Low-Complexity Multiple Description Coding of Video Based on Block 3D-transforms

Andrey Norkin*, Atanas Gotchev, Karen Egiazarian, Jaakko Astola

Institute of Signal Processing, Tampere University of Technology
P.O.Box 553, FIN-33101 Tampere, FINLAND
email: firstname.familyname@tut.fi

January 8, 2007

Abstract

The paper presents a multiple description (MD) video coder based on three-dimensional (3D) transforms. Two balanced descriptions are created from a video sequence. In the encoder, video sequence is represented in a form of coarse sequence approximation (shaper) included in both descriptions and residual sequence (details) which is split between two descriptions. The shaper is obtained by block-wise pruned 3D-DCT. The residual sequence is coded by 3D-DCT or hybrid 3D-transform. The coding scheme is targeted to mobile devices. It has low computational complexity and improved robustness of transmission over unreliable networks. The coder is able to work at very low redundancies. Although the coding scheme is simple, yet it outperforms some MD coders based on motion-compensated prediction, especially in the low-redundancy region. The margin is up to 3 dB for reconstruction from one description.

1 Introduction

Nowadays, video is more often being encoded in mobile devices and transmitted over less reliable wireless channels. Traditionally, the objective in video coding has been to achieve high compression, which was

*Corresponding author

attained with the cost of increasing encoding complexity. However, portable devices, such as camera phones, still lack enough computational power and are energy-consumption constraint. Besides, a highly compressed video sequence is more vulnerable to transmission errors, which are often present in wireless networks due to multi-path fading, shadowing, and environmental noise. Thus, there is a need of a low-complexity video coder with acceptable compression efficiency and strong error-resilience capabilities.

Lower computational complexity in transform-based video coders can be achieved by properly addressing the motion estimation problem, as it is the most complex part of such coders. For the case of high and moderate frame rates ensuring smooth motion, motion-compensated (MC) prediction can be replaced by a proper transform along the temporal axis to handle the temporal correlation between frames in the video sequence. Thus, the decorrelating transform adds one more dimension, becoming a 3D one, and if a low complexity algorithm for such a transform exists, savings in overall complexity and power consumption can be expected compared to traditional video coders [8], [25], [7], [3]. Discrete cosine transform (DCT) has been favored for its very efficient 1D implementations. As DCT is a separable transform, efficient implementations of 3D-DCT can be achieved too [4], [7], [25]. Previous research on this topic shows that simple (baseline) 3D-DCT video encoder is three to four times faster than the optimized H.263 encoder [12], for the price of some compression efficiency loss, quite acceptable for portable devices [15].

A 3D-DCT video coder is also advantageous in error resilience. In MC-based coders, the decoding error would propagate further into subsequent frames until the error is corrected by intra-coded frame. The error could also spread over the bigger frame area because of motion-compensated prediction. Unlike MD-based coders, 3D-DCT video coders enjoy no error propagation in the subsequent frames. Therefore, we have chosen the 3D-DCT video coding approach for designing a low-complexity video coder with strong error resilience.

A well-known approach addressing the source-channel robustness problem is so-called multiple description coding (MDC) [11]. Multiple encoded bitstreams, called descriptions, are generated from the source information. They are correlated and have similar importance. The descriptions are independently decodable at the basic quality level and, when several descriptions are reconstructed together, improved quality is obtained. The advantages of MDC are strengthened when MDC is connected with multi-path (multi-channel) transport [2]. In this case, each bitstream (description) is sent to the receiver over a separate independent path (channel), which increases the probability of receiving at least one description.

Recently, a great number of multiple description (MD) video coders have appeared, most of them based on MC prediction. However, MC-based MD video coders risk having a mismatch between the prediction loops in the encoder and decoder when one description is lost. The mismatch could propagate further in the consequent frames if not corrected. In order to prevent this problem, three separate prediction loops are used at the encoder [21] to control the mismatch. Another solution is to use a separate prediction loop for every

description [1], [29]. However, both approaches decrease the compression efficiency and the approach in [21] also leads to increased computational complexity and possibly to the increased power consumption. A good review of MDC approaches to video coding is given in [30]. A number of MD and error resilient video coders based on 3D-transforms (e.g. wavelets, lapped orthogonal transforms (LOT), DCT) have been proposed [17], [13], [27], [33].

In this work, we investigate an MD video coder (3D-2sMDC), which does not exploit motion compensation as initially proposed in [18]. Using 3D-transform instead of motion compensated prediction reduces the computational complexity of the coder, meanwhile eliminating the problem of mismatch between the encoder and decoder. The proposed MD video coder is a generalization of our 2-stage image MD coding approach [19] to coding of video sequences [18]. Our coder has balanced computational load between the encoder and decoder. It is also able to work at a very low redundancy introduced by MD coding. Despite the fact that 3D-DCT video coders have usually lower compression ratio than MC-based video coders [15], our coder outperforms some MD video coders based on motion-compensated prediction. The margin is up to 3 dB in the low redundancy region for the reconstruction from one description.

The paper is organized as follows. Section 2 overviews in general the encoding and decoding process while Section 3 describes each block of the proposed scheme in detail. Section 4 gives the analysis of the proposed scheme and Section 5 discusses its computational complexity. Section 6 offers the packetization strategy, Section 7 presents the simulation results while Section 8 concludes the paper.

2 General coding scheme

2.1 Encoder operation

In our scheme, video sequence is coded in two stages as shown in Fig. 1. In the first stage (dashed rectangle), a coarse sequence approximation is obtained and included in both descriptions. The second stage produces enhancement information, which has higher bitrate and is split between two descriptions. The idea of the method is to get a coarse signal approximation which is the best possible for the given bitrate, while decorrelating the residual sequence as much as possible.

The operation of the proposed encoder is described in the following. First, a sequence of frames is split into groups of 16 frames. Each group is split into 3D cubes of size $16\times16\times16$. 3D-DCT is applied to each cube. The lower-frequency DCT coefficients in the $8 \times 8 \times 8$ cube are coarsely quantized with quantization step Q_s and entropy-coded (see Fig. 2(a)) composing the shaper, other coefficients are set to zero. Inverse quantization is applied to coefficients followed by the inverse 3D-DCT. An optional deblocking filter (Fig. 1) serves to

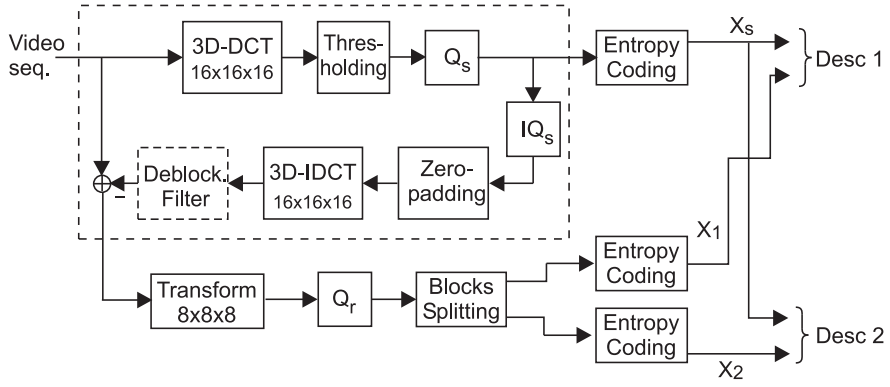
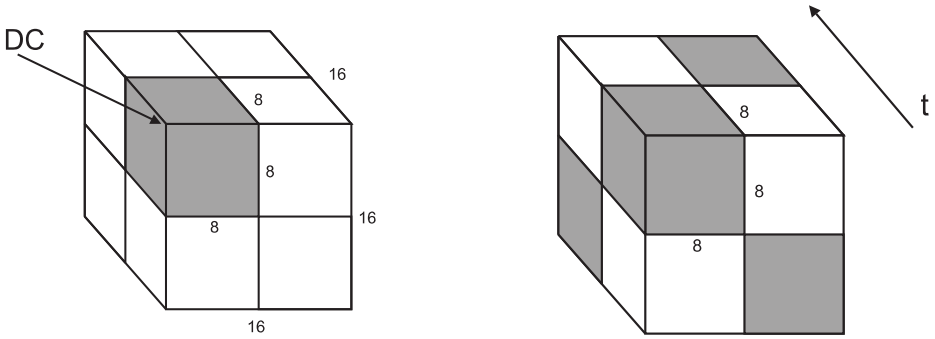


Figure 1: Encoder scheme.



(a) (b)

Figure 2: Coding patterns: (a) 3D-DCT cube for shaper coding: only coefficients in the gray volumes are coded, other coefficients are set to zero; (b) Split pattern for volumes of a residual sequence: "gray" - description 1; "white" - description 2.

remove the block edges in spatial domain. Then, the sequence reconstructed from shaper is subtracted from the original sequence to get the residual sequence.

The residual sequence is coded by a 3-D block transform and transform coefficients are finely quantized with a uniform quantization step (Q_r), split into two parts in a manner shown in Fig. 2(b), and entropy-coded. One part together with the shaper forms *Description 1*, while the second part combined again with the shaper forms *Description 2*. Thus, each description consists of a coarse sequence approximation (*shaper*) and *half* of the transform volumes of the residual sequence.

The shaper is included in both descriptions to facilitate successful reconstruction when one description is lost. Thus, the *redundancy* of the proposed coder is only determined by the shaper quality, which is controlled by the shaper quantization step Q_s. Larger quantization step corresponds to lower level of redundancy and lower quality of side reconstruction (reconstruction from only one description). Alternatively, smaller quantization

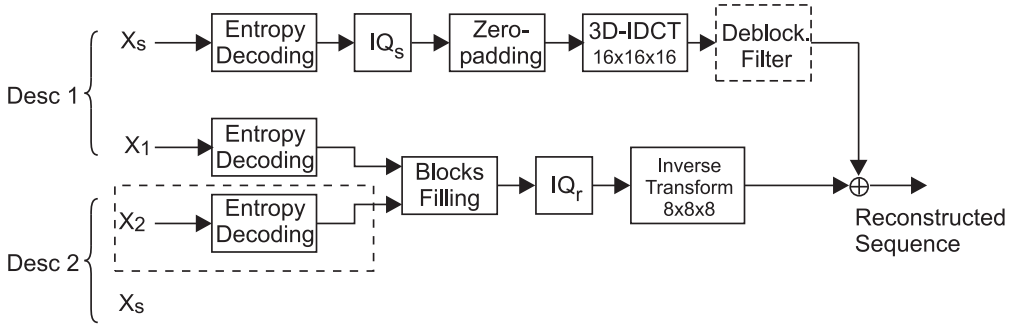


Figure 3: Decoder scheme. Central reconstruction. Side reconstruction (description 1) when the content of the dashed rectangle is removed.

step results in higher quality side reconstruction. The *quality* of a two-channel reconstruction is controlled by the quantization step Q_r used in the coding of the residual sequence. As the residual volumes are divided into two equal parts, the encoder produces balanced descriptions both in terms of PSNR and bitrate.

2.2 Decoder operation

The decoder (Fig. 3) operates as follows. When the decoder receives two descriptions, it extracts the shaper (X_s) from one of the descriptions. Then, the shaper is entropy-decoded and inverse quantization is applied. The $8 \times 8 \times 8$ volume of coefficients is zero-padded to the size $16 \times 16 \times 16$, and inverse DCT is applied. The deblocking filter is applied if it was applied in the encoder.

In case of *central* reconstruction (reconstruction from two descriptions), each part of the residual sequence (X_1 and X_2) is extracted from the corresponding description and entropy decoded. Then, volumes of the corresponding descriptions are coded and combined together as in Fig. 2(b). The inverse quantization and inverse transform (IDCT or Hybrid inverse transform) are applied to coefficients and the residual sequence is added to the shaper to obtain the reconstruction of the original sequence.

As *side* reconstruction, we call the reconstruction from one description, e.g. *Description 1* (reconstruction from *Description 2* is symmetrical). The side decoder scheme can be obtained from Fig. 3 if content of the dashed rectangle is removed. In this case, the shaper is reconstructed from its available copy in *Description 1*. The residual sequence, however, has only half of the coefficient volumes (X_1). The missing volumes X_2 are simply filled with zeros. Then, the decoding process is identical to that of the central reconstruction. As the residual sequence has only half of the coefficient volumes, side reconstruction has lower, however, still acceptable quality. For example, sequence "Silent voice" coded at 64.5 kbps with 10% redundancy can be reconstructed with PSNR = 31.49 dB from two descriptions, and 26.91 dB from one description (see Table

2).

3 Detailed system description

3.1 The coarse sequence approximation

The idea of the first coding stage is to concentrate as much information as possible into the shaper within strict bitrate constraints. We would also like to reduce artifacts and distortions appearing in the reconstructed coarse approximation. The idea is to reduce spatial and temporal resolution of the coarse sequence approximation in order to code it more efficiently with lower bitrate [6]. Then, the original resolution sequence can be reconstructed by interpolation as a post-processing step. A good interpolation and decimation method would concentrate more information in the coarse approximation and correspondingly make the residual signal closer to white noise. A computationally inexpensive approach is to embed interpolation in the 3D-transform.

The downscaling factor for the shaper was chosen equal to two in both spatial and temporal directions. The proposed scheme is able to use other downscaling factors equal to powers of two. However, the downscaling factor two has been chosen as the one producing the best results for QCIF and CIF resolution. To reduce computational complexity, we combine downsampling with forward transform (and backward transform with interpolation). Thus, the original sequence is split into volumes of size $16 \times 16 \times 16$, and 3D-DCT is applied to each volume. Pruned DCT is used in this stage that allows to reduce computational complexity (see Fig. 2(a)). Computational complexity can be decreased even further when the mobile device has hardware implementation of DCT.

Only $8 \times 8 \times 8$ cubes of low-frequency coefficients in each $16 \times 16 \times 16$ coefficient volume are used; other coefficients are set to zero (see Fig. 2(a)). The AC coefficients of the $8 \times 8 \times 8$ cube are uniformly quantized with quantization step Q_s. DC coefficients are quantized with the quantization step Q_{DC}.

In the $8 \times 8 \times 8$ volume, we use coefficient scanning described in [31], which is similar to a 2-D zigzag scan. Although there exist more advanced types of quantization and scanning of 3-D volumes exist [8], [5], we have found that simple scanning performs quite well for representing the coarse sequence approximation.

An optional deblocking filter may be used to eliminate the blocking artifacts caused by quantization and coefficient thresholding.

DC coefficients of transformed shaper volumes are coded by DPCM prediction. The DC coefficient of a volume is predicted from the DC coefficient of the temporally preceding volume. As shaper is included in

both descriptions, there is no mismatch between the states of the encoder and decoder in case when one description is lost.

First, the DC coefficient prediction errors and the AC coefficients undergo zero run-length (RL) encoding. It combines runs of successive zeros and the following non-zero coefficients into two-tuples where the first number is the number of leading zeros, and the second number is the absolute value of the first non-zero coefficient following the zero-run.

Variable-length encoding is implemented as a standard Huffman encoder similar to one in H.263 [12]. The codebook has the size 100 and is calculated for the two-tuples which are the output of RL-coding. All values exceeding the range of the codebook are encoded with an "escape" code followed by the actual value.

Two different codebooks are used: one for coding the shaper and another for coding the residual sequence.

3.2 Residual sequence coding

The residual sequence is obtained by subtracting the reconstructed shaper from the original sequence. As the residual sequence consists of high-frequency details, we do not add any redundancy in this stage. The residual sequence is split into groups of 8 frames in such a way that two groups of 8 frames correspond to one group of 16 frames obtained from the coarse sequence approximation. Each group of 8 frames undergoes block 3D-transform. The transform coefficients are uniformly quantized with quantization step Q_r and split between two descriptions in a pattern shown in Fig. 2(b).

Two different transforms are used in this work to code the residual sequence. The first transform is 3D-DCT and the second is hybrid transform. The hybrid transform consists of a lapped orthogonal transform (LOT) [16] in vertical and horizontal directions, and DCT in temporal direction. Both DCT and hybrid transform produce $8 \times 8 \times 8$ volumes of coefficients which are split between two descriptions. Using LOT in spatial domain of the hybrid transform smoothes blocking artifacts when reconstructing from one description. In this case, LOT spatially spreads the error caused by loosing transform coefficient blocks. Although LOT could be applied in the temporal direction to reduce blocking artifacts in temporal domain, we avoid to use it because of additional delay that it introduces in the encoding and decoding process.

As will be demonstrated in Section 7, the hybrid transform outperforms DCT in terms of PSNR and visual quality. Moreover, using LOT in spatial dimensions gives better visual results compared to DCT. However, blocking artifacts introduced by coarse coding of the shaper are not completely concealed by the residual sequence coded with the hybrid transform. These artifacts impede efficient compression of the residual sequence by the hybrid transform. Therefore, the *deblocking filter* is applied to the reconstructed shaper (see Fig. 1) prior to subtracting it from the original sequence. In the experiments, we use the deblocking filter

from H.263+ standard [12].

In residual sequence coding, transform coefficients are uniformly quantized with quantization step Q_r. DC prediction is not used in the second stage to avoid the mismatch between the encoder and decoder if one description is lost. The scanning of coefficients is 3D-zigzag scanning [31]. The entropy coding is RL coding followed by Huffman coding with the codebook different that the one used in coding the coarse sequence approximation.

4 Scheme analysis

4.1 Redundancy and reconstruction quality

Let D_0 denote the *central distortion* (distortion when reconstructing from two descriptions), and *side distortions* (when reconstructing from only one description) are D_1 and D_2. In case of balanced descriptions, side distortions are equal, i.e. $D_1 = D_2$. Denote as D_s the distortion of the video sequence reconstructed only from the shaper. Consider 3D-DCT coding of the residual sequence. The side distortion D_1 is formed by the blocks, half of which are coded with the distortion D_0, and half with distortion of shaper D_s. Here we assume that all blocks of the *Description 1* have the same expected distortion as the blocks of *Description 2*. Consequently,

$$D_1 = \frac{1}{2}(D_s + D_0). \tag{1}$$

Expression (1) can also be used in case when hybrid transform is used for coding the residual. As LOT is by definition an orthogonal transform, mean-squared error distortion in spatial domain is equal to distortion in the transform domain. Side distortion in transform domain is determined by loosing half of the transform coefficient blocks. Thus, expression (1) is also valid for hybrid transform. It is obvious that D_s depends on the bitrate R_s allocated to shaper. Then, we can write (1) as

$$D_1(R_s, R_r) = \frac{1}{2}(D_s(R_s) + D_0(R_r, R_s)), \tag{2}$$

where R_r is the bitrate allocated for coding the residual sequence and R_s is bitrate allocated to shaper. However, for higher bitrates, $D_s(R_s) \gg D_0(R_r)$. Thus, one can say that D_1 mostly depends on R_s.

Redundancy ρ of the proposed scheme is the bitrate allocated to shaper, $\rho = R_s$. Shaper bitrate R_s and side reconstruction distortion D_1 depend on the quantization step Q_s and characteristics of the video sequence. Central reconstruction distortion D_0 is mostly determined by the quantization step Q_r.

Thus, the encoder has two control parameters: Q_s and Q_r. Changing Q_r, the encoder controls the central distortion. Changing Q_s, the encoder controls redundancy and side distortion.

4.2 Optimization

The proposed scheme can be optimized for changing channel behavior. Denote p the probability of the packet loss and R the target bitrate. Then, in case of balanced descriptions we have to minimize

$$2p(1-p)D_1 + (1-p)^2 D_0 \tag{3}$$

subject to

$$2R_s + R_r \leq R. \tag{4}$$

Taking (1) into consideration, (3) can be transformed to the unconstrained minimization task

$$J(R_s, R_r) = p(1-p)(D_s(R_s) + D_0(R_s, R_r)) + (1-p)^2 D_0(R_s, R_r) + \lambda(2R_s + R_r - R). \tag{5}$$

Unfortunately, it is not feasible to find the distortion-rate functions $D_0(R_s, R_r)$ and $D_s(R_s)$ in real-time to solve the optimization task. However, the distortion-rate (D-R) function of a 3D-coder can be modeled as

$$D(R) = b2^{-aR} - c, \tag{6}$$

where a,b, and c are parameters which depend on characteristics of the video sequence. Hence,

$$D_s(R_s) = b2^{-aR_s} - c. \tag{7}$$

Assuming that the source is successfully refinable in regard to the squared-error distortion measure (this is true, for example, for i.i.d Gaussian source [10]) we can write

$$D_0(R_s, R_r) = b2^{-a(R_s+R_r)} - c. \tag{8}$$

Then, substituting (7) and (8) into (5) and differentiating the resulting Lagrangian with respect to R_s, R_f, and λ, we can find a closed form solution of the optimization task (5). The obtained optimal R_s and R_r are

$$R_s^* = \frac{1}{2}R + \frac{1}{2a}\log_2(p)$$

$$R_r^* = -\frac{1}{a}\log_2(p), \tag{9}$$

where R_s^* and R_r^* are rates of the shaper and residual sequence.

Hence, optimal redundancy ρ^* of the proposed scheme is

$$\rho^* = R_s^* = \frac{1}{2}R + \frac{1}{2a}\log_2(p). \tag{10}$$

Transform	Pruned $16 \times 16 \times 16$	3-D VR $16 \times 16 \times 16$	RCF $16 \times 16 \times 16$	3-D VR $8 \times 8 \times 8$	RCF $8 \times 8 \times 8$
Mults/point	2.625	3.5	6	2.625	4.5
Adds/point	6.672	15.188	15.188	10.875	10.875
Mults+adds/point	9.297	18.688	21.188	13.5	15.375

Table 1: Operations count for 3D-DCT II. Comparison of algorithms.

Optimal redundancy ρ^* depends on the target bitrate R, probability of packet loss p, and parameter a of the source D-R function. It does not depend on D-R parameters b and c. We have found that parameter a usually takes similar values for video sequences with the same resolution and frame rate. Thus, one does not need to estimate a in real-time. Instead, one can use a typical value of a to perform optimal bit allocation during encoding. For example, sequences with CIF resolution and 30 frames per second usually have the value of a between 34 and 44 for bitrates under 1.4 bits per pixel.

One can notice that for values $R \leq -\frac{1}{a}\log_2(p)$, optimal redundancy ρ^* is zero or negative. For these values of R and p, the encoder should not use MDC. Instead, single description coding has to be used. It is seen from (10) that the upper limit for redundancy is $R/2$, which is obtained for $p = 1$. That means that all the bits are allocated to shaper, which is duplicated in both descriptions.

5 Computational complexity

To perform a 3D-DCT of an $N \times N \times N$ cube, one has to perform $3N^2$ one-dimensional DCTs of size N. However, if one needs only the $N/2 \times N/2 \times N/2$ low-frequency coefficients, as in the case of shaper coding, smaller amount of DCTs need to be computed. Three stages of separable row-column-frame (RCF) transform require $[N^2 + 1/2N^2 + 1/4N^2] = 1.75N^2$ DCTs for one cube. The same is true for the inverse transform.

The encoder needs only 8 lowest coefficients of 1D-DCT. For this reason, we use pruned DCT as in [26]. The computation of 8 lowest coefficients of pruned DCT II [20] of size 16 requires 24 multiplications and 61 additions [26]. That gives 2.625 multiplications and 6.672 additions per point and brings substantial reduction in computational complexity. For comparison, full separable DCT II (decimation in frequency (DIF) algorithm) [20] of size 16 would require 6 multiplications and 15.188 additions per point.

The operation count for different 3D-DCT schemes is provided in Table 1. The adopted "pruned" algorithm is compared to fast 3-D vector-radix decimation-in-frequency DCT (3-D VR DCT) [4] and row-column-frame (RCF) approach, where 1D-DCT is computed by DIF algorithm [20]. One can see that the adopted "pruned" algorithm has the lowest computational complexity. In terms of operations per pixel, partial DCT $16 \times 16 \times 16$

is less computationally expensive than even full $8 \times 8 \times 8$ DCT used to code the residual sequence.

In [15], baseline 3D-DCT encoder is compared to the optimized H.263 encoder [32]. It was found [15] that baseline 3D-DCT encoder is up to four times faster than the optimized H.263 encoder. In the baseline 3D-DCT encoder [15], DCT was implemented by RCF approach, that gives 15.375 operations/point. In our scheme, forward pruned 3D-DCT for the shaper requires only 9.3 op/point. Adding the inverse transform, one gets 18.6 op/points. The $8 \times 8 \times 8$ DCT of the residual sequence can be implemented by 3-D VR DCT [4], which requires 13.5 op/point. Thus, the overall complexity of the transforms used in the proposed encoder is estimated as 32.1 op/point that is about twice higher than the complexity of the transforms used in baseline 3D-DCT (15.375 op/point).

The overall computational complexity of the encoder includes quantization and entropy coding of the shaper coefficients. However, the number of coefficients coded in shaper is 8 times lower than the number of coefficients in the residual sequence as only 512 lower DCT coefficients in each $16 \times 16 \times 16$ block are coded in shaper. Thus, quantization and entropy coding of the shaper would take about 8 times less computations than quantization and entropy coding of the residual sequence. Thus, we estimate that the overall complexity of the proposed encoder is not more than twice the complexity of baseline 3D-DCT [15]. This means that the proposed coder has up to two times lower computational complexity than the optimized H.263 [32]. The difference in computational complexity between the proposed coder and H.263+ with scalability (providing error resilience) is even bigger. However, the proposed coder has single description performance similar or even higher than H.263+ [12] with SNR scalability, that is shown in Sec. 7.

6 Packetization and transmission

The bitstream of the proposed video coder is packetized as follows. A group of pictures (16 frames) is split into 3D-volumes of size $16 \times 16 \times 16$. One packet should contain one or more shaper volumes which results in 512 coefficients (due to thresholding), which are entropy-coded.

In case of single description coding, one shaper volume is followed by eight spatially corresponding volumes of the residual sequence, which have the size of $8 \times 8 \times 8$. In case of multiple description coding, a packet from *Description 1* contains a shaper volume and four residual volumes taken in the pattern shown in Fig. 2(b). The *Description 2* contains the same shaper volume and four residual volumes, which are not included into *Description 1*. If the size of such a block (one shaper volume and four residual volumes) is small, several blocks are packed into one packet.

The DC coefficient of the shaper volume is predicted from the DC coefficient of a temporally preceding volume.

If both packets containing the same shaper volume are lost, DC coefficient is obtained as the previous DC coefficient in the same spatial location or as an average of DC coefficients of the spatially adjacent volumes. This concealment may introduce mismatch in DPCM loop between the encoder and decoder. However, the mismatch does not spread out of the border of this block, like in case of MC-coding. The mismatch is corrected by the DC coefficient update which can be requested over a feedback channel or may be done periodically.

To further improve robustness against burst errors, bitstream can be reordered in a way that descriptions corresponding to one 3-D volume are transmitted in the packets which are not consecutive. It will decrease the probability that both descriptions are lost due to consequent packet losses. Another solution to improve error resilience is to send the packets of *Description 1* over one link, and packets from *Description 2* over another link.

7 Simulation results

This section presents the comparison of the proposed MD coder with other MD coders. The experiments are performed on sequences "Tempete" (CIF, 30 fps, 10 s), "Silent voice" (QCIF, 15 fps, 10 s), and "Coastguard" (CIF, 30 fps). We measure the reconstruction quality by using the *peak signal-to-noise ratio* (PSNR). The distortion is average luminance PSNR over time, all color components are coded. We compare our scheme mainly with H.263-based coders as our goal is low-complexity encoder. Apparently, the proposed scheme cannot compete with H.264 in terms of compression performance. However, H.264 encoder is much more complex.

7.1 Single description performance

Fig. 4 plots PSNR versus bitrate for sequence "Tempete". The compared coders are single description coders. "3D-2stage" coder is a single-description variety of the coder described above. The shaper is sent only once, and the residual sequence is sent in a single description. "3D-DCT" is a simple 3D-DCT coder described in [8, 15]. "H.263" is a Telenor implementation of H.263. "H.263-SNR" is an H.263+ with SNR scalability, implemented at the University of British Columbia [9, 23]. One can see that H.263 coder outperforms other coders. Our 3D-2stage has approximately the same performance as H.263+ with SNR scalability and its PSNR is half to one dB lower than that of H.263+. Simple 3D-DCT coder showed the worst performance.

Fig. 4 shows PSNR of the first 100 frames of "Tempete" sequence. The sequence is encoded to target bitrate 450 Kbits/s. Fig. 4 demostrates that 3D-DCT coding exhibits temporal degradation of quality on

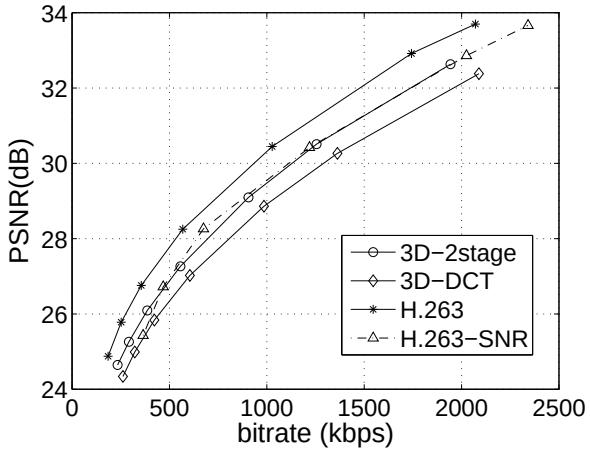


Figure 4: Sequence "Tempete", single description coding.

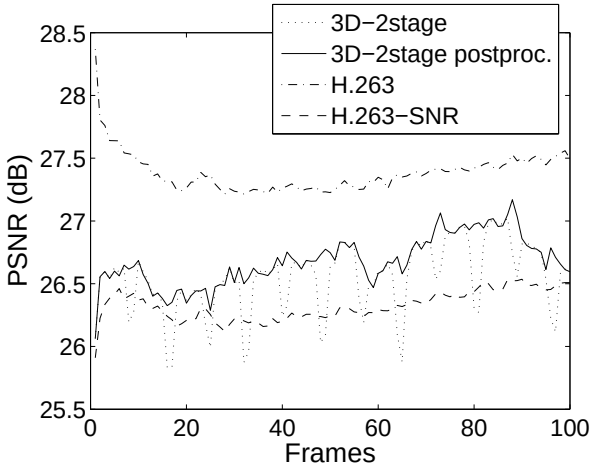


Figure 5: Sequence "Tempete" coded at 450 kbps, single description coding.

the borders of 8-frame blocks. These temporal artifacts are caused by block-wise DCT and perceived like abrupt movements. However, these artifacts can be efficiently concealed with postprocessing on the decoder side. In this experiment, we applied MPEG-4 deblocking filter [14] to block borders in temporal domain. As a result, temporal artifacts are smoothed. The perceived quality of the video sequence has also improved. Some specialized methods for deblocking in temporal domain can be applied as in [24]. Postprocessing in temporal and spatial domain can also improve reconstruction quality in case of description loss. However, in the following experiments, we do not use postprocessing in order to have fair comparison with other MDC methods.

7.2 Performance of different residual coding methods

In the following, we compare the performance of MD coders in terms of side reconstruction distortion, while they have the same central distortion. Three variants of the proposed 3D-2sMDC coder are compared. These MD coders use different schemes for coding the residual sequence. "Scheme 1" is the 2-stage coder, which uses hybrid transform for the residual sequence coding and the deblocking filtering of shaper. "Scheme 2" employs DCT for coding the residual sequence. "Scheme 3" is similar to "Scheme 2" except that it uses the deblocking filter (see Fig. 1). We have compared these schemes with simple MD coder based on 3D-DCT and MDSQ [28]. MDSQ is applied to the first N coefficients of $8 \times 8 \times 8$ 3D-DCT cubes. Then, MDSQ indices are sent to corresponding descriptions, and the rest of $512 - N$ coefficients are split between two descriptions (even coefficients go to description 1 and odd coefficients to description 2).

Fig. 6 shows the result of side reconstruction for the reference sequence "Tempete". The average central distortion (reconstruction from both descriptions) is fixed for all encoders, $D_0 = 28.3$ dB. The mean side distortion (reconstruction from one description) versus bitrate is compared. One can see that "Scheme 1" outperforms other coders, especially in the low-redundancy region. One can also see that the deblocking filtering applied to the shaper ("Scheme 3") does not give much advantage for the coder using 3D-DCT for coding the residual sequence. However, the deblocking filtering of the shaper is necessary in the "Scheme 1" as it considerably enhances visual quality. The deblocking filtering requires twice less operations comparing to the sequence of the same format in H.263+ because the block size in the shaper is twice larger than that in H.263+. All the three variants of our coder outperform the "3D-MDSQ" coder to the extent of 2 dB.

7.3 Network performance of the proposed method

Fig. 7 shows shows performance of the proposed coder in network environment with error bursts. In this experiment, bursty packet loss behavior is simulated by a two-state Markov model. These two states are

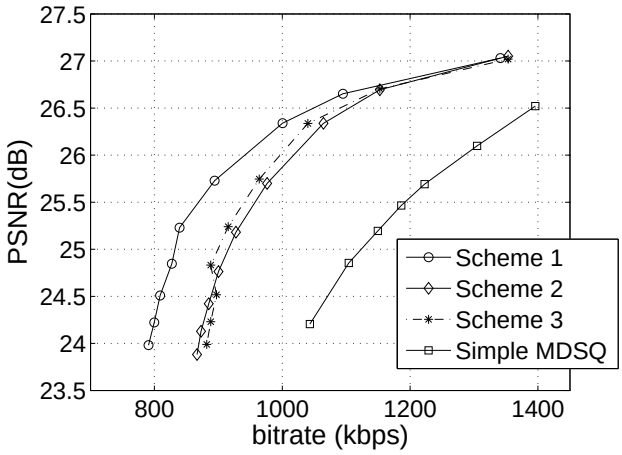


Figure 6: Sequence "Tempete", 3D-2sMDC, mean side reconstruction. $D_0 \approx 28.3$ dB.

G (good) when packets are correctly received and B (bad) when packets are either lost or delayed. This model is fully described by transition probabilities p_{BG} from state B to state G and p_{GB} from G to B. This model can also be described by average loss probability $P_{\mathrm{B}} = \Pr(\mathrm{B}) = \frac{p_{\mathrm{GB}}}{p_{\mathrm{GB}}+p_{\mathrm{BG}}}$ and the average burst length $L_{\mathrm{B}} = 1/p_{\mathrm{BG}}$.

In this experiment, sequence "Tempete" (CIF, 30 fps) has been coded to bitrate 450 kbps into packets not exceeding the size of 1000 bytes for one packet. The coded sequence is transmitted over two channels modeled by two-state Markov models with $P_{\mathrm{B}} = 0.1$ and $L_{\mathrm{B}} = 5$. Packet losses in *Channel 1* are uncorrelated with errors in *Channel 2*. Packets corresponding to *Description 1* are transmitted over *Channel 1*, and packets corresponding to *Description 2* are transmitted over *Channel 2*. Two channels are used to to unsure uncorrelated losses of Description 1 and Description 2. Similar results can be achieved by interleaving packets (descriptions) corresponding to same spatial locations. When both descriptions are lost, error concealment described in Sec. 6 is used. Optimal redundancy for "Tempete" sequence estimated by (10) for bitrate 450 kbps (0.148 bpp) is 21%.

Fig. 7 shows network performance of 3D-2sMDC and 3D-2sMDC with postrocessing (temporal deblocking). Performance of a single description 3D-2stage coder with postprocessing in a lossless environment is also given in Fig. 7 as a reference. One can see that using MDC for error resilience helps to maintain an acceptable level of quality when transmitting over network with packet losses.

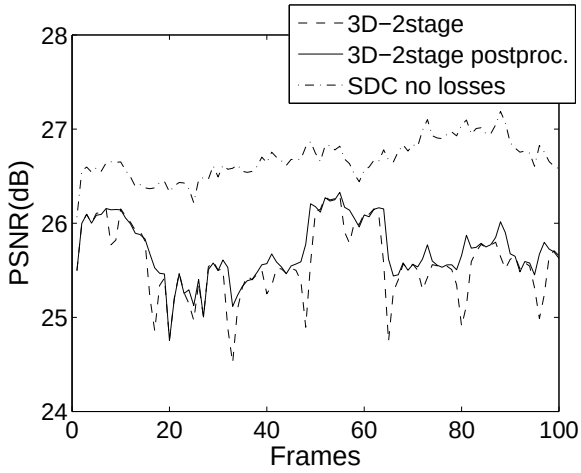


Figure 7: Network performance, packet loss rate 10%. Sequence "Tempete", coded at 450 kbps. Comparison of 3D-2sMDC and 3D-2sMDC with posfiltering. Performance of single description coder without losses is given as a reference.

7.4 Comparison with other MD coders

The next set of experiments is performed on the first 16 frames of the reference sequence Coastguard (CIF, 30 fps). The first coder is the proposed 3D-2sMDC coder Scheme 1. The "H.263 spatial" method exploits H.263+ [23] to generate layered bitstream. The base layer is included in both descriptions while the enhancement layer is split between two descriptions on a GOB basis. The "H.263 SNR" is similar to the previous method with the difference that it uses SNR scalability to create two layers.

Fig. 8 plots the single description distortion versus bitrate of the Coastguard sequence for three coders described above. The average central distortion is $D_0 = 28.5$ dB. One can see that 3D-2stage method outperforms the two other methods.

The results indicate that the proposed MD coder based on 3D-transforms outperforms simple MD coders based on H.263+ and the coder based on MDSQ and 3D-DCT. For the coder with SNR scalability, we were not able to get the bitrates as low as we have got with our "3D-2stage" method. In the experiments, we do not compare our coder with MD coders based on H.264 as such encoders have much higher computational complexity than the proposed encoder.

Another set of experiments is performed on the reference sequence "Silent voice" (QCIF, 15 fps). The proposed 3D-2sMDC coder is compared with MDTC coder that uses three prediction loops in the encoder [22], [21]. The 3D-2sMDC coder exploits "Scheme 1" as in the previous set of experiments. The rate-distortion performance of these two coders is shown in Fig. 9. The PSNR of two-description reconstruction

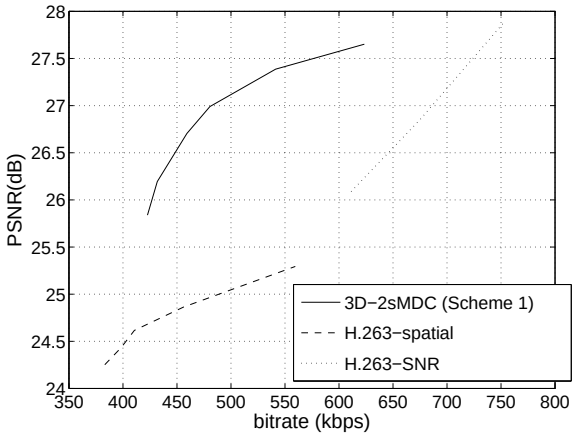


Figure 8: Sequence "Coastguard", mean side reconstruction. $D_0 \approx 28.5$ dB.

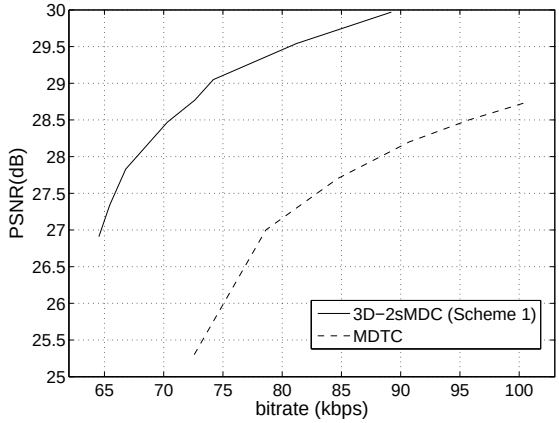


Figure 9: Sequence "Silent voice", mean side reconstruction. $D_0 \approx 31.53$ dB.

of 3D-2sMDC coder is $D_0 = 31.47 - 31.57$ dB and central distortion of MDTC coder is $D_0 = 31.49$ dB.

The results show that the proposed 3D-2sMDC coder outperforms MDTC coder, especially in a low-redundancy region. The side reconstruction performance of our coder could be explained by the following. MC-based multiple description video coder has to control the mismatch between the encoder and decoder. It could be done, for example, by explicitly coding the mismatch signal, as it is done in [21, 22]. In opposite, MD coder based on 3D-transforms does not need to code the residual signal, thus, gaining advantage of very low redundancies (see Table 2). The redundancy in Table 2 is calculated as the additional bitrate for MD coder comparing to the single description 2-stage coder based on 3D-transforms.

One of the drawbacks of our coder is high delay. High delays are common for coders exploiting 3D-transforms (e.g., coders based on 3D-DCT or 3D-wavelets). Waiting for 16 frames to apply 3D transform introduces

Central PSNR (dB)	Mean-side PSNR (dB)	Bitrate (kbps)	Redundancy (%)
31.49	26.91	64.5	9.8
31.51	27.34	65.5	11.4
31.51	27.83	66.8	13.7
31.57	28.47	70.3	19.6
31.52	29.05	74.2	26.3
31.47	29.54	81.2	38.2
31.53	29.97	89.2	51.8

Table 2: Reconstruction results. Sequence "Silent voice".

additional delay of slightly more than half a second for the frame rate 30 fps and about one second for 15 fps. The proposed coder also requires larger memory than MC-based video coder, as it requires to keep the 16 frames in the buffer before applying the DCT. This property is common for most of 3D transform video coders. However, we suppose that most of modern mobile devices have enough memory to perform the encoding.

Fig. 10 shows frame 13 of the reference sequence Tempete reconstructed from both descriptions (Fig. 10(a)) and from *Description 1* alone (Fig. 10(b)). The sequence is coded by 3D-2sMDC (Scheme 1) encoder to bitrate $R = 0.292$ bpp. One can see that although the image reconstructed from one description has some distortions caused by loss of transform coefficient volumes of the residual sequence, the overall picture is smooth and pleasant to eyes.

8 Conclusion

We have proposed an MDC scheme for coding of video which does not use motion-compensated prediction. The coder exploits 3D-transforms to remove correlation in video sequence. The coding process is done in two stages: the first stage produces coarse sequence approximation (shaper) trying to fit as much information as possible in the limited bit budget. The second stage encodes the residual sequence, which is the difference between the original sequence and the shaper-reconstructed one. The shaper is obtained by pruned 3D-DCT, and the residual signal is coded by 3D-DCT or hybrid 3D-transform. The redundancy is introduced by including the shaper in both descriptions. The amount of redundancy is easily controlled by the shaper quantization step. The scheme can also be easily optimized for suboptimal bit-allocation. This optimization can run in real time during the encoding process.

The proposed MD video coder has low computational complexity, which makes it suitable for mobile devices

with low computational power and limited battery life. The coder has been shown to outperform MDTC video coder and some simple MD coders based on H.263+. The coder performs especially well in a low-redundancy region. The encoder is also less computationally expensive than the H.263 encoder.

9 Acknowledgments

This work was supported by the Academy of Finland, project No. 213462 (Finnish Centre of Excellence program (2006 - 2011)).

References

[1] J. Apostolopoulos, "Error-resilient video compression through the use of multiple states," in *Proc. Int. Conf. Image Processing*, vol. 3, Sept. 2000, pp. 352–355.

[2] J. Apostolopoulos and S. Wee, "Unbalanced multiple description video communication using path diversity," in *Proc. Int. Conf. Image Processing*, vol. 1, Oct. 2001, pp. 966–969.

[3] M. Bakr and A. Salama, "Implementation of 3D-DCT based video encoder/decoder system," in *Proc. IEEE MWSCAS-2002*, vol. 2, Aug. 2002, pp. II–13–16.

[4] S. Boussakta and H. Alshibami, "Fast algorithm for the 3-D DCT-II," *IEEE Trans. Signal Processing*, vol. 52, pp. 992–1001, Apr. 2004.

[5] N. Bozinovic and J. Konrad, "Motion analysis in 3D DCT domain and its application to video coding," *IEEE Circuits Syst. Video Technol.*, vol. 20, pp. 510–528, 2005.

[6] A. Bruchstein, M. Elad, and R. Kimmel, "Down-scaling for better transform compression," *IEEE Trans. Image Processing*, vol. 12, no. 9, pp. 1132–1144, Sept. 2003.

[7] A. Burg, R. Keller, J. Wassner, N. Felber, and W. Fichtner, "A 3D-DCT real-time video compression system for low complexity single-chip VLSI implementation," in *Proc. of the MoMuC2000*, Tokyo, 2000, pp. 1B–5–1.

[8] R. Chan and M. Lee, "3D-DCT quantization as a compression technique for video sequences," in *Proc. IEEE Conf. Virtual Systems and Multimedia (VSMM'97)*, Sept. 1997, pp. 188–196.

[9] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: video coding at low bitrates," *IEEE Circuits Syst. Video Technol.*, vol. 8, pp. 849–866, Nov. 1998.

[10] W. Equitz and T. Cover, "Successive refinement of information," *IEEE Trans. Inform. Theory*, vol. 37, no. 2, pp. 269–275, Mar 1991.

[11] V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Mag.*, vol. 18, pp. 74–93, September 2001.

[12] ITU-T, *Video coding for low bitrate communication.* ITU-T Recommendation, Draft on H.263v2, 1999.

[13] J. Kim, R. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the internet using unequal error protection," *IEEE Trans. Image Processing*, vol. 12, pp. 121–131, 2003.

[14] S. D. Kim, J. Yi, H. M. Kim, and J. B. Ra, "A deblocking filter with two separate modes in block-based video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, pp. 156–160, Feb. 1999.

[15] J. Koivusaari and J. Takala, "Simplified three-dimensional discrete cosine transform based video codec," in SPIE-IS&T Electronic Imaging, Multimedia on Mobile Devices, vol. 5684, San Jose, CA, Jan. 2005, pp. 11–20.

[16] H. S. Malvar and D. H. Staelin, "The LOT: transform coding without blocking effects," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 37, pp. 553–559, Apr. 1989.

[17] H. Man, R. Queiroz, and M. Smith, "Three-dimensional subband coding techniques for wireless video communications," *IEEE Circuits Syst. Video Technol.*, vol. 12, pp. 386–397, Jun. 2002.

[18] A. Norkin, A. Gotchev, K. Egiazarian, and J. Astola, "A low-complexity multiple description video coder based on 3D-transforms," in *Proc. European Signal Processing Conference (EUSIPCO'06)*, Florence, Italy, Sept. 2006.

[19] ——, "Two-stage multiple description image coders: Analysis and comparative study," *EURASIP Journal Signal Processing: Image Communication*, vol. 21/8, pp. 609–625, Sept. 2006.

[20] K. Rao and R. Yip, *Discrete cosine transform: algorithms, advantages, applications.* 12–28 Oval Road, London: Academic Press Limited, 1990.

[21] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, and R. Puri, "Multiple description coding for video using motion-compensated prediction," in *Proc. IEEE Int. Conf. Image Processing (ICIP99)*, vol. 3, Oct. 1999, pp. 837–841.

[22] ——, "Multiple description coding for video using motion-compensated temporal prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 193–204, Mar. 2002.

[23] L. Roberts, "TMN 8 (h.263+) encoder/decoder, version 3.0," Signal Processing and Multimedia Lab., Univ. British Columbia, May 1997.

[24] D. Rusanovskyy and K. Egiazarian, "Postprocessing for three-diminsional discrete cosine transform based video coding," in *Proc. 7th Int. Conf. Advanced Concepts for Intelligent Vision Systems, ACIVS*, Antwerp., Belgium, Sept. 2005, pp. 618–625.

[25] S. Saponara, L. Fanucci, and P. Terreni, "Low-power VLSI architectures for 3D discrete cosine transform (DCT)," in *Proc. IEEE MWSCAS-2003*, vol. 3, Dec. 2003, pp. 1567–1570.

[26] A. Skodras, "Fast discrete cosine transform pruning," *IEEE Trans. Signal Processing*, vol. 42, pp. 1833–1837, July. 1994.

[27] S. Somasundaram and K. Subbalakshmi, "3-d multiple description video coding for packet switched networks," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME'03)*, vol. 1, July 2003, pp. I – 589–592.

[28] V. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 821–834, May 1993.

[29] V. Vaishampayan and S. John, "Balanced interframe multiple description video compression," in *Proc. IEEE Int. Conf. Image Processing (ICIP99)*, vol. 3, Oct. 1999, pp. 812 – 816.

[30] Y. Wang, A. Reibman, and S. Lin, "Multiple description coding for video delivery," *Proceedings of the IEEE*, vol. 93, pp. 57–70, Jan. 2005.

[31] B.-L. Yeo and B. Liu, "Volume rendering of DCT-based compressed 3D scalar data," *IEEE Trans. Visualization and Computer Graphics*, vol. 1, pp. 29–49, Mar. 1995.

[32] K. Yu, J. Lv, J. Li, and S. Li, "Practical real-time video codec for mobile devices," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME 2003)*, vol. 3, July 2003, pp. 509–512.

[33] M. Yu, Z. Wenqin, G. Jiang, and Z. Yin, "An approach to 3D scalable multiple description video coding with content delivery networks," in *Proc. IEEE Int. Workshop VLSI Design and Video Technology*, May 2005, pp. 191–194.

(a) Reconstruction from both descriptions, $D_0 = 28.52$.

(b) Reconstruction from *Description 1*, $D_1 = 24.73$.

Figure 10: Sequence Tempete, frame 13.