Motion Differential Set Partition Coding for Image Sequence and Video Compression

Yang Hu and William A. Pearlman[1]

Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, U.S.A.

Abstract

Efficient image sequence coding exploits both intra- and inter-frame correlations. Set partition coding (SPC) is efficient in intra-frame de-correlation for still images. Based on SPC, a novel image sequence coding system, called *motion differential SPC* (M-D-SPC), is presented in this paper. It removes inter-frame redundancy by re-using the significance map of a previously SPC coded frame. Every frame is encoded and decoded separate from other frames. Furthermore, there is no reconstruction of encoded frames in the encoder, as is done with interframe prediction methods. The M-D-SPC exhibits an *auxiliary key frame* coding framework, which achieves higher coding efficiency compared to the all-intra-coding schemes and meanwhile maintains the beneficial features of SPC all-intra-coding, such as computational simplicity, rate scalability, error non-propagation, and random frame access. SPIHT-based simulations on hyperspectral images, 3D/4D medical images, and video show greater compression efficiency than the standard intraframe

Email address: , `huy5@rpi.edu`, `pearlw@ecse.rpi.edu` (Yang Hu and William A. Pearlman)

[1]We gratefully acknowledge the support of the Office of Naval Research under Contract No. N00014-05-1-0507.

coding method of motion JPEG2000.
Keywords: video coding, image sequence coding, motion coding, set partition coding, significance map.

1. Introduction

Image sequence coding achieves high efficiency by exploiting both spatial (intra-frame) and temporal (inter-frame) correlations. Set partition coding (SPC) is an efficient means for encoding transforms of datasets, regardless of dimension, and exploits well corelations within and among image frames. For two dimensions, starting with Shapiro's pioneering work on Embedded Zerotree Wavelet (EZW) coding [1] to the JPEG 2000 [2, 3] image compression standard to the new 2D lossy-to-lossless compression algorithm recently standardized by the Consultative Committee for Space Data Systems (CCSDS) [4, 5], SPC has served as an auxiliary component to the coding efficiency of wavelet-based approaches. SPC has also been successfully applied into several other well-known image coding algorithms including SPIHT [6], SPECK [7], SWEET [8], SBHP [9] and EZBC [10]. As an effective way of representing image data in the wavelet domain, SPC (sometimes combined with other entropy coding methods, such as arithmetic coding) takes advantage of the beneficial localization properties of wavelet transforms to achieve competitive rate-distortion performance. A comprehensive tutorial on SPC and its usage in wavelet coding systems can be found in [11, 12].

To exploit temporal correlation, there are two main approaches: inter-frame prediction and three-dimensional (3D) decomposition. Inter-frame prediction with motion compensation is the more computationally intensive and is widely used in the MPEGx and H.26x video coding standards. The 3D decomposition includes 3D wavelet transform, 3D complex dual-tree wavelet transform [13], and so on. In the 3D decomposition approaches, both 2D and 3D SPC methods can be used. For example, 3D JPEG2000 coding can be realized with a 3D wavelet transform followed either by coding cubic subblocks [14] or by coding planar (x-y) subblocks in every transformed frame [3]. The coding scheme with any of the above inter-frame de-correlation methods is typically referred to as *inter-coding*, where the frame decoding is at the GOF (group of frames) level depending on the period of intra-coded frames. Inter-coding achieves higher coding efficiency, but these inter-frame de-correlation techniques are the main contributors to the computational complexity and inter-frame error propagation. Interframe coding requires prediction from previous reconstruction frames that must be decoded at the source as well as in the receiver. When used with motion estimation and compensation, reconstruction at the encoder side can be highly complex. In addition and germane to intent of this work, the inter-coding framework prohibits random frame access, which is an important feature for post-production editing applications. To overcome these drawbacks, intra coding is used for every frame of an image sequence, such as in motion JPEG 2000 (MJ2K) [15] — the digital cinema distribution standard, where every frame is coded independently with JPEG 2000. However, the *all-intra-coding* methods sacrifice coding efficiency to some extent due to the unexploited inter-frame correlation.

The primary motivation of this paper is to present a method that exploits inter-frame correlation to achieve higher coding efficiency and meanwhile maintains the beneficial features of all-intra-coding such as computational simplicity, parallel processing, no error-propagation between frames, and random frame access. An *auxiliary key frame* coding framework is presented in this paper, named *motion differential SPC* (M-D-SPC), that exploits to some degree the correlation among image frames. The M-D-SPC adaptively splits an image sequence into key-frames and correlated-frames, which are coded with conventional SPC and differential-SPC, respectively. The differential-SPC of a correlated-frame uses the significance map of a previously SPC coded key-frame; and the decoding of any frame (key- or correlated-frame), involving only the common data (the significance map and sign information) and the self data of that specific frame, is separate from the decoding of other frames. The idea of re-using the significance map was initially proposed in [16] based on SPIHT. To the best of our knowledge, it was the first such approach designed for SPC methods to exploit inter-frame correlation. Compared to [16], this paper extends differential-SPIHT to differential-SPC so that it can be uniformly applicable to other SPC methods, such as SPECK, SBHP, EZBC, etc. An adaptive frame splitting method is introduced into M-D-SPC to combat the non-stationarity of the inter-frame correlation. We shall show that M-D-SPC gives competitive performance to other methods in compressing hyperspectral images, 3D and 4D medical images, and video with low to moderate motion. M-D SPC also retains the SPC properties of a rate-scalable bit-stream and low computational complexity.

The rest of this paper is organized as follows. A brief description of SPC is given in section 2. In section 3, M-D-SPC is presented. Simulation results are given in section 4, and section 5 concludes the paper.

2. Set Partition Coding

In this section, we provide a brief overview of SPC. In SPC, the *set* is a group of coefficients in a specially designed data structure — an efficient data structure such as zerotree or spatial-orientation-tree takes advantage of the characteristics of natural images in the wavelet domain to group a large number of insignificant coefficients into an insignificant set; for a significant set, a pre-designed *partitioning* rule is applied to divide it into smaller subsets recursively so that the significant coefficients can be efficiently isolated. Some representative set partition methods are shown in Table 1, with their corresponding SPC algorithms and the exploited characteristics of wavelet coefficients. These characteristics are listed below.

1. Self-similarity across scales.
2. Energy clustering in frequency (or known as the decaying spectral distribution for natural imagery).
3. Energy clustering in space.

Methods	**Characteristics**	**Corresponding Algorithms**
Zerotree	1,2	EZW
SOT	1,2,3	SPIHT
Quadtree	3	SWEET, SPECK, SBHP, EZBC
Octave	2	SWEET, SPECK, SBHP, EZBC

The SOT stands for Spacial Orientation Tree.

Table 1: Representative set partitioning methods.

A 2D wavelet transformed frame is viewed as a collection of non-overlapping sets indexed by descriptors (e.g., the top-left coefficient coordinate and the subband number of the top level coefficients of a spatial-orientation-tree in SPIHT, the top-left coefficient coordinate and the size of a block in quadtree partitioning). When the magnitudes of all coefficients in a set (with single or multiple elements) are less than the current threshold, the set is deemed *insignificant* and a "0" is sent to the codestream; otherwise, the set is deemed *significant* and a "1" is sent. Initially, all sets are insignificant. SPC iteratively tests insignificant sets by comparing them with the threshold of each iteration, called *significance tests*. Let X^l be the wavelet coefficient at location l. The thresholds are usually chosen as $T = 2^n$, starting from integer $n = n_{max} = \lfloor \log_2(\max_l |X^l|) \rfloor$. When a set tests as significant, it is partitioned into subsets according to the pre-defined set partitioning rule. The subsets are further significance tested and partitioned if needed until the individual significant coefficients are isolated. Then, the sign bit of the significant coefficient is sent and its coordinates are moved to an ordered list, called the LSP (List of Significant Points). These tests proceed until all the previously insignificant sets and coefficients have been tested at the operative threshold. This procedure is called the *sorting pass*. After a sorting pass at threshold $T = 2^n$ (bit plane n), the bits in the n-th bit-plane of LSP coefficients, which are previously found significant at higher thresholds, are sent to the codestream. This procedure is called the *refinement pass*. Then the threshold is lowered by a factor of 2, or equivalently the n is lowered by 1, for the next iteration. There is a flexible number of iterations and each iteration includes a sorting pass and a refinement pass.

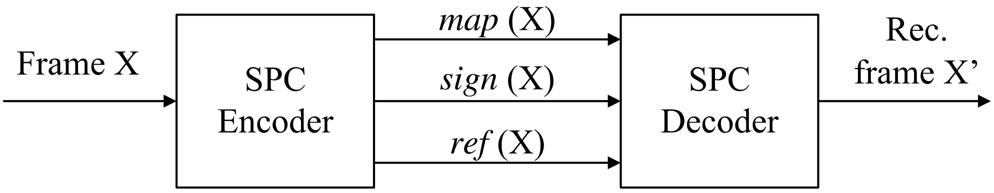


Figure 1: Conventional SPC codec.

The output of significance tests from the sorting passes is called the *significance map*, which conveys the execution path of the encoder to the decoder. The sign bits constitute another class of data from the sorting passes. A third class contains the refinement bits from the refinement passes. The SPC codec is shown in Fig. 1, where *map*(*), *sign*(*), and *ref*(*) indicate the significance map, sign bits, and refinement bits, respectively. Among them, the significance map consumes most of the codestream. For SPIHT-coded 8-bit 512×512 grayscale Lena (3-level wavelet transform with 9/7 [17] filters), the portions of the three kinds of data are provided in Fig. 2 for different recovery PSNR, and the corresponding bit rates are given in Table 2. For commonly used recovery PSNR (from 30 to 40 dB), the significance map consumes about 70% of the codestream, as in Fig. 2(a); after applying arithmetic coding to the significance map, it still consumes about 60% of the codestream, as in Fig. 2(b).

If conventional 2D SPC is used to code every frame of an image sequence, the significance map will take a large percentage of the codestream. The M-D-SPC reduces the bit consumption of the significance map by means of re-using one map among several highly correlated (similar) frames, and thus

	Bit Rate (bpp)	
PSNR (dB)	Without AC	Map with AC
22.3	0.062	0.025
26.9	0.099	0.052
30.0	0.152	0.095
33.2	0.245	0.176
36.4	0.417	0.332
39.5	0.748	0.634
43.3	1.527	1.328
48.5	2.744	2.470

Table 2: The bit rates at different recovery PSNR for SPIHT coded 8-bit 512×512 grayscale Lena (3-level wavelet transform with 9/7 filters).

reduces the overall bit rates. The details of this method are given in the following section.

3. Motion Differential SPC

The Motion Differential SPC (M-D-SPC) is so named, since it has a similar framework with other so-called *motion-coding* methods, such as the motion JPEG [18] and motion JPEG2000 [15] in that random access to any frame at the decoder is assured. In M-D-SPC, the frames of an image sequence are compressed either as *key-frames* or as *correlated-frames*. The former are coded with the conventional 2D SPC. The latter are coded with differential-SPC, which utilizes the significance map and the sign data of a previously coded key-frame to predict the locations and signs of the significant coefficients of the current frame.

3.1. M-D-SPC Encoder

In an image sequence, after each key-frame, there could be N ($N \in \{0, 1, 2, ...\}$) correlated-frames. In this "1-key-N-correlated" frame structure, there are two methods to decide the number N. First, N can be fixed to some value for the whole image sequence. For example, if $N = 1$ for the whole sequence, then every other frame is coded as a key-frame, and the frame after every key-frame is a correlated-frame re-using the significance map of its previous key-frame. This fixed N method is simple, but it requires the inter-frame correlation to be stationary for the whole sequence to be efficient. However, most image sequences do not satisfy this requirement, so we use a second method that determines N adaptively during the coding procedure. To be concrete, some correlation metric, such as PSNR (peak signal-to-noise ratio), is adopted, and a threshold T_{PSNR} is selected in terms of this metric. If the PSNR between the key-frame and the current frame is greater than the threshold T_{PSNR}, the current frame will be coded as a correlated-frame; otherwise, the current frame will be coded with conventional 2D SPC and serves as a new key-frame, until a newer key-frame appears.

The codec dealing with such a frame structure is illustrated in Fig. 3, where the frames X_0 and X_i ($i \in \{1, 2, ..., N\}$) represent the key- and correlated-frames, respectively. The correlated-frames X_i ($i \in \{1, 2, ..., N\}$) can be encoded in parallel, as shown in Fig. 3, or in serial.

3.2. Differential-SPC

Differential-SPC is the coding scheme applied to the correlated-frames, X_i's, $i \in \{1, 2, ..., N\}$. Let the bit-plane in the last coding iteration be n_{last}. The significance map, $map(X_0)$, iteratively indicates the locations of the significant coefficients of X_0, called *significant locations*. The remaining locations with insignificant coefficients with respect to $2^{n_{last}}$ are called *insignificant locations*. For frame X_0, it is obvious that all the coefficients on the significant locations are significant and on the insignificant locations are insignificant. However, when applying $map(X_0)$ to correlated-frame X_i, there are four possible cases:

I Significant coefficient in significant location.

II Insignificant coefficient in insignificant location.

III Insignificant coefficient in significant location.

IV Significant coefficient in insignificant location.

If X_0 and X_i are highly correlated, the first two cases are much more likely to happen. To make sure no reconstruction fidelity loss compared to the conventional SPC with n_{last}, the differential-SPC should assure all significant coefficients with respect to $2^{n_{last}}$ being coded, that is, all the significant coefficients in the significant locations (Case I) and in the insignificant locations (Case IV) should be coded. The stages *SPC** and *differential-sorting* in differential-SPC cover the Case I and IV, respectively, as illustrated in Fig. 4.

*3.2.1. SPC**

The SPC* is an important step in differential-SPC. It uses the significance map and sign data of a previously coded key-frame to predict locations and signs of the significant coefficients of the correlated-frame. The algorithm of SPC* is provided in Algorithm 1. Different from conventional SPC, there is no significance test to generate a new significance map. Instead, SPC* reads in the significance map of key-frame, $map(X_0)$, to iteratively identify/predict the possibly significant positions for X_i, $i \in \{1, 2, ..., N\}$. At each iteration, the selected significant positions of X_i are exactly the same as those of X_0. The X_i coefficients in these significant positions are typically significant with a high probability, because the key- and correlated-frames are highly correlated. However, due to the difference of X_0 and X_i, the following indicators are introduced.

Let l represent the significant location at the iteration n with threshold $T = 2^n$. The coefficient X_i^l involves sign and magnitude. The sign of X_i^l is indicated by a prediction indicator p based on the sign of X_0^l.

$$p = \begin{cases} 0 & \text{if } sign(X_0^l) = sign(X_i^l) \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

For the magnitude, conventional SPC need not explicitly send the n-th bit of the newly found significant coefficient in coding iteration n, because it must be "1" and it must be the Most Significant Bit (MSB) of the coefficient. But for X_i^l,

- it may be insignificant ($|X_i^l| < T$ or its MSB-plane is lower than n);
- it may be *just* significant ($T \leq |X_i^l| < 2T$ or its MSB-plane is n);

Algorithm 1 SPC* Encoding

1. **Initialization:**
   (a) Create a list LSP (List of Significant Points), initially empty, to contain "1"-labelled (called significant) coefficients.
   (b) Create a list LIS (List of Insignificant Sets) to contain "0"-labelled (called insignificant) sets.
   (c) Put descriptors of the initial sets onto LIS. These initial sets come from a non-overlapped partition of the (transformed) frame X_i ($i \in \{1, 2, , N\}$) according to the adopted set partition method.
   (d) Read n_{max} from buffer, the most significant bit of the largest magnitude of the key-frame X_0. Set $n = n_{max}$.
2. **Sorting Pass:**
   (a) For each set on the LIS, read next bit from the buffered key-frame map $map(X_0)$.
      i. If "0", return to 2a.
      ii. Otherwise, if "1", do the following:
         A. If set has more than one element, then divide the set into subsets according to the pre-defined set partition rule, and add each subset to the end of the LIS.
         B. If set has one element, then read next bit from the buffered $sign(X_0)$ and send its sign prediction indicator p as in (1); send its value indicator v according to (2); update the coefficient value to its $n-1$ least significant bits (LSBs) as in (3) and add it to the end of the LSP.
         C. Remove the set from LIS.
3. **Refinement Pass:**
   For each coefficient on the LSP, which is added to the list in previous iterations, send its n-th LSB and update the coefficient value to its $n-1$ LSBs.
4. If $n > n_{last}$, where n_{last} is the last coded bit-plane of the key-frame, then set $n = n-1$ and return to 2a. Otherwise stop.

- it may be *super* significant ($|X_i^l| \geq 2T$ or its MSB-plane is higher than n).

Thus a value indicator v is sent to specify the n-th and higher bit-planes of X_i^l.

$$v = \lfloor \frac{|X_i^l|}{T} \rfloor \tag{2}$$

which is equivalent to right-shifting the binary expression of $|X_i^l|$ by n. Theoretically, there is no upper limit of v, so smaller v's are coded with arithmetic coding [19], while larger v's are coded with Exponential-Golomb code of order-0 (EG-0 code) [20], which is a prefix-free code used in H.264 texture coding for the large coefficient values. Then the coefficient is updated to be its $n-1$ Least Significant Bits (LSBs) as

$$X_i^l = |X_i^l| - T \times v \tag{3}$$

The refinement pass of SPC* is similar to that of conventional SPC. For the coefficients that are previously found significant, the bits in bit-plane n are sent out as $ref(X_i)$ in Fig. 4. Then, update the just refined coefficient value to be its $n-1$ LSBs. Restricted by the input significance map, the SPC* has the same number of iterations as the SPC procedure of the key-frame. The X_i^* in Fig. 4 denotes the frame with updated coefficients after all SPC* iterations. So far, all the significant coefficients in the significant locations (Case I) are coded.

3.2.2. Differential-Sorting

Differential-sorting in Algorithm 2 is adopted to code the significant coefficients in the insignificant locations (Case IV). In differential-sorting, the threshold is set to be $T = 2^{n_{last}}$, and the sorting is enacted on the sets of the initial partition of the updated frame X_i^*. The significance tests are performed as in conventional SPC. The generated significance map conveys the positions of significant coefficients of X_i^* and are denoted by $diff\_map(X_i^*)$. For each significant coefficient, a sign indicator s is sent out.

$$s = \begin{cases} 0 & \text{if positive} \\ 1 & \text{if negative} \end{cases} \tag{4}$$

In addition, a value indicator v is also sent out to specify the n-th and higher bit-planes of the magnitude as in (2).

3.3. M-D-SPC Decoder

The bitstream structure and the separate decoding of individual frames are demonstrated in Fig. 5, where the conventional and differential SPC decoders are adopted for key- and correlated-frames, respectively. There are no correlation calculations in decoding. Instead, a 1-bit marker is included at the beginning of the self-data of each frame indicating the frame type (1: key-frame; 0: correlated-frame). The map & $sign(X_0)$ are the common data shared by the key-frame and N correlated-frames. The decoding of any frame is separate from the decoding of other frames, which enables random frame access and prevents inter-frame error propagation (given error-free common data).

In differential-SPC decoding, differential-sorting follows SPC* decoding to make sure all the significant coefficients are reconstructed, as shown in Fig. 4. Given an n_{last}, differential-SPC guarantees no PSNR loss, compared to its corresponding conventional SPC, because it encodes and reconstructs all the

Algorithm 2 Differential-sorting Encoding

1. **Initialization:**
   (a) Create a list LIS (List of Insignificant Sets) to contain "0"-labelled (called insignificant) sets.
   (b) Put descriptors of the initial sets onto LIS. These initial sets come from a non-overlapped partition of the updated frame X_i^* ($i \in \{1, 2, , N\}$), according to the adopted set partition method.
   (c) Set $n = n_{last}$.
2. **Sorting Pass:**
   (a) For each set on the LIS, do
      i. If maximum element in set is less than 2^n, send "0" to codestream.
      ii. Otherwise, send "1" to codestream, and do the following:
         A. If set has more than one element, then divide the set into subsets according to the pre-defined set partition rule, and add each subset to the end of the LIS.
         B. If set has one element, then send its sign indicator s and value indicator v according to (4) and (2), respectively, to the codestream.
         C. Remove the set from LIS.

coefficients with magnitude larger than or equal to $2^{n_{last}}$, just as conventional SPC does. Combined with reversible integer transforms, M-D-SPC is able to compress losslessly, which is required by some medical image applications.

3.4. Features

The coding framework of M-D-SPC differs from the all-intra-coding in that the significance map re-use and the correlation estimation (PSNR calculation between key frame and current frame) in the encoder use a key frame. However, it is also different from the inter-coding. On one hand, it encodes single frames at a time, while DPCM (differential pulse-code modulation) or its MC (motion compensation) form always uses a difference from a previous (or future) reconstructed frame(s) that is encoded also from a previous difference. On the other hand, any frame can be decoded separately. The M-D-SPC thus exhibits an *auxiliary key frame* coding structure.

This novel coding structure exploits inter-frame correlation to achieve higher coding efficiency and meanwhile maintains the beneficial features of intra coding. First, multiple correlated-frames can be coded in parallel because of no inter-frame interactions between them. Second, random frame access is assured, since the decoding of any frame is separate from that of other frames. Third, it maintains low computational complexity, because there is no special temporal de-correlation process, unlike the computation-consuming inter-frame prediction and 3D decomposition, and the encoder need not simulate decoding to create reference frames. Finally, it reduces the memory requirements by buffering only a key-frame and its significance map and sign data, instead of a group of reference frames.

4. Simulation Results

Simulations are based on SPIHT, which is a representative SPC algorithm that has been widely known and studied. Both C and MATLAB implementations of SPIHT are available online [21, 22]. The so-called M-D-SPIHT is simulated to compare with other coding schemes having the similar intra-decoding structure. One is the conventional 2D SPIHT, which is applied consecutively to every frame of the image sequence. The other is motion JPEG2000 (MJ2K), which codes each frame as a JPEG2000 [2, 3] image. The M-D-SPIHT uses the adaptive "1-key-N-correlated" frame structure. The PSNR serves as the correlation metric. We calculate the PSNR between the key-frame and the current frame, and compare it with the threshold T_{PSNR} (in dB) to decide the current frame to be coded with differential-SPIHT (differential-SPC) or conventional 2D SPIHT. If the current frame is coded with conventional 2D SPIHT, it will serve as the key-frame for the following frames, until a newer key-frame appears.

In simulation, all schemes adopt 3-level 2D DWT and arithmetic coding. As in the JPEG2000, the 5/3 [23] and 9/7 [17] filters are adopted for lossless and lossy coding, respectively. The comparisons are in terms of the coding efficiency in bpp (bits per pixel).

4.1. Compression Efficiency on Hyperspectral Images

Hyperspectral images have been used in geology, atmosphere, ecology, etc. domains as a mature technology. The state-of-the-art hyperspectral sensor Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) generates image consisting of 224 contiguous spectral bands covering wavelengths from 400 to 2500 nanometers (nm) with approximately 10 nm spectral resolution [24]. There are raw, radiance, and reflectance AVIRIS images. The raw images are converted to radiance images by means of the radiometric calibration, and the radiance images are further converted to reflectance images through correcting the atmospheric effects. The radiance or reflectance data are used most, depending on the applications. For most analyses, they give equivalent results. Hyperspectral images are typically intended for automatic analysis by computers. The distortion caused by lossy compression, which is typically designed based on human visual system, is not acceptable for most applications [25]. In our tests, we compress AVIRIS radiance data losslessly. Prior to compression, we resize the tested AVIRIS scene from 224 bands of size 614×512 to 224 bands of size 512×512, by discarding the 102 columns on the right of each band. The tested volume images are the resized scenes ($512 \times 512 \times 224$, 16-bit depth), and each spectral band is viewed as a frame. M-D-SPIHT exploits the spectral (inter-band) correlation.

Figure 6 presents the lossless compressed bit rates at different T_{PSNR} for *cuprite* scene 01. We can see that the M-D-SPIHT is most efficient at $T_{PSNR} = 45$ dB, at which there are 37 key-frames and 187 correlated-frames. Therefore, $T_{PSNR} = 45$ dB is used to compress the other hyperspectral images and the results are given in Table 3, where the 3D SPIHT [26] results are also provided as a reference. From the testing results, M-D-SPIHT outperforms the 2D SPIHT and MJ2K by saving about 9% and 11% bit rates, respectively. Although M-D-SPIHT costs about 15% more bits compared with 3D SPIHT, it is worthwhile to indicate that the lower bit rates achieved by 3D SPIHT is at the price of higher computational complexity and the

Sequence	SPIHT MD	2D	3D	MJ2K
cuprite sc04	6.27	6.93	5.63	7.10
moffet field sc01	6.16	6.85	5.71	7.57
moffet field sc03	6.27	6.93	5.08	6.29
jasper ridge sc01	7.17	7.70	5.94	7.89
jasper ridge sc03	7.15	7.68	5.92	7.86
average (normalization)	**6.60** (100%)	**7.22** (109.4%)	**5.66** (85.8%)	**7.34** (111.2%)

Table 3: Hyperspectral image lossless bit rates (bpp)

loss of temporal (band-crossing) flexibilities. The 3D SPIHT uses 3D coding units (spatial-temporal orientation trees) on the hierarchical structure of 3D wavelet transform, where the 3rd dimensional transform increases computational complexity. In addition, the group of frames (GOF) of 3D SPIHT requires much larger memory, and enables random access only at the GOF level. If applied to video, 3D SPIHT may cause latency for large GOF size. The auxiliary key frame coding framework of M-D-SPC gives a coding performance somewhere between inter and all-intra coding as can be expected.

4.2. Compression Efficiency on 3D/4D Medical Images

Medical image sets, such as images generated by computer tomography (CT) and functional Magnetic Resonance (MR), are increasingly used in diagnosis. The 3D volumetric images are two-dimensional image slices that represent cross sections of a subject. Four-dimensional (4D) medical images, which can be seen as a time series of 3D images, represent the live action of human anatomy and consume even larger amounts of resources in transmission and storage than 3D image data. In medical applications, lossless compression is extremely important, because any distortion may lead to an faulty diagnosis. In simulations, lossless compression are performed on both 3D and 4D medical images. M-D-SPIHT uses specific T_{PSNR} for every subject. All the tested medical images are of 8-bit depth.

For 3D medical images, every 2D slice is seen as a frame, and the M-D-SPIHT exploits the dependency between slices. The tested 3D CT images are slices 1-64, and 3D MR images are slices 1-48, all of size 512×512. Simulation results are provided in Table 4, where M-D-SPIHT outperforms 2D SPIHT and MJ2K in terms of saving compressed bit rates by about 11% and 8%, respectively.

4D medical images are time series of 3D volume images. Since they are typically temporally smooth, M-D-SPIHT exploits the inter-volume (temporal) dependency. Assume that a 2D slice F_n in volume n is a key-frame coded with 2D SPIHT. If the corresponding 2D slice F_{n+1} in volume $n+1$ is highly correlated with F_n, then F_{n+1} will be coded with differential SPIHT (differential-SPC) referring to the map of F_n; otherwise, F_{n+1} will be coded with 2D SPIHT and serve as a new key-frame. Then code the corresponding slice F_{n+2} in volume $n+2$, with differential SPIHT referring the map of F_n or F_{n+1}, or with 2D SPIHT and serve as a new key-frame. Simulations are performed on 4D images given in Table 5, and the compressed bit rates are provided in Table 6. On average, M-D-SPIHT saves compressed bit rates by 2% and 4%, respectively, compared with 2D SPIHT and MJ2K.

Sequence	SPIHT MD (T_{PSNR})	2D	3D	MJ2K
CT_aperts	1.16 (33 dB)	1.28	0.92	1.22
CT_carotid	1.88 (33 dB)	2.04	1.56	1.96
CT_skull	3.01 (27 dB)	3.37	2.34	3.32
CT_wrist	1.97 (35 dB)	2.21	1.48	2.07
MR_liver_t1	3.05 (33 dB)	3.32	2.34	3.25
MR_ped_chest	2.63 (40 dB)	3.05	1.93	3.00
MR_sag_head	2.65 (31 dB)	2.88	2.19	2.91
average (normalization)	**2.34** (100%)	**2.59** (110.7%)	**1.82** (77.8%)	**2.53** (108.1%)

Table 4: 3D medical image lossless bit rates (bpp)

4.3. Compression Efficiency on Video

Lossy compression is performed for video sequences *foreman* (352 × 288, 300 frames, 30 Hz), *salesman* (360 × 288, 449 frames, 30 Hz), and *hall monitor* (352 × 288, 300 frames, 30 Hz).As in hyperspectral image compression, a simulation is first performed on *foreman* at different T_{PSNR}'s, and the lowest bit rate is achieved with T_{PSNR} = 27 dB, which is adopted by the

image	type	volumes	slices	size
volunteer4d	N/A	001 - 016	001 - 016	128 × 120
ct4d	CT	001 - 018	001 - 192	256 × 256
siem	fMRI	001 - 112	001 - 016	64 × 64

Table 5: Information of the tested 4D medical images

Sequence	SPIHT MD (T_{PSNR})	SPIHT 2D	MJ2K
volunteer4d	3.28 (39 dB)	3.39	3.35
ct4d	3.17 (41 dB)	3.33	3.24
siem	4.81 (37 dB)	4.81	5.13
average (normalization)	**3.75** (100%)	**3.84** (102.4%)	**3.91** (104.3%)

Table 6: 4D medical image lossless bit rates (bpp)

M-D-SPIHT for the video simulation in this sub-section. With $T_{PSNR} = 27$ dB, the M-D-SPIHT coded *hall monitor* has 49 key frames and 251 correlated frames; *salesman* has 41 key frames and 408 correlated frames. The rate-distortion performance of tested video sequences are provided in Fig. 7. As a reference, H.264 [27] all-intra-coding results are also given in the figures. Tests are with the H.264 reference software JM 11.0 [28] default configuration, including CABAC (context-adaptive binary arithmetic coding), rate-distortion optimization, and other advanced tools. It is worthwhile to indicate that the H.264 all-intra-coding is a framework different from the other tested methods. It is based on DCT, instead of DWT; its spatial-scalability and rate-control are based on a layered structure, which is not as flexible as the DWT based methods. H.264 exploits the spatial correlation efficiently at the cost of increased complexity.

From the test results, the M-D-SPIHT outperforms the conventional 2D SPIHT. Actually, with a properly chosen T_{PSNR}, the conventional 2D SPIHT is the lower performance boundary of M-D-SPIHT. For example, the video sequence *calendar and train* includes complex movements, and the maximum inter-frame PSNR is about 23 dB only. With the $T_{PSNR} = 27$ dB, there are no correlated-frames to be differential SPIHT coded, and, therefore, the M-D-SPIHT performs as the conventional 2D SPIHT. Generally speaking, M-D-SPIHT performs better for sequences with higher inter-frame correlation, i.e., with slower movements. For sequences with similar movements, such as the sequences *hall monitor* and *salesman*, which are all with still camera, the degree of background complexity also decides the performance of MD-SPIHT. Complex texture requires many bits to encode, while simple texture requires much less. Therefore, M-D-SPIHT is expected to show more improvement in efficiency in encoding complex, nearly static background than with simple background texture. The results verify this expectation. As the background complexity increases (from *hall monitor* to *salesman*), the performance of the M-D-SPIHT increases accordingly. For *salesman* at 30 dB recovery PSNR, M-D-SPIHT reduces the overall bit rate to about 50% of that of the 2D SPIHT or MJ2K.

4.4. Computational Complexity Evaluation

In Fig. 7, encoding times (in seconds) are provided for some Rate-Distortion points for M-D-SPIHT, 2D-SPIHT, and H.264 all-intra-coding. These times are achieved with Intel Core 2 Duo CPU, 2.95 GB RAM. The M-D-SPIHT program was not optimized to a commercial application level, and these times are shown just to give an indication of the method's speed. We can see that the computational complexity of both M-D-SPIHT and 2D-SPIHT is much lower than H.264 all-intra-coding. The time spent by M-D-SPIHT includes the correlation estimation (PSNR calculation between current and key frames) time and encoding operation time. The correlation estimation takes 5.8 seconds and 8.8 seconds for hall monitor and salesman sequences, respectively. The encoding operations of M-D-SPIHT well maintain the low computational complexity of conventional SPIHT. By exploiting the efficiency of correlation estimation (as future work), a more computationally efficient M-D-SPC system can be expected. The decoding process of M-D-SPC is nearly symmetric to encoding, but is faster, as it does not calculate correlations or the top bit plane (n_{max}).

5. Conclusion

A novel image sequence coding system M-D-SPC is presented in this paper. An image sequence is split into key-frames and correlated-frames, which are coded with conventional 2D SPC and differential-SPC, respectively. The differential-SPC encoding of a correlated-frame refers to the significance map of its previous SPC coded key-frame to exploit inter-frame correlation; the decoding of any frame (key- or correlated-frame), involving only the common data (the significance map and sign information) and the self data of that specific frame, is separate from the decoding of other frames. Such *auxiliary key frame* coding framework improves coding efficiency and meanwhile maintains the beneficial features of all-intra-coding such as computational simplicity, parallel processing, non-error-propagation (inter-frame), and random frame access. Simulations are performed on SPIHT, but the proposed auxiliary key frame coding technique can be utilized with other coders that generate significance maps, such as SPECK, SBHP, EZBC, and EZW. From the SPIHT-based simulation results on hyperspectral images, 3D/4D medical images, and video sequences, M-D-SPIHT significantly outperforms the all-intra-coding methods, such as 2D SPIHT and motion JPEG2000, especially for sequences with high inter-frame correlations. We consider M-D-SPC to be an attractive alternative for so-called *motion coding*, e.g., motion JPEG and motion JPEG2000, which uses all-intra-coding framework.

References

[1] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, December, 1993.

[2] ISO/IEC, "Information Technology—JPEG 2000 Image Coding System, Part-1: Core Coding System," *ISO/IEC 15444-1*, 2000.

[3] ISO/IEC, "Information Technology—JPEG 2000 Image Coding System, Part-2: Core Coding System," *ISO/IEC 15444-2*, 2001.

[4] CCSDS, "Image data compression." CCSDS-122.0-B-1 Blue Book, Nov. 2005.

[5] P. S. Yeh, P. Armbruster, A. Kiely, B. Masschelein, G. Moury, C. Schaefer, and C. Thiebaut, "The new CCSDS image compression recommendation," in *Proc. IEEE Aerospace Conference*, pp. 4138–4145, March, 2005.

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

[7] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 11, pp. 1219–1235, November, 2004.

[8] J. Andrew, "A simple and efficient hierarchical image coder," in *Proc. Int. Conf. on Image Processing (ICIP)*, vol. 3, pp. 658–661, October, 1997.

[9] C. Chrysafis, A. Said, A. Drukarev, A. Islam, and W. A. Pearlman, "SBHP-a low complexity wavelet coder," in *IEEE Int. Conf. Acoust., Speech Signal Processing (ICASSP)*, vol. 6, pp. 2035–2038, June, 2000.

[10] S. T. Hsiang, *Highly scalable subband/wavelet image and video coding.* PhD thesis, Rensselaer Polytechnic Institute, 2002.

[11] W. A. Pearlman and A. Said, "Set Partition Coding: Part I of Set Partition Coding and Image Wavelet Coding Systems," *Foundations and Trends in Signal Processing*, vol. 2, no. 2, pp. 95–180, 2008.

[12] W. A. Pearlman and A. Said, "Image Wavelet Coding Systems: Part II of Set Partition Coding and Image Wavelet Coding Systems," *Foundations and Trends in Signal Processing*, vol. 2, no. 3, pp. 181–246, 2008.

[13] B. Wang, Y. Wang, I. Selesnick, and A. Vetro, "Video coding using 3D dual-tree wavelet transform," *Journal on Image and Video Processing*, vol. 2007, no. 1, 2007.

[14] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, "3-D embedded subband coding

with optimal truncation (3-D ESCOT)," *J. Applied and Computational Harmonic Analysis*, vol. 10, no. 5, pp. 290–315, 2001.

[15] ISO/IEC, "Information Technology—JPEG 2000 Image Coding System, Part-3: Motion JPEG 2000," *ISO/IEC 15444-3*, 2001.

[16] Y. Hu and W. A. Pearlman, "Differential-SPIHT for image sequence coding," in *IEEE Int. Conf. Acoust., Speech Signal Processing (ICASSP)*, pp. 894–897, March, 2010.

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

[18] W. B. Pennebaker and J. L. Mitchell, *JPEG still image data compression standard*. Kluwer Academic Publishers, 1993.

[19] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, no. 6, p. 540, 1987.

[20] J. Teuhola, "A compression method for clustered bit-vectors," *Information Processing Letters*, vol. 7, no. 6, pp. 308–311, 1978.

[21] QccPackSPIHT (SPIHT in C), `http://qccpack.sourceforge.net`.

[22] SPIHT in MATLAB, `http://www.cipr.rpi.edu/research/SPIHT/`.

[23] A. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo, "Lossless image compression using integer to integer wavelet transforms," in *Proc. Int. Conf. on Image Processing (ICIP)*, vol. 1, pp. 596–599, October, 1997.

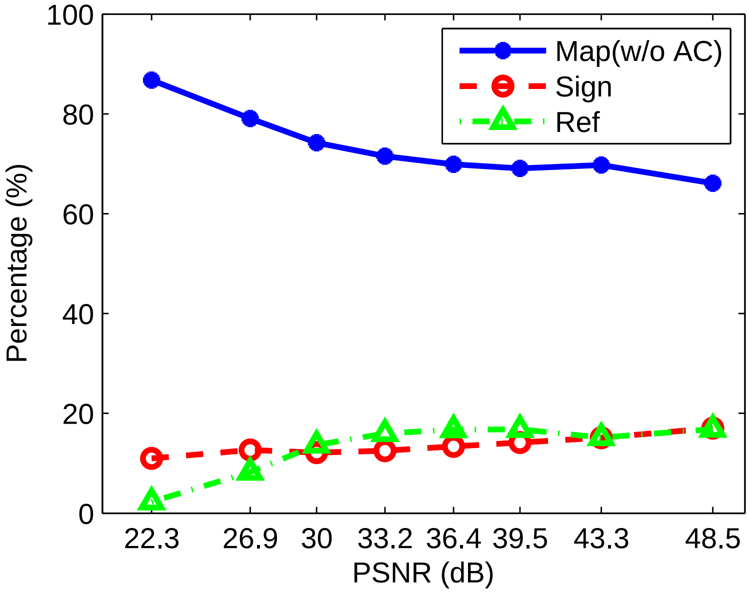
[24] AVIRIS Home Page, http://aviris.jpl.nasa.gov/.

[25] J. Mielikainen, "Lossless compression of hyperspectral images using lookup tables," *IEEE signal processing letters*, vol. 13, no. 3, pp. 157–160, 2006.

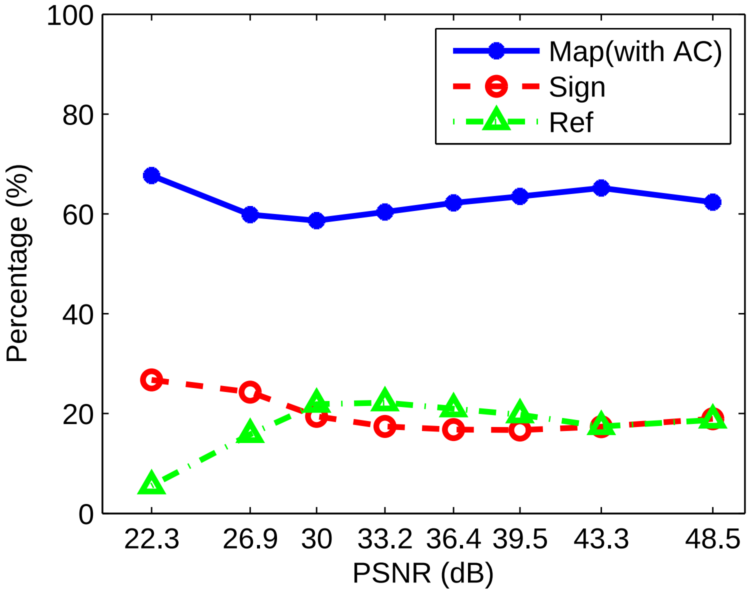
[26] S. Cho and W. A. Pearlman, "Error-resilient video coding with improved 3D SPIHT and error concealment," in *Proceedings of SPIE*, vol. 5022, pp. 125–136, 2003.

[27] ITU-T and I. J. 1, "Advanced video coding for generic audio-visual services," *ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, May 2003.

[28] H.264/AVC Reference Software JM 11.0, http://iphome.hhi.de/suehring/tml/download/old_jm/.



(a) Without arithmetic coding.



(b) Significance map with arithmetic coding

Figure 2: Percentages of the three kinds (map, sign, and refinement) of SPIHT output data for 8-bit 512×512 grayscale Lena (3-level wavelet transform with 9/7 filters).

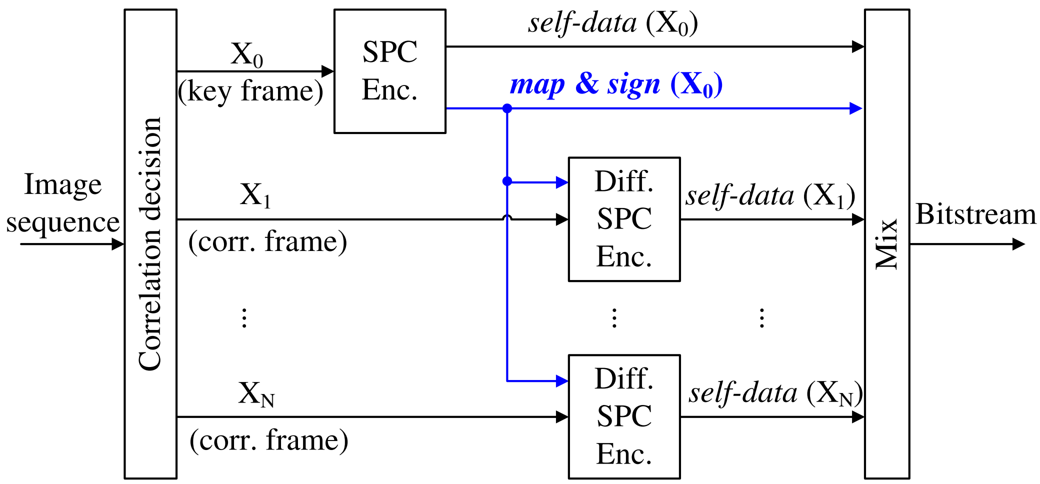


Figure 3: The M-D-SPC encoder, where the correlated-frames X_i, $i \in \{1, 2, ..., N\}$, are coded in parallel.

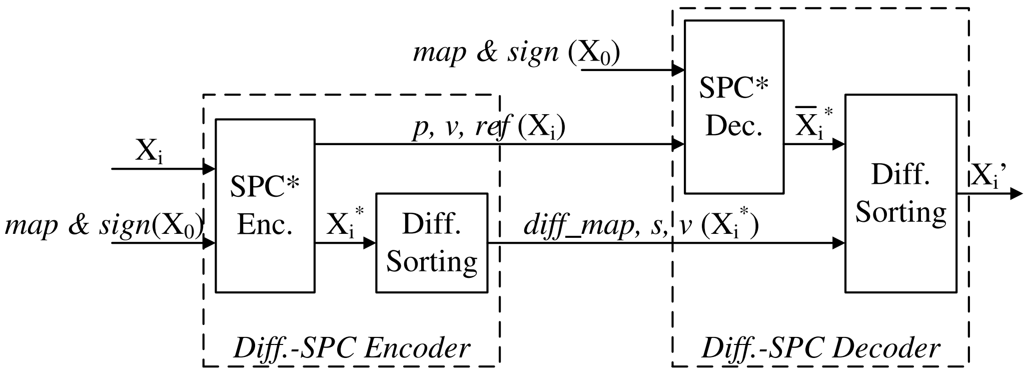


Figure 4: Differential-SPC — the coding procedure for a correlated-frame X_i ($i \in \{1, 2, ..., N\}$), re-using the significance map and sign of the key-frame X_0.

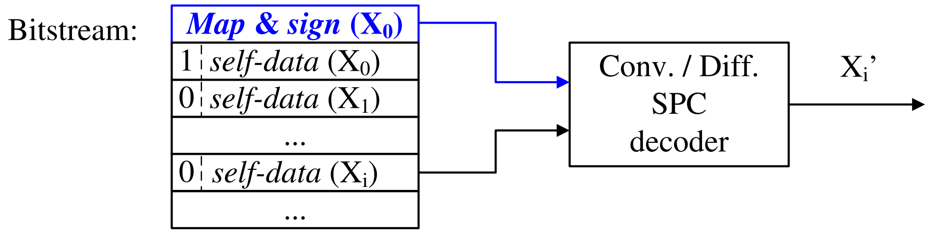


Figure 5: M-D-SPC decoding. The *map* & $sign(X_0)$ are the common data shared by $X_i, i \in \{0, 1, 2, ..., N\}$. The first bit of self-data indicates frame and decoder type. If 1, key-frame and conventional SPC decoder; if 0, correlated frame and differential SPC decoder.

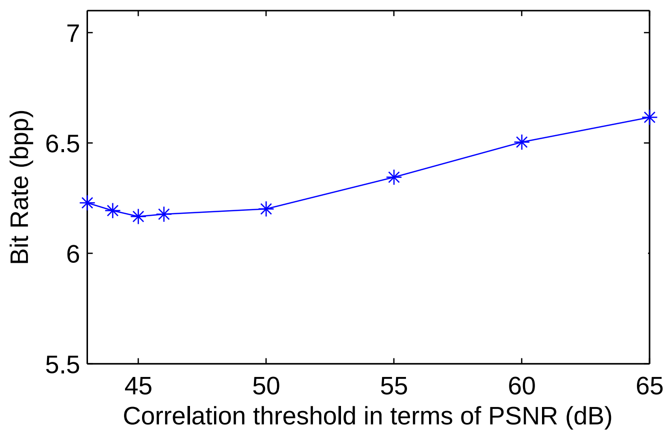
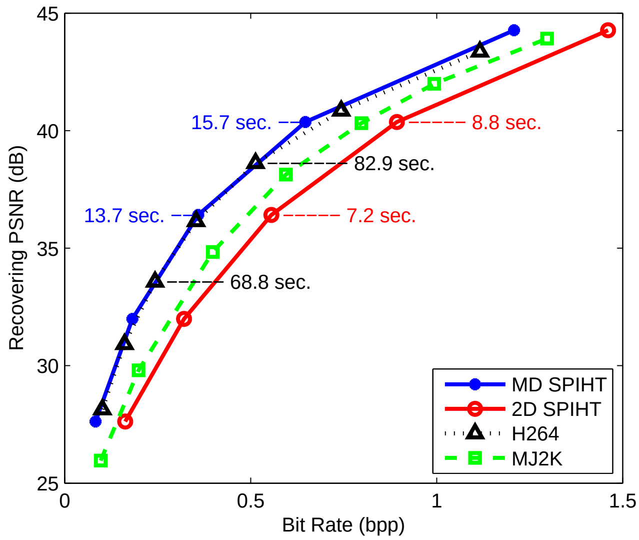
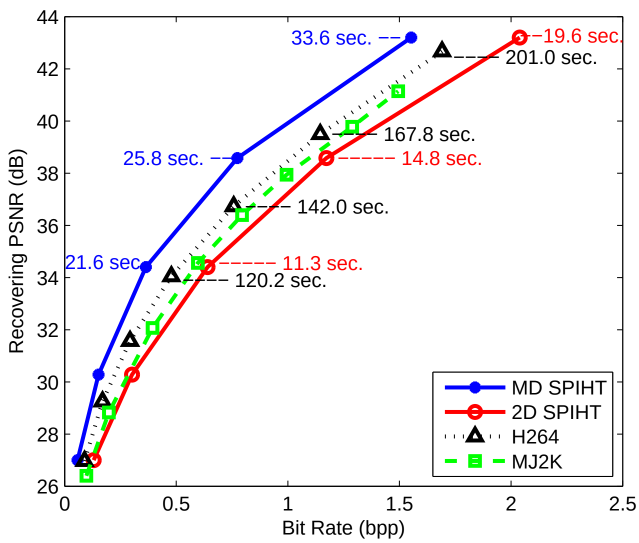


Figure 6: Lossless bit rates at different correlation thresholds T_{PSNR} for *cuprite* scene 01



(a) Hall Monitor (Grayscale, 352x288, 300 frames @ 30 Hz)



(b) Salesman (Grayscale, 360x288, 449 frames @ 30 Hz)

Figure 7: Rate-distortion (R-D) performance on video sequences of M-D-SPIHT, conventional 2D SPIHT, H.264 all-intra-coding (H.264), and motion JPEG 2000 (MJ2K). The encoding times for some R-D points are provided. Those of M-D-SPIHT include the correlation estimation times of 5.8 and 8.8 seconds for hall monitor and salesman, respectively.