountered, wavelet transform is unlikely to give large zero regions due the lack of large homogeneous regions. Thus the advantage of using the zerotree structure as in EZW, or set-partitioned zerotree structure as in SPIHT is weakened. On the other hand, SLCCA uses significance-based clustering and significance-based between cluster linkage, which is not affected by the existence of texture and line structure.

In very low bit rate video coding, motion-compensated error frames are to be highly compressed. As is experimentally evidenced in [25], SLCCA is more applicable for very low bit rate coding than zerotree-like algorithms, i.e., the objective performance measured by PSNR between SLCCA and SPIHT increases as the bit rate decreases. This is also empirically justified in Table 3, where the performance of SLCCA and SPIHT is compared on the 114th motion-compensated frame of the "Foreman" sequence shown in Fig. 10b. As seen, at 0.0625 bpp, SLCCA outperforms SPIHT by as much as 0.43 dB in PSNR with an average increase of 0.23 dB.

3 Coding Results

The performance comparison of different video coding algorithms is fairly difficult. One cause is that the MPEG-4 test sequences were distributed in original ITU-T 601 format and everybody was welcome to use his or her own format conversion. Another cause is the wealth of different rate control algorithms. To ensure a fair performance comparison among H.263, ZTE, and VSLCCA, the test sequences used in VSLCCA are the same as in [16] and there is no rate control being applied in both VSLCCA and H.263, instead, all the frames have been quantized with the same uniform scalar quantizer with the step size being used to adjust the final bit rate.

In VSLCCA, motion estimation is performed only on the luminance component. For motion compensation of the chrominance components the corresponding luminance motion vectors are divided by two, and OBMC with block size 4×4 and raised cosine window function with two pixels overlap is applied. Then, four and three scale dyadic wavelet decomposition is carried out on the luminance and chrominance components, respectively, by using 9/7 biorthogonal filter bank [41]. Both the luminance and chrominance components are quantized with the same uniform scalar quantizer. In all the experiments, 5×5 square structuring element shown in Fig. 2d is used and clusters having less than three significant coefficients are removed. The objective performance is measured by PSNR defined as

$$\text{PSNR [dB]} = 20 \log_{10} \frac{255}{\text{RMSE}},$$

where RMSE is the root mean-squared error between the original and reconstructed frames. All the reported results (bit rates and PSNR performance) are computed from the decoded bitstream.

Performance comparison is carried out on eight standard MPEG-4 test sequences (four Class A sequences: "Akiyo," "Container Ship," "Hall Monitor," and "Mother & Daughter;" and four Class B sequences: "Coast Guard," "Foreman," "News," and "Silent Voice") in QCIF resolution.

Due to their prime importance, first intraframe coding results are given. Then, coding results of the entire sequences (including first intraframe followed by interframes) are presented.

3.1 Intraframe Coding

The intraframe coding comparison is done on the first frame of all the eight test sequences at 14 kb (0.55 bpp) and 28 kb (1.10 bpp). First, H.263 was run, then the quantizer step size was adjusted in VSLCCA to exactly match the bit rate obtained by H.263. The results of H.263, ZTE and VSLCCA are summarized in Tables 4 and 5 at the corresponding two bit rates. As shown in Table 4, at 14 kb for the luminance component, VSLCCA outperforms H.263 by 1.79 dB on average, and also exceeds ZTE ranging from 0.64–1.39 dB. At 28 kb, the difference between VSLCCA and H.263 increases, i.e., VSLCCA exceeds H.263 by 1.31–3.25 dB. At the same bit rate VSLCCA is also superior to ZTE by 1.38 dB on average. For the chrominance components at 14 kb, VSLCCA is superior to H.263 by 1.27 dB or 1.09 dB on average for the U or V components, respectively. At 28 kb, VSLCCA outperforms H.263 by 1.62 dB or 1.44 dB on average for the U or V components, respectively. For the averaged chrominance components, VSLCCA also exceeds ZTE by 0.46–2.10 dB or 1.23–2.23 dB at 14 kb or 28 kb, respectively.

The coding results for the "Akiyo" and "Foreman" sequences are shown in Figs. 11 and 12, respectively. Both figures include the original image and the reconstructed images from H.263 and VSLCCA at 14 kb and 28 kb. At the bit rate as low as 14 kb, the visual advantage of VSLCCA over H.263 is distinctive in both images by the elimination of the annoying blocking artifacts of H.263. As the bit rate increases to 28 kb, PSNR attained by both the algorithms in both images is quite high and the visual quality of the two algorithms becomes more compatible even though VSLCCA maintains much higher PSNR.

3.2 Entire Sequence Coding

For interframe coding comparison, for H.263 both the unrestricted motion vector mode and advanced prediction mode are used (Annexes D and F, respectively). Class A sequences are sampled at frame rate 5 fps and encoded at bit rate 10 kbps, or sampled at 10 fps and encoded at 24 kbps. Class B sequences are sampled at 7.5 fps and encoded at 48 kbps, or sampled at 15 fps and encoded at 112 kbps. The coding results are summarized in Tables 6–9. First H.263 was run, and then in VSLCCA the quantizer step size was adjusted to match the bit rate of H.263.

For the luminance component, for Class A test sequences at 5 fps and 10 kbps VSLCCA is superior to H.263 and ZTE by 0.73 dB and 1.10 dB on average, respectively. At 10 fps and 24 kbps, VSLCCA is superior to H.263 and ZTE by 0.35 dB and 0.99 dB, respectively. For Class B test sequences, for the luminance component at 7.5 fps and 48 kbps VSLCCA still outperforms H.263 by 0.23 dB on average. However, as the bit rate increases to 112 kbps at frame rate 15 fps, the two coding algorithms in terms of objective performance are compatible. When compared to ZTE, VSLCCA is superior by 0.24–0.62 dB or 0.03–0.53 dB at 7.5 fps and 48 kbps or at 15 fps and 112 kbps, respectively.

For the chrominance components, for Class A test sequences at 5 fps and 10 kbps VSLCCA outperforms H.263 by 0.95 dB and 0.50 dB on average for the U and V components, respectively. At 10 fps and 24 kbps for the averaged chrominance components VSLCCA is superior to H.263 by 0.32 dB on average. For Class B test sequences the difference between VSLCCA and H.263 decreases, at 7.5 fps and 48 kbps VSLCCA exceeds H.263 by 0.08 dB and 0.13 dB on average for the U and V components, respectively. At 15 fps and 112 kbps for the U component the performance of the two algorithms are compatible while for the V component H.263 performs slightly better by 0.07 dB on average.

The coding results for the 54th frame of the "Akiyo" and 160th frame of the "Foreman" sequences are shown in Figs. 13 and 14, respectively. The "Akiyo" sequence is coded at 5 fps, 10 kbps, and the "Foreman" sequence is coded at 7.5 fps, 48 kbps. Both figures include the original, and decoded images from VSLCCA and H.263. The superior visual quality of VSLCCA is clearly visible in both sequences in the elimination of both mosquito noise and blocking artifacts of H.263. The frame-by-frame luminance PSNR comparison between H.263 and VSLCCA for the above two video sequences at the corresponding bit rates are given in Figs. 15 and 16, respectively.

4 Conclusions

The paper presented a novel hybrid wavelet-based coding algorithm termed video significance-linked connected component analysis for very low bit rate video coding applications. The proposed fine-tuned motion estimation combined with exhaustive overlapped block motion compensation produced coherent motion-compensated error frames with significantly reduced frame energy to which the wavelet transform associated with innovative data organization and representation strategies could be most successfully applied. In VSLCCA, significance-linked connected component analysis was used to organize and represent wavelet-transformed error frames. The information of significance-linked connected components or clusters in each subband was assumed an 18th order Markov source with an alphabet of four symbols and encoded by adaptive arithmetic coding. The significant magnitudes in each subband was also assumed an 18th order Markov source but with an alphabet of only two symbols and encoded in bit-plane order by adaptive arithmetic coding. The contexts used to define conditional probabilities needed by adaptive arithmetic coding in both cases were the same. With strong empirical evidences, we may say wavelet transform with innovative data organization and representation strategies represents an invaluable asset for not only still image coding but also video coding. Extensive computer experiments on several standard test sequences have shown that VSLCCA consistently outperforms both non-wavelet low bit rate video coding standard H.263 and high performance wavelet low bit rate video coder ZTE. VSLCCA is among the best low bit rate video coders.

Acknowledgment

The authors would like to thank Dr. Hung-Ju Lee from Sarnoff Corporation for providing the test video sequences used in the experiments, and Telenor R&D for providing the H.263 test model software. The authors would also like to thank the reviewers for their invaluable comments and suggestions that improved the quality of the paper.

References

[1] ITU-T Draft Recommendation H.263, "Video coding for low bitrate communications," Dec. 1995.

[2] ITU-T Recommendation H.261, "Video codec for audiovisual services at $p \times 64$ kb/s," 1990.

[3] ISO/IEC IS 11172 (MPEG-1), "Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s," 1993, Technical Report, Motion Picture Experts Group.

[4] ISO/IEC DIS 13818 (MPEG-2), "Generic coding of moving pictures and associated audio information," 1994, Technical Report, Motion Picture Experts Group.

[5] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 19–31, Feb. 1997.

[6] G. Karlsson and M. Vetterli, "Three dimensional sub-band coding of video," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1988, pp. 1110–1113.

[7] C.-H. Chou and C.-W. Chen, "A perceptually optimized 3-D subband codec for video communication over wireless channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 143–156, Apr. 1996.

[8] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 3, no. 4, pp. 572–590, Sept. 1994.

[9] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 559–571, Sept. 1994.

[10] C.I. Podilchuk, N.S. Jayant, and N. Farvardin, "Three-dimensional subband coding of video," *IEEE Transactions on Image Processing*, vol. 4, no. 2, pp. 125–139, Feb. 1995.

[11] Y.-Q. Zhang and S. Zafar, "Motion-compensated wavelet transform coding for color video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 3, pp. 285–296, Sept. 1992.

[12] K. Tsunashima, J.B. Stampleman, and Jr. V.M. Bove, "A scalable motion-compensated subband image coder," *IEEE Transactions on Communications*, vol. 42, pp. 1894–1901, 1994.

[13] J. Katto, J.-I. Ohki, S. Nogaki, and M. Ohta, "A wavelet codec with overlapped motion compensation for very low bit-rate environment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 3, pp. 328–338, June 1994.

[14] M.T. Orchard and G.J. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 693–699, Sept. 1994.

[15] G. Bhutani and W.A. Pearlman, "Image sequence coding using the zero-tree method," in *Proceedings of SPIE Conference Visual Communications and Image Processing*, 1993, vol. 2094, pp. 463–471.

[16] 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, Feb. 1997.

[17] S.P. Voukelatos and J.J. Soragham, "Very low bit-rate color video coding using adaptive subband vector quantization with dynamic bit allocation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 424–428, 1997.

[18] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

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

[20] S. Servetto, K. Ramchandran, and M.T. Orchard, "Wavelet based image coding via morphological prediction of significance," in *Proceedings of IEEE International Conference on Image Processing*, Oct. 1995, pp. 530–533.

[21] B.-B. Chai, J. Vass, and X. Zhuang, "Highly efficient codec based on significance-linked connected component analysis of wavelet coefficients," in *Proceedings of SPIE AeroSense*, Orlando, FL, Apr. 1997, vol. 3078, pp. 647–655.

[22] B.-B. Chai, J. Vass, and X. Zhuang, "Significance-linked connected component analysis for high performance low bit rate wavelet coding," in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Princeton, NJ, June 1997, pp. 145–150.

[23] B.-B. Chai, J. Vass, and X. Zhuang, "Significance-linked connected component analysis for low bit rate image coding," in *Proceedings of IEEE International Conference on Image Processing*, Santa Barbara, CA, Oct. 1997, pp. 637–640.

[24] B.-B. Chai, J. Vass, and X. Zhuang, "A novel data representation strategy for wavelet image compression," in *Proceedings of IEEE Workshop on Nonlinear Signal and Image Processing*, Mackinac Island, MI, Sept. 1997.

[25] B.-B. Chai, J. Vass, and X. Zhuang, "Significance-linked connected component analysis for wavelet image coding," *IEEE Transactions on Image Processing*, vol. 8, no. 6, pp. 774–784, June 1999.

[26] B.-B. Chai, J. Vass, and X. Zhuang, "Statistically adaptive wavelet image coding," in *Visual Information Representation, Communication, and Image Processing*, C.-W. Cheng and Y.-Q. Zhang, Eds., pp. 73–97. Marcel Dekker, New York, NY, May 1999.

[27] B.-J. Kim and W.A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," in *Proceedings of Data Compression Conference*, 1997, pp. 251–260.

[28] J. Vass, B.-B. Chai, and X. Zhuang, "3DSLCCA – A highly scalable very low bit rate software-only wavelet video codec," in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Redondo Beach, CA, Dec. 1998, pp. 474–479.

[29] J. Vass, B.-B. Chai, and X. Zhuang, "Significance-linked wavelet video coder," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Seattle, WA, May 1998, pp. 2829–2832.

[30] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[31] R.M. Haralick and L.G. Shapiro, *Computer and Robot Vision*, Addison-Wesley, Reading, MA, 1992.

[32] L. Vincent, "Morphological grayscale reconstruction in image analysis: Applications and effective algorithms," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 176–201, Apr. 1993.

[33] I.H. Witten, M. Neal, and J.G. Cleary, "Arithmetic coding for data compression," *Communications of ACM*, vol. 30, no. 6, pp. 520–540, June 1987.

[34] I. Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[35] J.B. Allen and L.R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proceedings of IEEE*, vol. 65, pp. 1558–1564, 1977.

[36] M. Vetterli, "Multi-dimensional subband coding: Some theory and algorithms," *Signal Processing*, vol. 6, pp. 97–112, 1984.

[37] Z. Xiong, K. Ramchandran, and M.T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 677–693, May 1997.

[38] J.W. Woods and S.D. O'Neil, "Subband coding of images," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 5, pp. 1278–1288, Oct. 1986.

[39] J. Kovačević, "Subband coding systems incorporating quantizer models," *IEEE Transactions on Image Processing*, vol. 4, no. 5, pp. 543–553, May 1995.

[40] P.H. Westernik, D.E. Boekee, J. Biemond, and J.W. Woods, "Subband coding of images using vector quantization," *IEEE Transactions on Communications*, vol. 36, pp. 713–719, June 1988.

[41] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[42] R.M. Haralick, S.R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 532–550, July 1987.

[43] A.S. Lewis and G. Knowles, "A 64 Kb/s video codec using the 2-D wavelet transform," in *Proceedings of Data Compression Conference*, Snowbird, Utah, 1991.

[44] X. Li and X. Zhuang, "The decay and correlation properties in wavelet transform," Tech. Rep., University of Missouri-Columbia, Mar. 1997.

[45] K.O. Lillevold *et al.*, "Telenor R&D, H.263 test model simulation software," Dec. 1995, Available at ftp://bonde.nta.no/pub/tmn/software.

[46] T. Wiegand, M. Lightstone, M. Mukherjee, T.G. Campbell, and S.K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 182–190, Apr. 1996.

[47] J. Vass, K. Palaniappan, and X. Zhuang, "Automatic spatio-temporal video sequence segmentation," in *Proceedings of IEEE International Conference on Image Processing*, Oct. 1998, pp. 958–962.

[48] W. Li and M. Kunt, “Morphological segmentation applied to displaced frame difference coding,” *Signal Processing*, vol. 38, pp. 45–56, 1994.

[49] D. Wang, C. Labit, and J. Ronsin, “Segmentation-based motion compensated video coding using morphological filters,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 549–555, June 1997.

Bitstream Component	Size [byte]
Explicit Seed Position	11
Significance Map	4989
Significant Magnitudes	3192
Total	8192

Table 1: Bit budget distribution among the explicit seed positions, significance map, and magnitudes of significant coefficients for the "Lena" image at 0.25 bpp.

Function	Encoder		Decoder	
	Time [sec]	Relative Time	Time [sec]	Relative Time
Wavelet Transformation	0.46	30.7%	0.55	39.3%
Quantization	0.07	4.7%	0.11	7.9%
Conditioned Dilation	0.14	9.3%	-	-
Postprocessing	0.28	18.7%	-	-
Encoding	0.55	36.6%	0.74	52.8%
Total	1.50	100.0%	1.40	100.0%

Table 2: SLCCA timing results (seconds) for 512 × 512 "Lena" image at 0.25 bpp.

Bit Rate [bpp] Algorithm	0.0625	0.125	0.25	0.5
SLCCA	20.70	22.24	24.26	27.22
SPIHT	20.27	22.04	24.14	27.04

Table 3: Performance comparison of SLCCA and SPIHT for the 114th motion-compensated error frame of the "Foreman" sequence.

Sequence	File Size	Luminance, PSNR [dB]			Chrominance, PSNR [dB]				
	[Bits]				U	V	Avg.	U	V
		H.263	ZTE	VSLCCA	H.263		ZTE	VSLCCA	
Akiyo	14000	33.06	34.62	35.34	35.23	37.38	36.19	37.12	39.45
Coast Guard	13888	30.12		31.17	41.98	44.94		42.83	45.12
Container Ship	14088	28.81		30.69	36.53	35.15		37.35	36.59
Foreman	13976	30.11	30.86	32.25	38.22	38.31	38.69	39.26	39.03
Hall Monitor	14104	29.68		31.83	36.04	39.99		37.40	40.29
Mother & Daughter	13624	33.78		35.53	39.27	39.67		40.85	41.57
News	14080	28.60	29.38	30.02	33.24	34.40	33.47	34.90	35.78
Silent Voice	13688	30.34		32.02	34.76	36.53		35.74	37.25

Table 4: Performance comparison of intraframe coding results at 14 kb.

Sequence	File Size	Luminance, PSNR [dB]			Chrominance, PSNR [dB]				
	[Bits]				U	V	Avg.	U	V
		H.263	ZTE	VSLCCA	H.263		ZTE	VSLCCA	
Akiyo	28248	38.42	40.18	41.67	40.09	41.53	40.81	42.82	43.25
Coast Guard	29280	34.23		35.54	44.13	45.68		44.88	46.69
Container Ship	28088	34.33		36.49	39.36	38.71		40.46	39.85
Foreman	26968	35.05	35.27	37.02	40.19	41.23	40.76	41.46	42.51
Hall Monitor	29040	35.62		38.32	38.66	40.87		40.75	42.25
Mother & Daughter	28088	38.77		40.46	42.27	42.70		44.07	44.90
News	27320	33.38	34.49	35.38	36.87	37.90	36.84	38.34	39.26
Silent Voice	26536	34.27		35.72	37.44	38.90		39.20	40.32

Table 5: Performance comparison of intraframe coding results at 28 kb.

Sequence	Bit Rate	Luminance, PSNR [dB]			Chrominance, PSNR [dB]				
	[kbps]				U	V	Avg.	U	V
		H.263	ZTE	VSLCCA	H.263		ZTE	VSLCCA	
Akiyo	8.87	34.61	34.61	35.55	38.45	40.52	39.86	39.81	41.43
Container Ship	10.04	30.79		31.13	37.81	36.98		37.94	36.72
Hall Monitor	8.18	30.36	30.25	31.51	36.21	39.35	38.05	37.45	39.95
Mother & Daughter	9.54	33.40		33.87	38.91	39.79		39.96	40.54

Table 6: Performance comparison of coding results for Class A sequences at 5 fps, 10 kbps.

Sequence	Bit Rate	Luminance, PSNR [dB]			Chrominance, PSNR [dB]				
	[kbps]				U	V	Avg.	U	V
		H.263	ZTE	VSLCCA	H.263		ZTE	VSLCCA	
Akiyo	22.29	37.46	36.64	37.98	41.77	42.53	44.02	42.31	43.05
Container Ship	23.30	32.84		33.20	39.05	38.20		38.97	37.82
Hall Monitor	21.41	34.46	34.11	34.74	38.35	40.41	39.63	38.77	41.15
Mother & Daughter	23.81	35.53		35.75	40.85	41.26		41.12	41.75

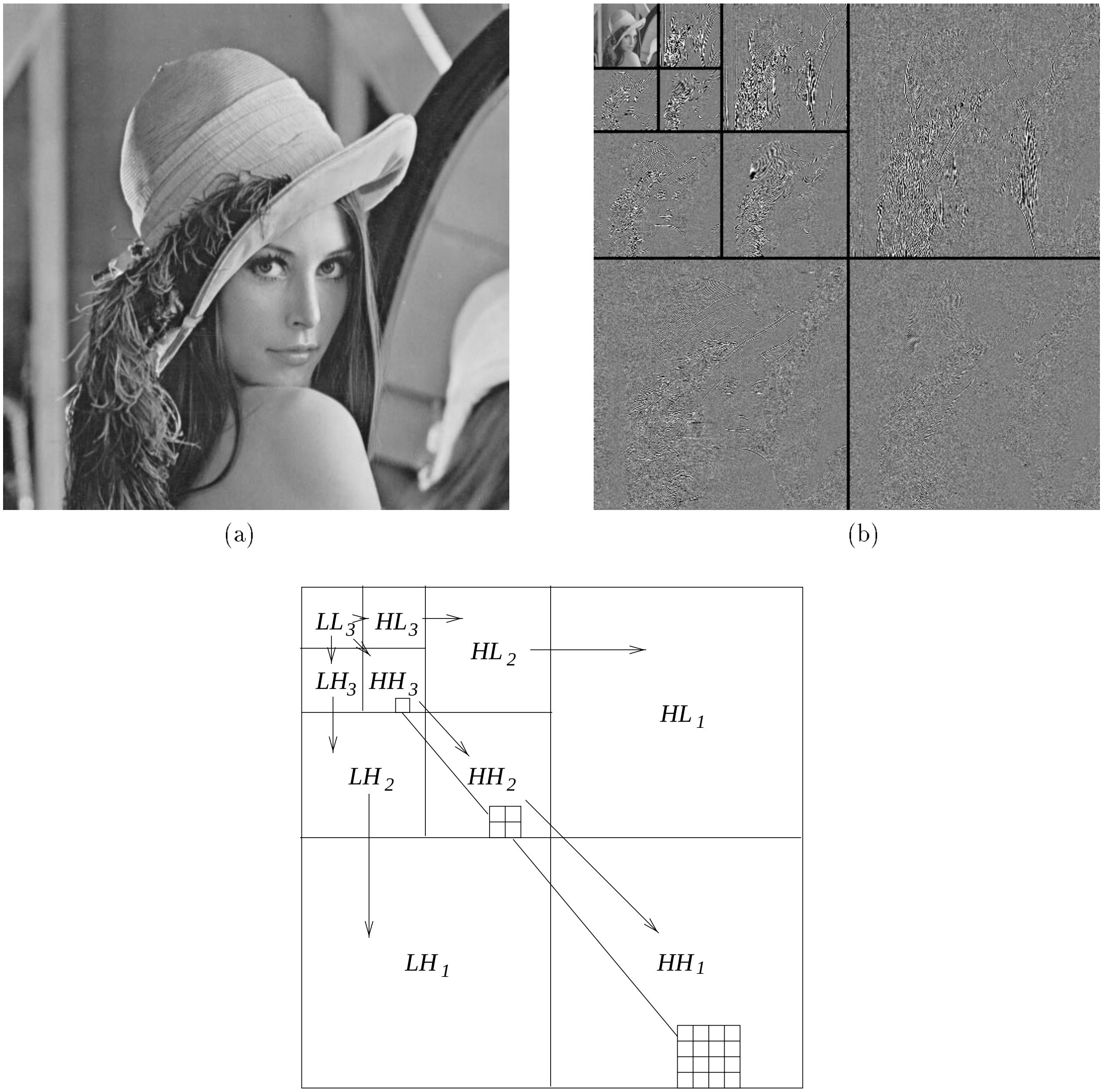
Table 7: Performance comparison of coding results for Class A sequences at 10 fps, 24 kbps.

Sequence	Bit Rate	Luminance, PSNR [dB]			Chrominance, PSNR [dB]				
	[kbps]				U	V	Avg.	U	V
		H.263	ZTE	VSLCCA	H.263		ZTE	VSLCCA	
Coast Guard	46.03	29.74	29.20	29.82	39.79	41.72	40.88	39.88	41.84
Foreman	46.77	31.91		32.26	37.54	37.71		37.43	37.56
News	49.85	35.10	35.17	35.41	38.76	39.45	40.46	38.77	39.64
Silent Voice	49.94	35.94		36.10	39.36	40.18		39.70	40.52

Table 8: Performance comparison of coding results for Class B sequences at 7.5 fps, 48 kbps.

Sequence	Bit Rate	Luminance, PSNR [dB]			Chrominance, PSNR [dB]				
	[kbps]				U	V	Avg.	U	V
		H.263	ZTE	VSLCCA	H.263		ZTE	VSLCCA	
Coast Guard	111.63	31.50	31.01	31.54	40.74	42.57	41.90	40.66	42.37
Foreman	118.62	34.47		34.49	39.25	39.73		39.16	39.30
News	109.95	37.68	37.59	37.62	40.89	41.50	42.55	40.53	41.14
Silent Voice	119.38	38.33		38.57	41.10	41.70		41.71	42.43

Table 9: Performance comparison of coding results for Class B sequences at 15 fps, 112 kbps.



(a)

(b)

(c)

Figure 1: Illustration of wavelet transform. (a) Original “Lena” image. (b) Three-scale wavelet decomposition of the “Lena” image and (c) the corresponding parent-child relationship between subbands at different scales.

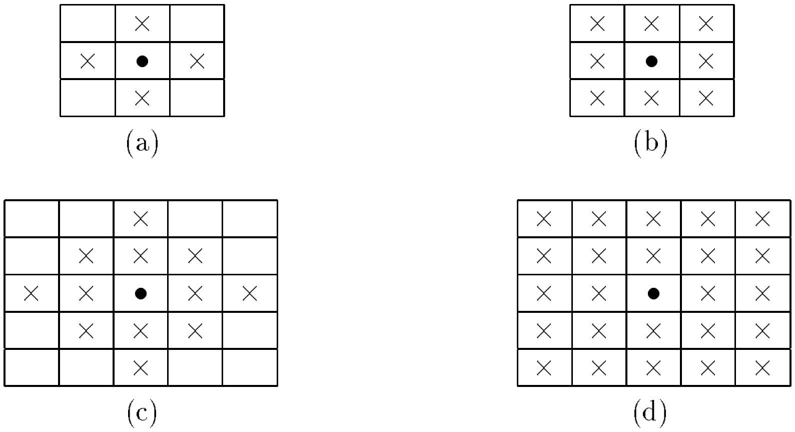


Figure 2: Structuring elements used in conditioned dilation. (a) 4-connected, (b) 8-connected, (c) diamond of size 13, and (d) 5 × 5.

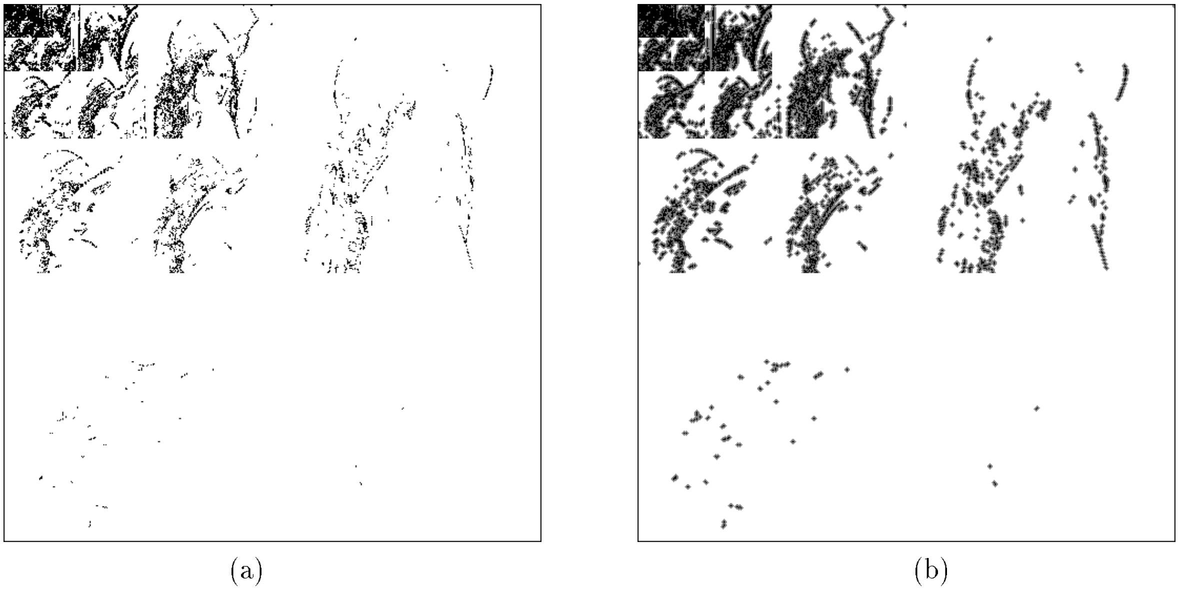
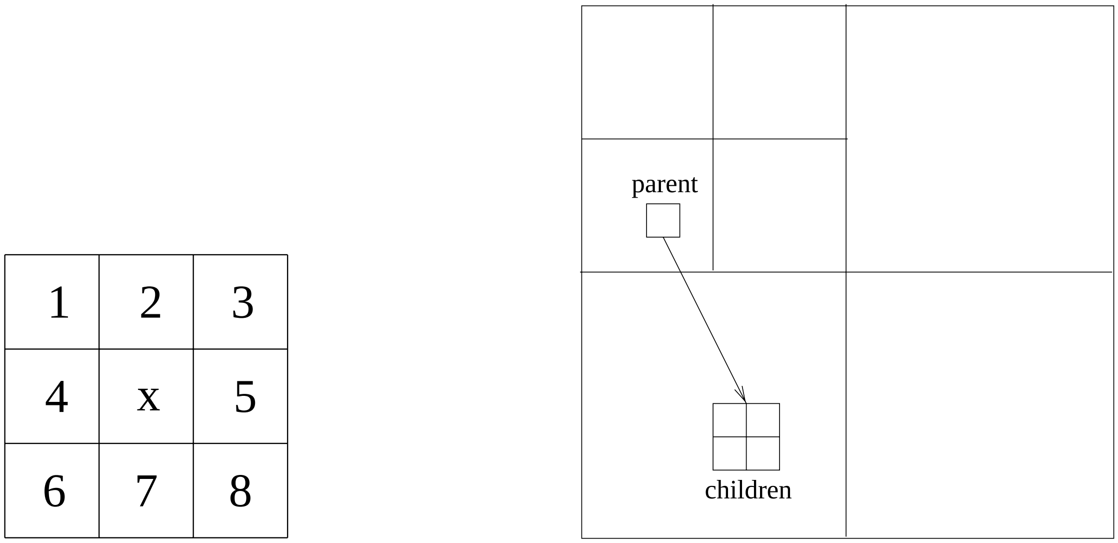


Figure 3: (a) Significance map for six-scale wavelet decomposition of "Lena" image after quantization (q=20.98) and removing isolated wavelet coefficients. White pixels denote insignificant coefficients and black pixels denote significant coefficients. (b) The transmitted significance map with diamond structuring element (Fig. 2c). White pixels denote insignificant coefficients that are not encoded at all. Black and gray pixels denote encoded significant and insignificant wavelet coefficients, respectively.



(a) (b)

Figure 4: Conditioning context for adaptive arithmetic coder. Dependency on (a) neighboring pixels and (b) parent pixel.

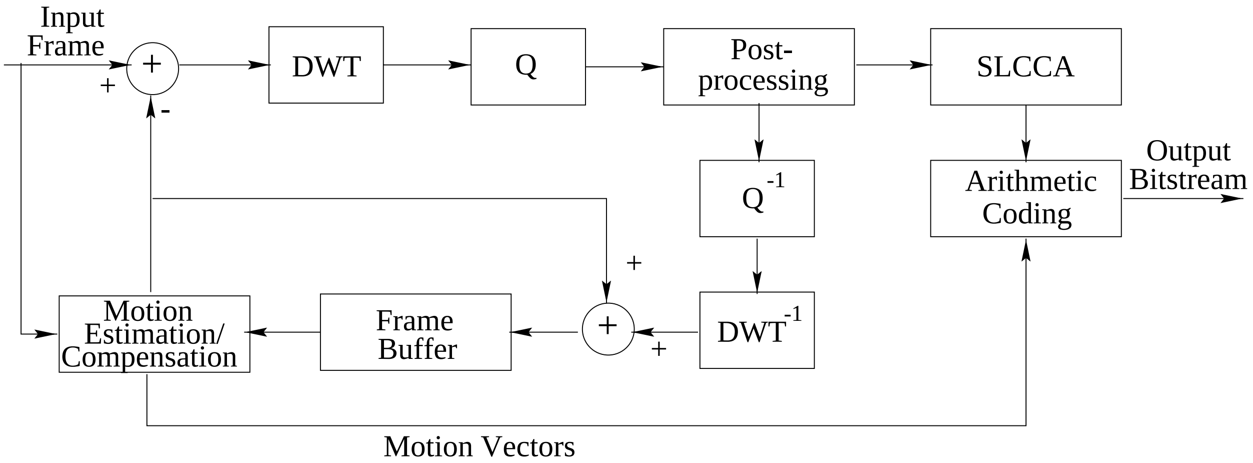


Figure 5: Block diagram of the proposed VSLCCA video coding algorithm.

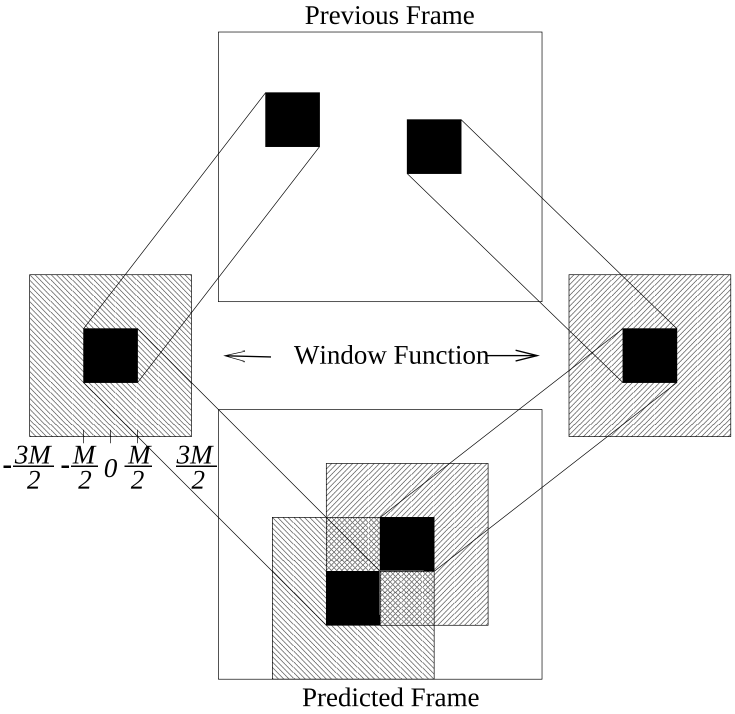

Previous Frame
Window Function
$-\frac{3M}{2}$ $-\frac{M}{2}$ 0 $\frac{M}{2}$ $\frac{3M}{2}$
Predicted Frame

Figure 6: Exhaustive overlapped block motion compensation used in VSLCCA, $M = 8$ denotes the block size.

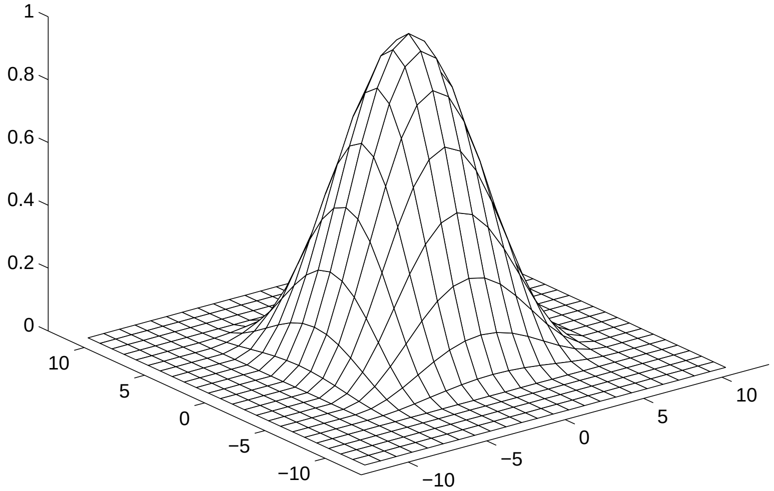
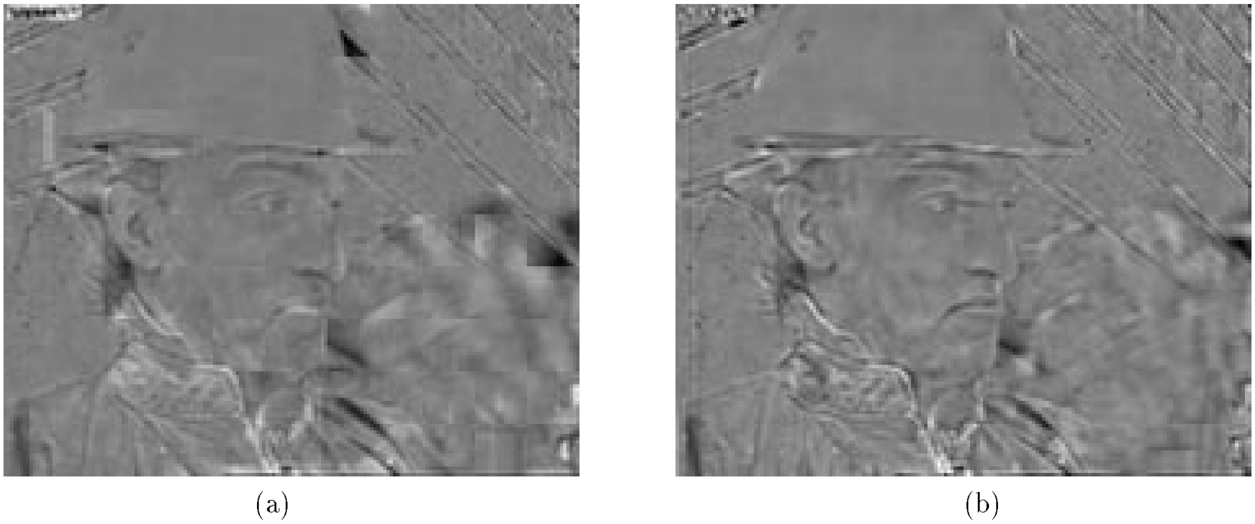


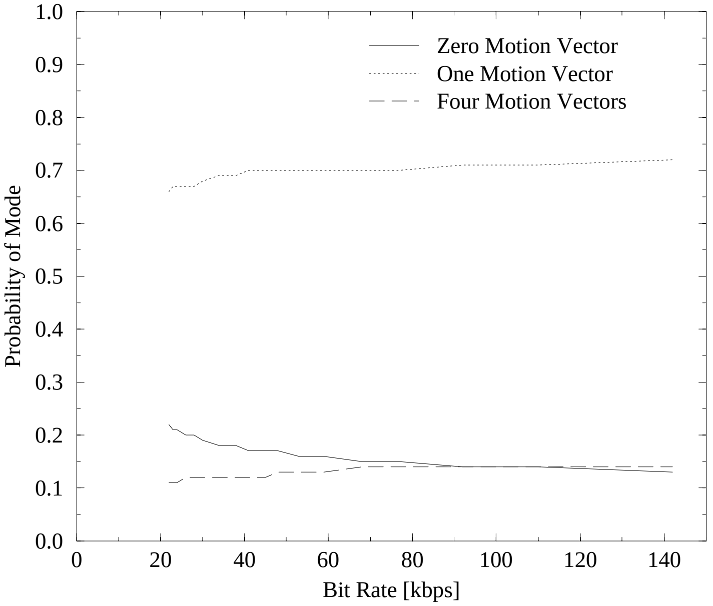
Figure 7: Raised cosine window for block size 8×8 with 4 pixels overlap. $(0, 0)$ is the center of the block.



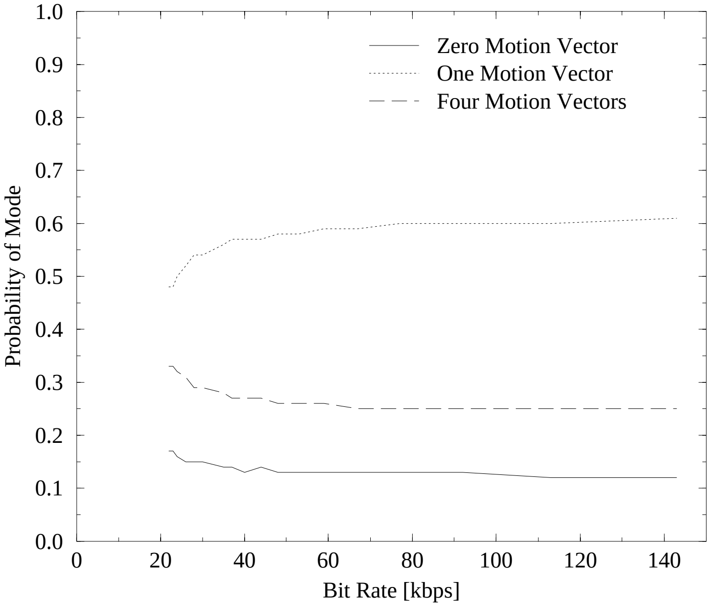
(a)

(b)

Figure 8: Motion-compensated error frame for 156th frame of “Foreman” sequence by (a) H.263 test model and (b) the proposed motion estimation and motion compensation algorithm.

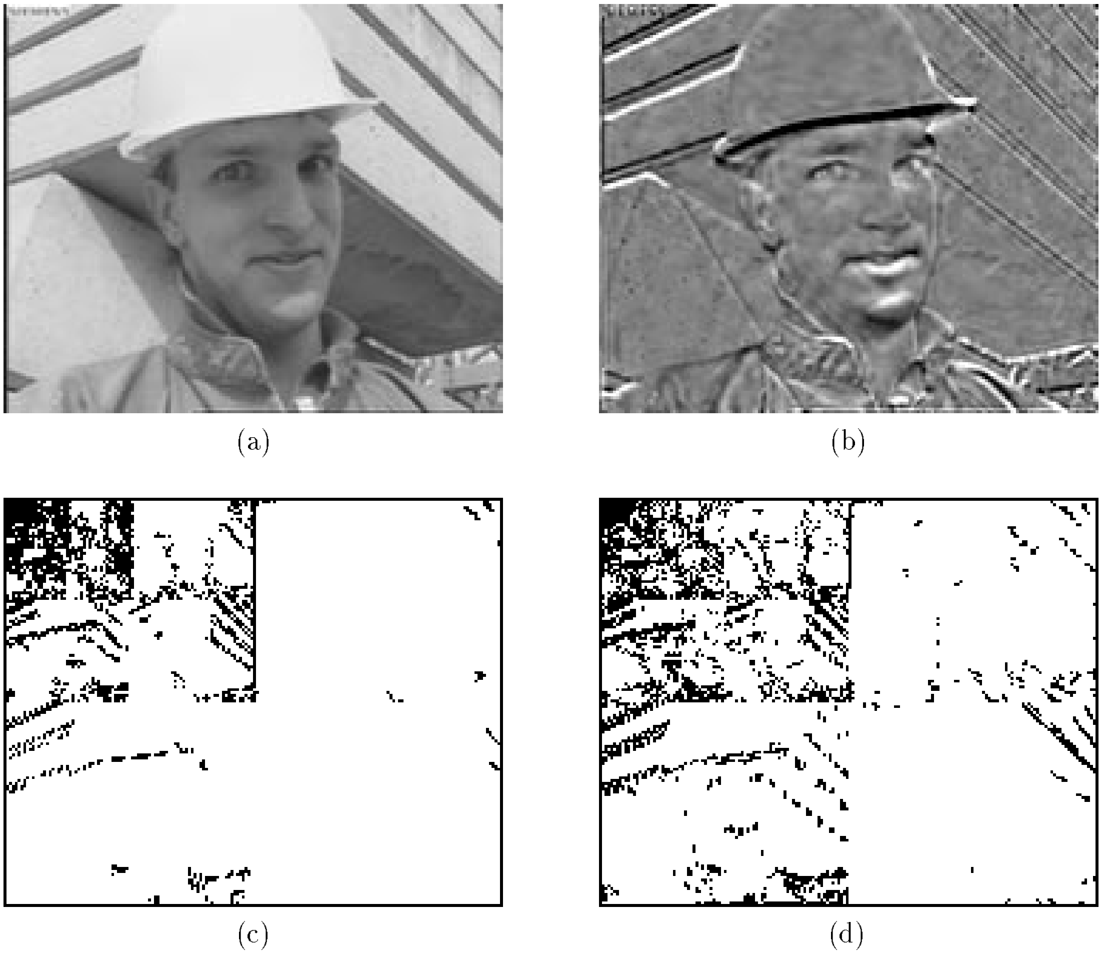


(a)



(b)

Figure 9: Mode selection versus average bit rate for the “Foreman” sequence sampled at 5 fps by (a) H.263 and (b) VSLCCA. For H.263 zero motion vector includes uncoded macroblocks and macroblocks with zero motion vector. On average, about one macroblock per frame is coded in intra mode.



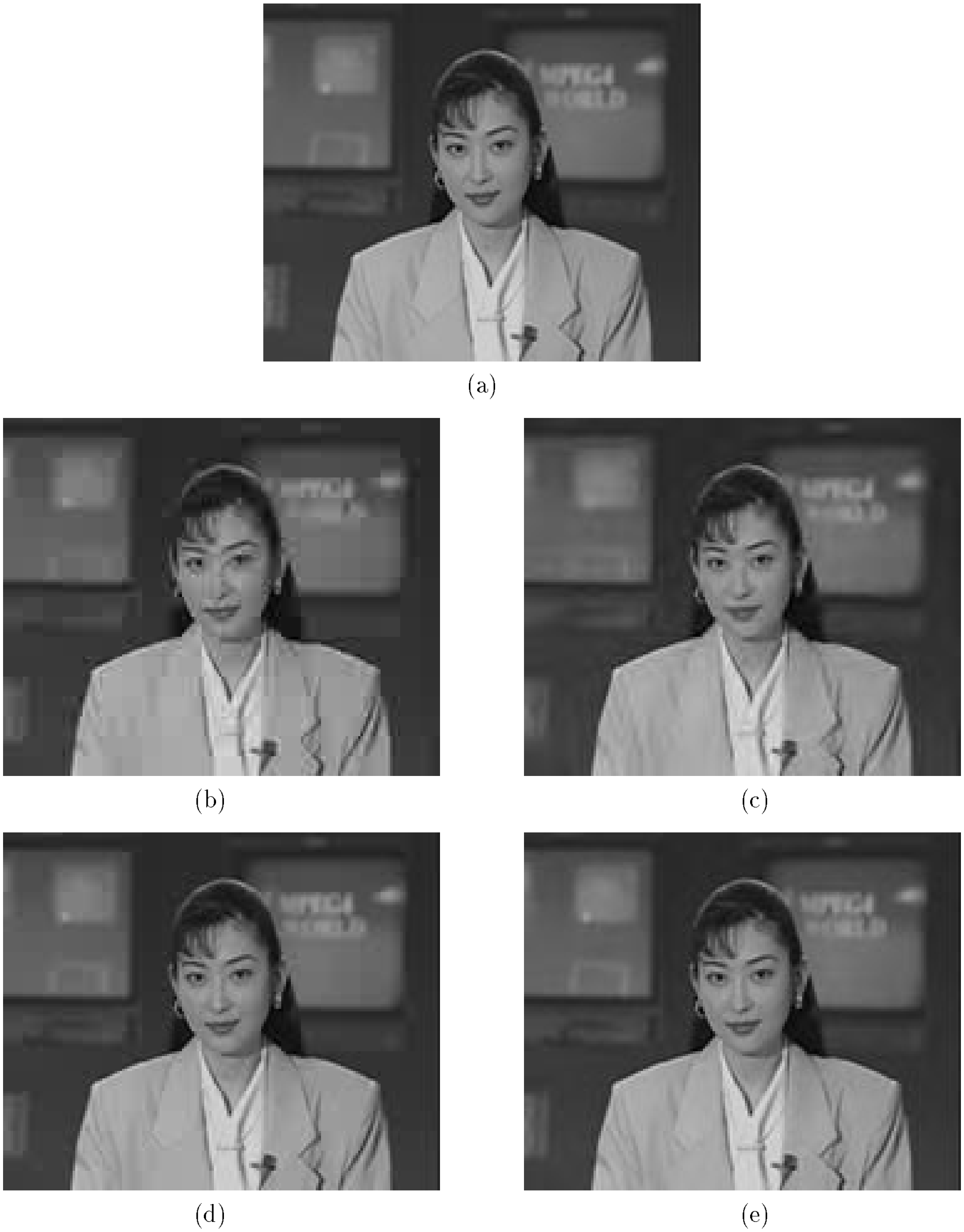
(a)

(b)

(c)

(d)

Figure 10: Illustration of the wavelet transform on the original and motion-compensated error of the 114th frame of the "Foreman" sequence. (a) Original and (b) motion-compensated error frames. Significance map after wavelet decomposition and quantization of the (c) original and (d) motion-compensated error frames. (Motion-compensated images have been processed for display.)



(a)

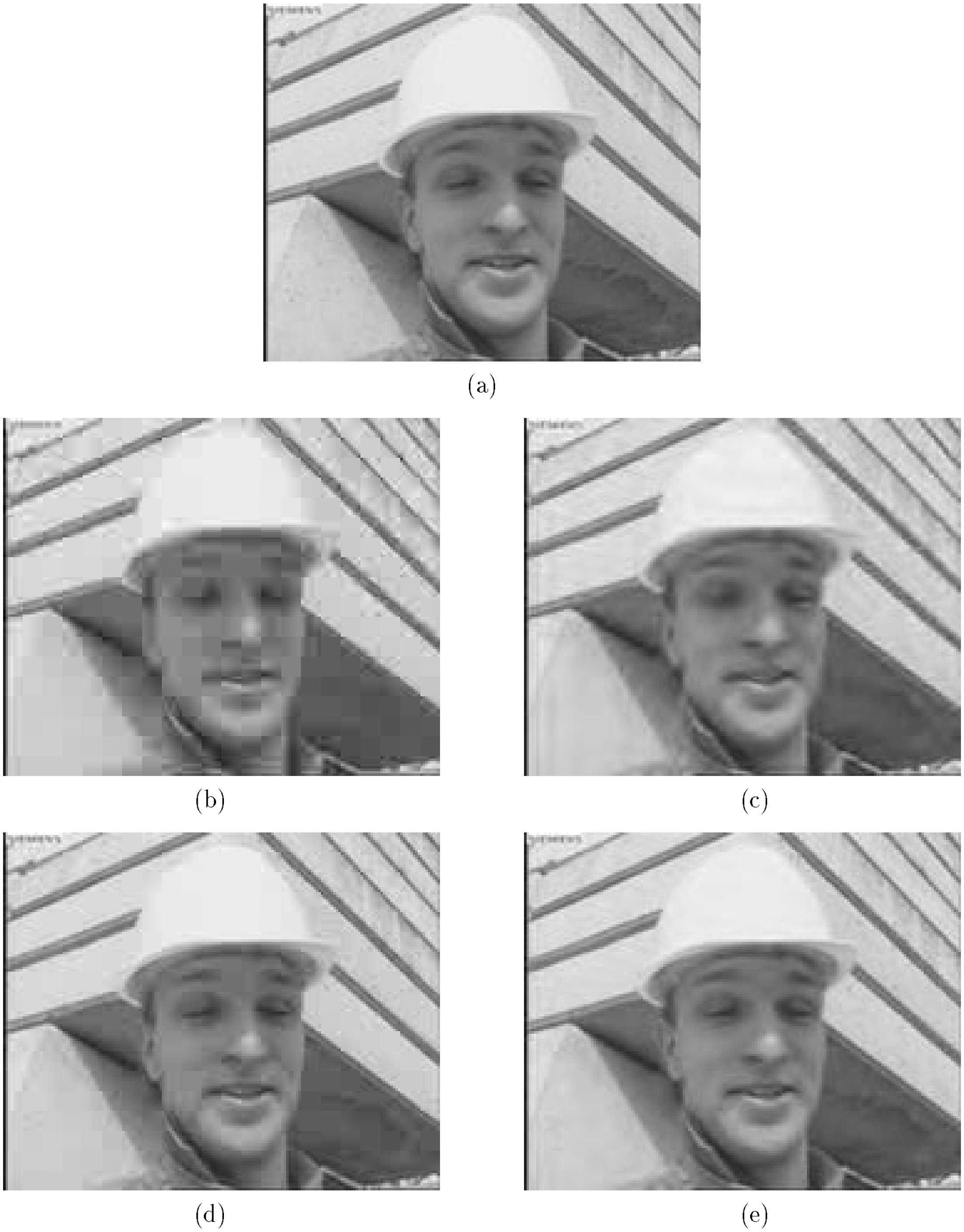
(b)

(c)

(d)

(e)

Figure 11: Intraframe coding results of the first frame of the “Akiyo” sequence. (a) The original frame, (b) reconstructed frame by H.263 at 14 kb, PSNR = 33.06 dB, and (c) by VSLCCA, PSNR = 35.34 dB. (d) Reconstructed frame by H.263 at 28 kb, PSNR = 38.42 dB, and (e) by VSLCCA, PSNR = 41.67 dB.


(a)

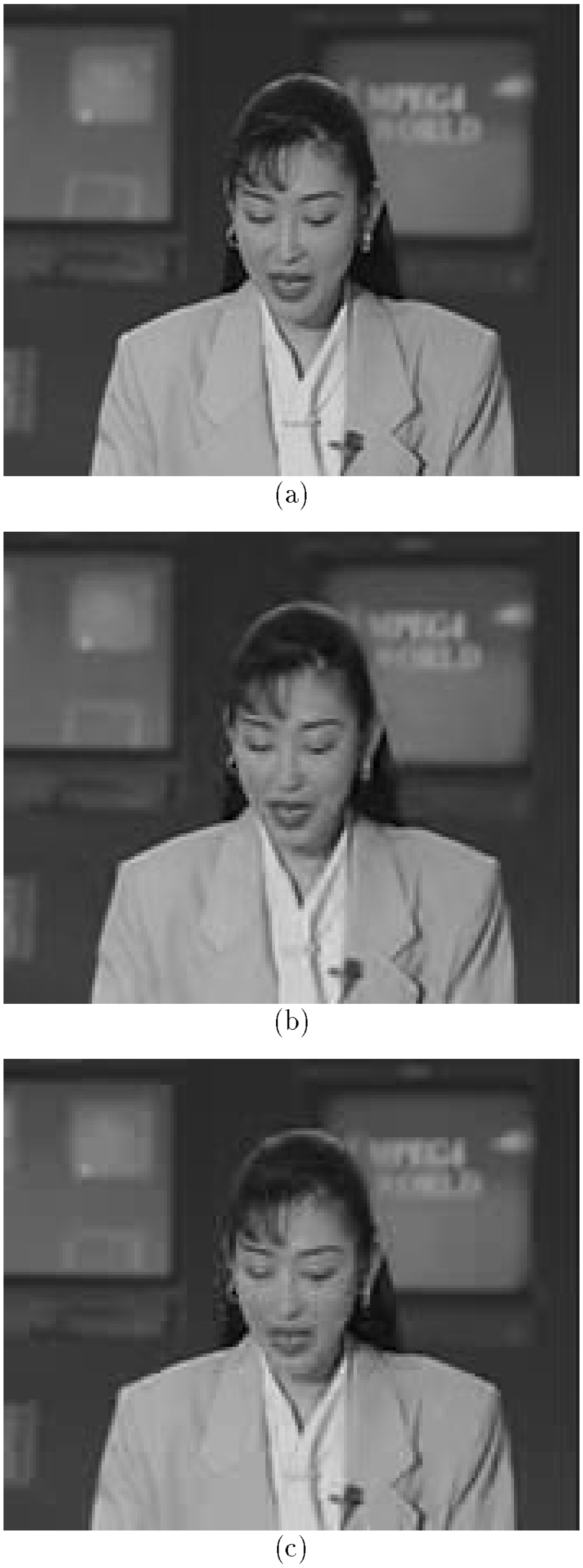
(b)

(c)

(d)

(e)

Figure 12: Intraframe coding results of the first frame of the "Foreman" sequence. (a) The original frame, (b) reconstructed frame by H.263 at 14 kb, PSNR = 30.11 dB, and (c) by VSLCCA, PSNR = 32.25 dB. (d) Reconstructed frame by H.263 at 28 kb, PSNR = 35.05 dB, and (e) by VSLCCA, PSNR = 37.02 dB.

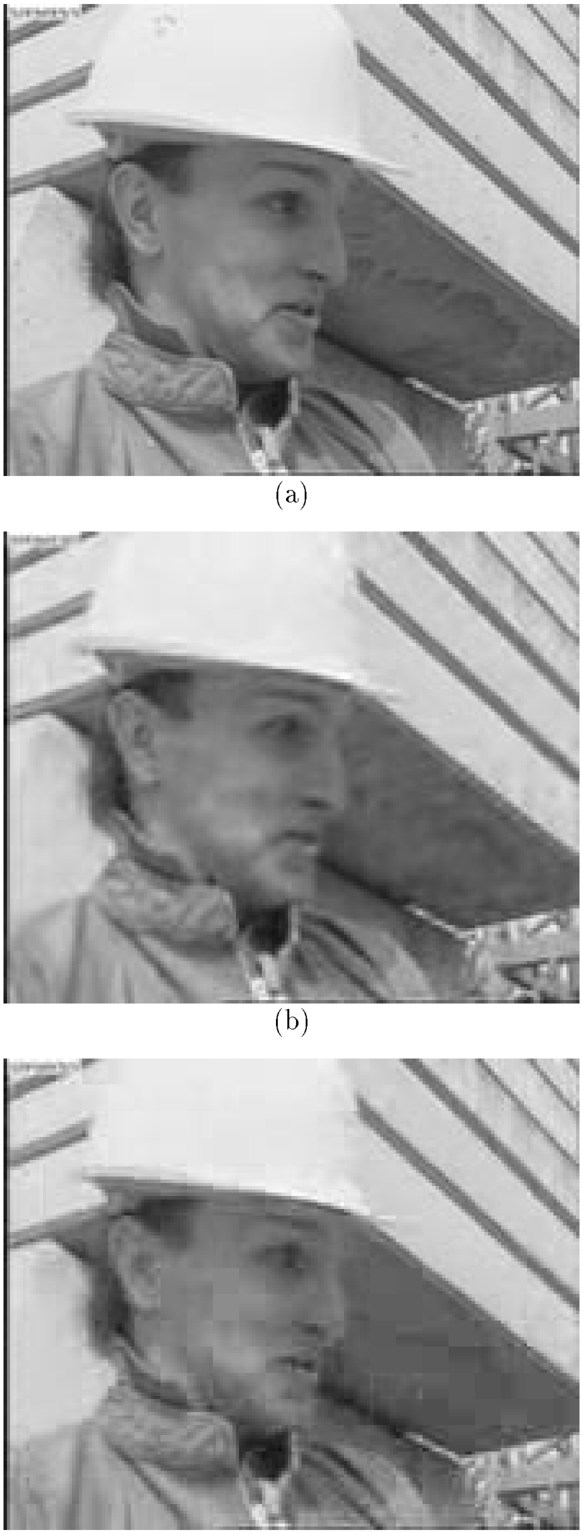


(a)

(b)

(c)

Figure 13: Interframe coding results of the 54th frame of the “Akiyo” sequence at 5 fps, 10 kbps. (a) The original frame, reconstructed frame by (b) VSLCCA and (c) H.263.



(a)

(b)

(c)

Figure 14: Interframe coding results of the 160th frame of the "Foreman" sequence at 7.5 fps, 48 kbps. (a) The original frame, reconstructed frame by (b) VSLCCA and (c) H.263.

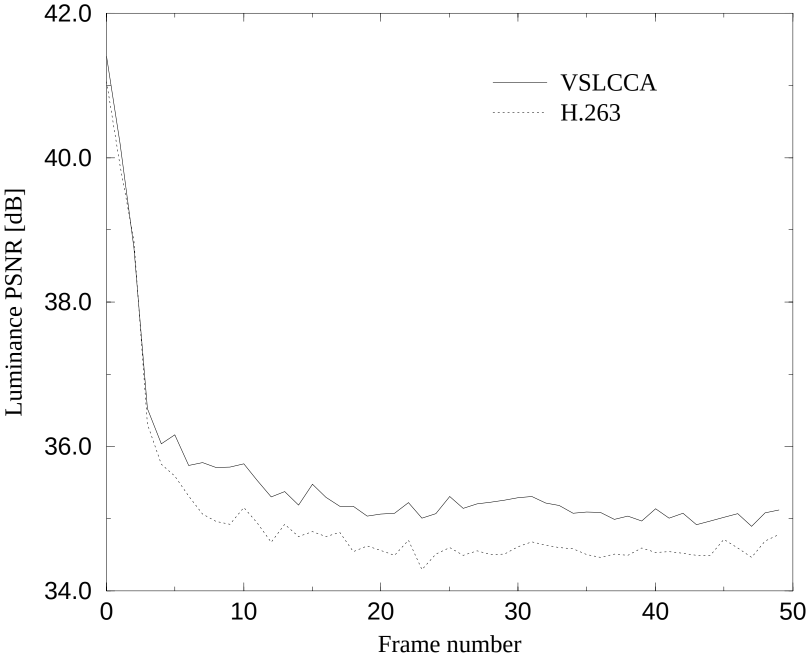


Figure 15: Performance comparison (PSNR, [dB]) between H.263 and VSLCCA for the luminance component of the "Akiyo" sequence at 5 fps, 10 kbps.

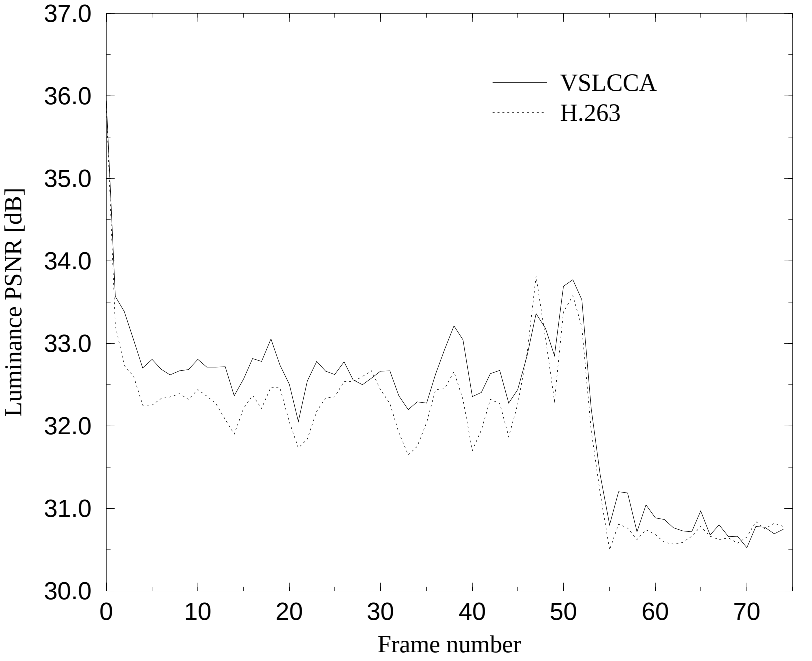


Figure 16: Performance comparison (PSNR, [dB]) between H.263 and VSLCCA for the luminance component of the "Foreman" sequence at 7.5 fps, 48 kbps.