14th European Signal Processing Conference (EUSIPCO 2006), Florence, Italy, September 4-8, 2006, copyright by EURASIP

A LOW-COMPLEXITY MULTIPLE DESCRIPTION VIDEO CODER BASED ON 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

ABSTRACT

The paper presents a multiple description (MD) video coder based on three-dimensional (3D) transforms. The coder has low computational complexity and high robustness to transmission errors and is targeted to mobile devices. The encoder represents video sequence in a form of coarse sequence approximation (shaper) included in both descriptions and residual sequence (details) 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 simple and yet outperforms some MD coders based on motion-compensated prediction, especially in the low-redundancy region.

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 the high compression, which is usually achieved with the cost of increasing encoding complexity. However, portable devices, such as camera phones, still suffer from lack of computational power and energy-consumption constraints. 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 (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 can be expected compared to traditional video coders [1]. 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. Previous research on this topic shows that simple (baseline) 3D-DCT video encoder is three to four times faster than an optimized H.263 encoder [2], for the price of an acceptable compression efficiency loss [3].

A 3D-DCT video coder is also advantageous in error resilience. 3D-DCT video coders enjoy no error propagation in the subsequent frames, a problem always present in MC-based coders. 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) [4]. Multiple encoded bitstreams are generated from the source information. The resulting descriptions 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 [5]. In this case, each bitstream (description) is sent to the receiver over 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 have 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. The solution to prevent this problem is to use three separate prediction loops at the encoder [6] to control the mismatch. Another solution is to use a separate prediction loop for every description [7]. However, both approaches decrease the compression efficiency. A good review of MDC approaches to video coding is given in [8].

In this paper, we propose an MD video coder (3D-2sMDC), which does not exploit motion compensation. 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 [9] to coding of video sequences. 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 [3], 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.

2. GENERAL CODER SCHEME

In our scheme, video sequence is coded in two stages as shown in Fig. 1. In the first stage (dashed rectangle), the 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 $16 \times 16 \times 16$. 3D-DCT is applied to each cube. Then, only the lower DCT coefficients in the $8 \times 8 \times 8$ cube are quantized and entropy-coded (see Fig. 2) composing the shaper, other coefficients are set to zero. Decoding of the shaper is done in the inverse order.

In our video coder, the residual sequence is coded by a 3-dimensional block transform. It can be either 3D-DCT or hybrid 3D-transform. Transform coefficients are finely quantized with a uniform quantization step (Q_d). Then, transform blocks are split into two parts in a manner depicted in Fig. 3 and entropy-coded. One part of blocks together with the shaper forms *Description 1*,

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

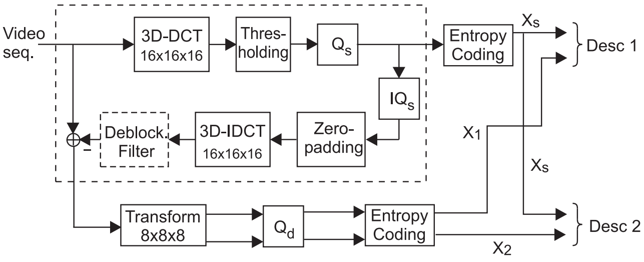


Figure 1: Encoder scheme.

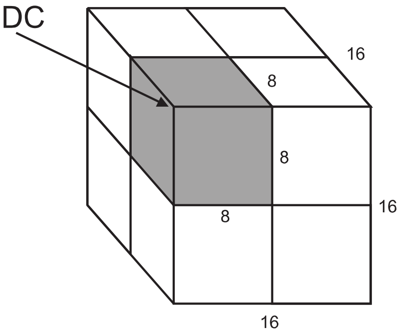


Figure 2: 3D-DCT cube for shaper coding: only coefficients in the gray block are coded, other coefficients are set to zero.

while the second part combined again with the shaper forms *Description 2*. Thus, each description consists of a coarse sequence approximation and *half* of the transform blocks of the residual sequence.

The shaper is presented in both descriptions to facilitate successful reconstruction when one description is lost. Thus, the redundancy in the proposed coder is only determined by the shaper quality 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 step results in higher quality side reconstruction. The quality of a two-channel reconstruction is controlled by the quantization step Q_d used in the coding of the residual sequence.

The operation of the decoder is the following. When the decoder receives two descriptions, it decodes the shaper and interpolates it to the original sides. Then the decoder adds the residual signal to shaper, thus, reconstructing the sequence with high quality. When one description is lost during the transmission, the decoder receives only half of the residual cubes. The coefficients in the lost cubes are set to zero and the inverse transform is applied. As the residual sequence have only half of the coefficient blocks, the reconstruction has lower, however, still acceptable quality.

The obtained coder provides balanced descriptions both in terms of PSNR and bitrate. The following three sections explain each stage of the coding process in detail.

3. COARSE SEQUENCE APPROXIMATION

The idea of the first 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 low bitrate [10]. Then, the original resolution sequence can be reconstructed by interpolation as a post-processing step. A good interpolation and decimation method would concentrates more information in the coarse approximation and correspondingly makes 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 is set to two. Combining downscaling with the transform, we split the original sequence into blocks of size $16 \times 16 \times 16$. 3D-DCT is applied to each block. Then, $8 \times 8 \times 8$ block of the lowest DCT coefficients is quantized and coded. We use scanning of the coefficients in the $8 \times 8 \times 8$ block described in [11], which is similar to zigzag scan. DC coefficients of transformed shaper cubes are coded by DPCM prediction followed by entropy coding. The DC coefficient of a cube is predicted from the DC coefficient of the temporally preceding cube.

The decoding of the shaper is done similarly. An $8 \times 8 \times 8$ block of coefficients is decoded from the bitsream. Then, this block is padded by zeros to the size of $16 \times 16 \times 16$, and inverse DCT is applied.

4. COMPUTATIONAL COMPLEXITY

There is no need to compute full forward 3D-DCT transform of size $16 \times 16 \times 16$ as only 1/8 of coefficients are used (Fig. 2). 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, 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 [12]. The computation of 8 lowest coefficients of pruned DCT II [13] of size 16 requires 24 multiplications and 61 additions [12]. That gives 2.625 multiplications and 6.672 additions per point. For comparison, full separable DCT II (decimation in frequency (DIF) algorithm) [13] of size 16 would require 6 multiplications and 15.188 additions per point, which brings substantial reduction in computational complexity.

The operation count for different 3D-DCT schemes is provided in Table 1. The adopted "pruned" algorithm is compared to fast 3-D VR DCT [14] and row-column-frame (RCF) approach, where 1D-DCT is computed by DIF algorithm [13]. 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.

The results from [3] show that baseline 3D-DCT encoder is three to four times faster than the optimized H.263 encoder. The 3D-DCT coder, used in comparison was implemented by RCF approach, that gives 15.357 operations/point. The forward pruned 3D-DCT for the shaper requires only 9.25 op/point. The overall computational complexity of the shaper coding includes also quantization and entropy coding of the shaper coefficients, and the inverse transform. The number of coefficients coded in the shaper is 8 times lower than the number of coefficients in the residual sequence. Thus, we estimate that the overall complexity of the proposed encoder is not more than twice of that of baseline 3D-DCT [3]. This means that the proposed coder has at least 1.5 to 2 times lower computational complexity than H.263. 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 the single description performance similar to H.263+ [2] with SNR scalability, which will be discussed in Section 7.

5. RESIDUAL SEQUENCE CODING

The residual sequence is obtained by subtracting the reconstructed shaper from the original sequence. 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 base layer. Each group of 8 frames is coded by block 3D-transform.

Two different transforms can be used to code the residual sequence. The first transform is 3D-DCT and the second is hybrid transform. Hybrid transform consists of a lapped orthogonal trans-

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.

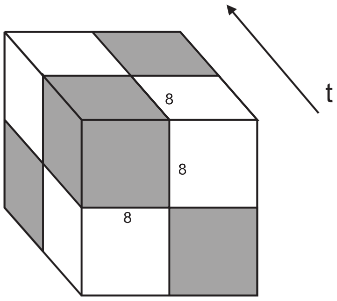


Figure 3: Split pattern for blocks of a residual sequence: ”gray” - description 1; ”white” - description 2.

form (LOT) in vertical and horizontal directions, and DCT in temporal direction. Both DCT and hybrid transform produce $8 \times 8 \times 8$ cubes of coefficients. The cubes of coefficients are split between two descriptions in a pattern shown in Fig. 3.

As it is demonstrated in Section 7, hybrid transform outperforms DCT in terms of PSNR and visual quality. Moreover, using LOT in spatial dimensions gives more pleasant picture compared to DCT-coded one. However, the blocking artifacts introduced by coarse coding of the shaper are not completely concealed by the residual sequence coded with hybrid transform. Moreover, these artifacts impede efficient compression of the residual sequence. Therefore, the deblocking filter is applied to the reconstructed shaper (see Fig. 1). In the experiments, we use the deblocking filter from H.263+ standard [2].

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-blocks. One packet includes at least one shaper block, which has 512 coefficients. It corresponds to $16 \times 16 \times 16$ cube. In case of a single description coding, one shaper block is followed by eight spatially corresponding blocks of the residual sequence, which have the size of $8 \times 8 \times 8$. In case of multiple description coding, *Description 1* consists of one shaper block and four residual blocks taken in the pattern shown in Fig. 3; *Description 2* consists of the same shaper block and four residual blocks, which are not included into *Description 1*.

The DC coefficient of a shaper cube is predicted from the DC coefficient of a temporally preceding block, and the prediction error is entropy coded. The DC coefficient of a residual block is not predicted. If two descriptions containing the same shaper cube are lost, the DC coefficient is set as the average of DC coefficients belonging to the temporally and spatially adjacent cubes. This 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 feedback channel or can be done periodically.

The bitstream can be reordered in a way that the descriptions corresponding to one block are transmitted in the packets that are not consecutive or transmitted over different paths. It will decrease the probability that both descriptions are lost due to bursts of errors.

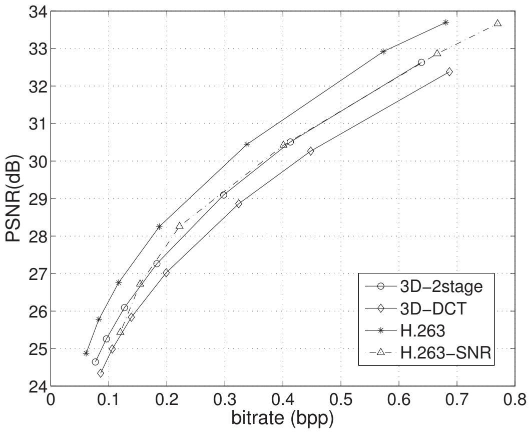


Figure 4: Sequence “Tempete”, single description coding.

7. EXPERIMENTAL 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) and “Silent voice” (QCIF, 15 fps, 10 s). We measured 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.

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 [1, 3]. “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 [15, 16]. 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.

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 [17]. 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. 5 shows the results 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

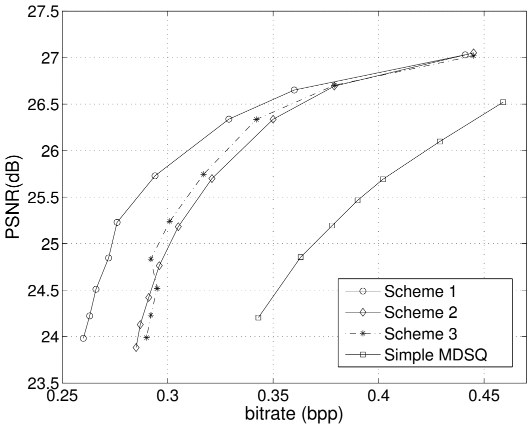


Figure 5: Sequence "Tempete", 3D-2sMDC, mean side reconstruction. $D_0 = 28.3$ dB.

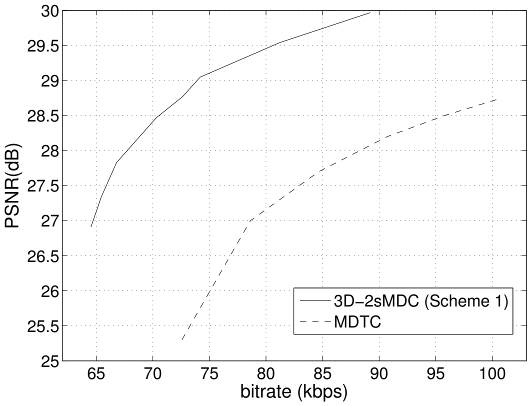


Figure 6: Sequence "Silent voice", mean side reconstruction. $D_0 \approx 31.53$ dB.

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.

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 [18, 6]. We do not compare the proposed coder with MD video coders based on computationally more complex H.264 as we target low-complexity encoding. 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. 6. The PSNR of two-description reconstruction 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 [6, 18]. In opposite, MD coder based on 3D-transforms does not need to code the residual signal. This allows such a coder to operate in a very low redundancies (see Table 2) and outperform some MC-based MD video coders for the side reconstruction. 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.

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".

One of the drawbacks of our coder is high delay. High delays are common for the coders exploiting 3D-transforms (e.g., coders based on 3D-DCT or 3D-wavelets). Waiting for 16 frames to apply 3D transform introduces additional delay of slightly more than half a second for the frame rate 30 fps and more than one second for 15 fps.

Fig. 7 shows frame 13 of the reference sequence Tempete reconstructed from both descriptions (Fig. 7(a)) and from *Description 1* alone (Fig. 7(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 blocks of the residual sequence, the overall picture is smooth and pleasant to eyes.

8. CONCLUSIONS

We have proposed an MD video coder 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.

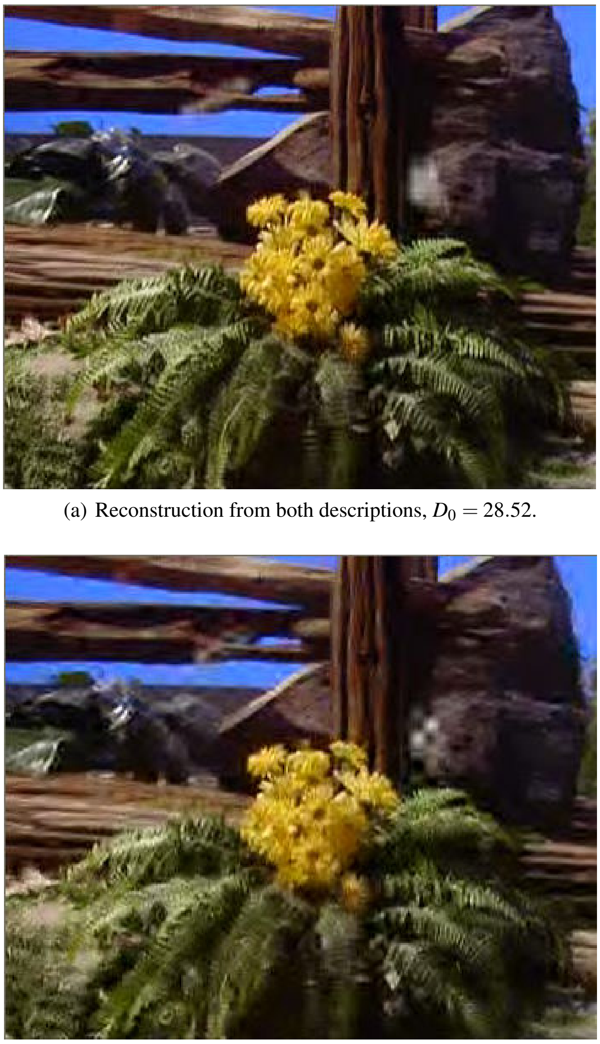
The proposed MD video coder has low computational complexity, which makes it suitable for mobile devices with low computational power and limited battery life. The coder has been shown to outperform MDTC video coder. The coder performs especially well in a low-redundancy region. The encoder is also less computationally expensive than H.263 encoder.

REFERENCES

[1] 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.

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

[3] J. Koivusaari and J. Takala, "Simplified three-dimensional discrete cosine transform based video codec," in *SPIE-IS&T Elec-*



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

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

Figure 7: Sequence Tempete, frame 13.

tronic Imaging, Multimedia on Mobile Devices, SPIE vol. 5684, San Jose, CA, Jan. 17-18, 2005, pp. 11-20.

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

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

[6] 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.

[7] 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.

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

[9] A. Norkin, A. Gotchev, K. Egiazarian, and J. Astola, “Two-stage multiple description image coders: Analysis and comparative study,” to appear in *Signal Processing: Image Communication*.

[10] A. Bruchstein, M. Elad, and R. Kimmel, “Down-scaling for better transform compression,” *IEEE Trans. Image Processing* vol. 12, pp. 1132–1144, Sept. 2003.

[11] 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.

[12] A. Scodras, “Fast discrete cosine transform pruning,” *IEEE Trans. Signal Processing*, vol. 42, pp. 1833–1837, July 1994.

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

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

[15] 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.

[16] Signal Processing and Multimedia Lab., Univ. British Columbia, “TMN 8 (H.263+) encoder/decoder, Version 3.0,” TMN 8 (H.263+) codec, May 1997.

[17] V. Vaishampayan, “Design of multiple description scalar quantizers,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 821–834, Mar. 1993.

[18] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, and R. Puri, “Multiple-description video coding using motion-compensated temporal prediction,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 193–204, Mar. 2002.