Low-complexity video compression based on 3-D DWT and fast entropy coding

Evgeny Belyaev

Tampere University of Technology
Department of Signal Processing, Computational Imaging Group

April 28, 2011

Problem definition

Scalable video coding (SVC) is the most preferable compression method for video transmission over wireless networks with time-varying bandwidth and packet losses.

Scalable video stream includes several layers:

- *base layer* is always coded independently and provide basic visual quality with low bit-rate.
- *enhancement layers* provide refinement quality and higher bit rate.

Scalable video stream provides:

- robust transmission due to unequal error protection of different layers[1];
- bandwidth adaptation due to dropping the higher enhancement layers[2].

[1] M.Gallant and F. Kossentini, "Rate-distortion optimized layered coding with unequal error protection for robust Internet video", 2001.

[2] Y.P. Fallah, H. Mansour, S. Khan, P. Nasiopoulos, H.M. Alnuweiri, "A Link Adaptation Scheme for Efficient Transmission of H.264 Scalable Video Over Multirate WLANs", 2008.

Problem definition

Currently the most popular scalable video coding approach is an extension of the H.264/SVC standard[1] which provides high compression efficiency due to motion compensation and inter-layer prediction. But, because of high computational complexity of these methods, implementation of H.264/SVC encoder in a mobile device is a difficult task.

As an alternative the scalable video coding based on three-dimensional discrete wavelet transform (3-D DWT)[2] have been proposed. This approach

- do not use motion estimation,
- but has highly complex entropy encoder.

3-D DWT codec complexity can be further decreased by development of the new low-complexity entropy coding methods.

[1] H. Schwarz, D.Marpe, and T.Wiegand, "Overview of the Scalable Video Coding Extension of the H.264 / AVC Standard", 2007.

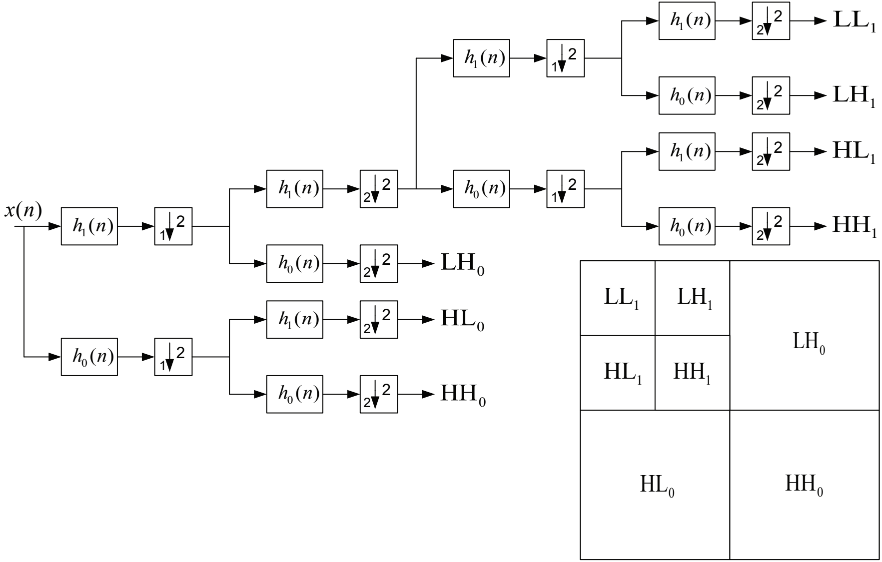
[2] B.J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)", 2000.

Main encoding stages

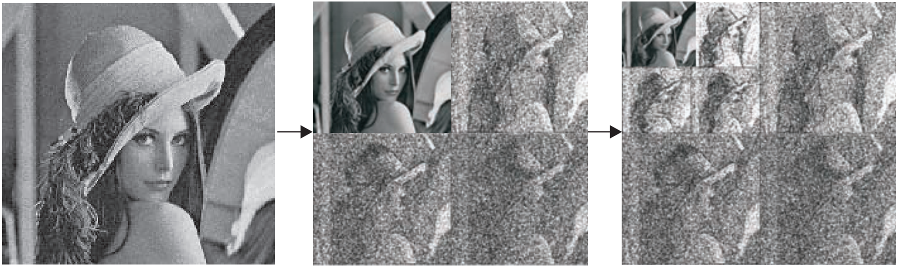
3-D DWT based video compression algorithm includes:

- Spatial two-dimensional discrete wavelet transform;
- One-dimensional discrete wavelet transform in temporal direction;
- Quantization of wavelet coefficients;
- Bit-plane entropy encoding of each wavelet matrix.

Two-dimensional discrete wavelet transform



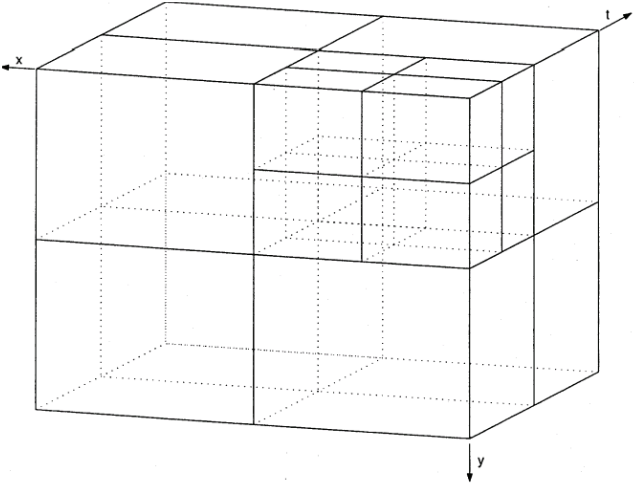
L-levels two-dimensional discrete wavelet transform



For each frame we have $(3 \times L + 1)$ wavelet matrix.

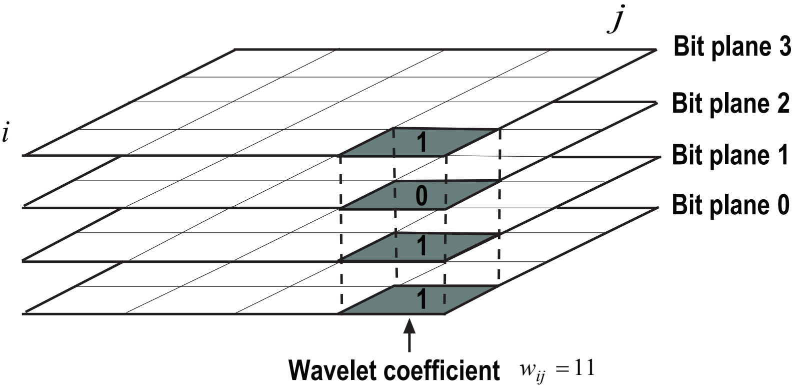
Discrete wavelet transform in temporal direction

We accumulate n frames (group of frames, GOP) and calculate wavelet transform in temporal direction.



Thus, for each GOP we have $(3 \times L + 1) \times n$ wavelet matrix.

Bit-plane entropy encoding of a wavelet matrix



- Quantity of non-zero bit-planes Q is compressed.
- Bit-planes are compressed starting from bit-plane $Q - 1$.
- All bits in each bit-plane are scanned from left to right and from top to bottom.

Bit-planes compression

- Each wavelet coefficient X has 8 neighbors which can be significant or insignificant. At the beginning all coefficients are insignificant.

D_0	V_0	D_1
H_0	X	H_1
D_2	V_1	D_3

1. If current coefficient and all neighbors are insignificant:
   - if current bit is 0, then zero-run counter N is increased by 1,
   - if current bit is 1, then zero-run counter N is compressed by Levenstein code and set to $N = 0$.
2. If at least one neighbor coefficient is significant and current coefficient is insignificant, then current bit is placed into output bit stream without compression.
3. If at steps 1 or 2 current bit was 1, then current wavelet coefficient X is placed into output bit stream without compression and become significant.

Zero-run compression by Levenstein codes

For compression of zero-run $\underbrace{000..0}_{N}1$, value N is represented by two parts:

1. Part 1:

$$\underbrace{\mathtt{binary\ representation\ of\ } c}_{\lfloor \log_2(\log_2 N_{max}) \rfloor + 1\ \mathtt{bits}},$$

   where

$$c = \begin{cases} 0, \text{ if } N = 0 \\ \lfloor \log_2 N \rfloor + 1, \text{ if } N > 0, \end{cases}$$

   N_{max} is maximum possible zero-run value.

2. Part 2: (if $c \geq 2$)

$$\underbrace{\mathtt{binary\ representation\ of\ } N - 2^{c-1}}_{c - 1\ \mathtt{bits}}.$$

Practical results

Codec	Description
x.264[1]	The fastest software implementation[2] of H.264/AVC encoder (in *-ultrafast* mode)
3-D SPIHT[3]	3-D DWT codec implementation with entropy encoder with 3-D set partitioning in hierarchical trees[4] proposed by W. A.Pearlman
3-D DWT Arithmetic	3-D DWT codec implementation with bit-plane entropy encoder based on binary adaptive range coder and context modeling in JPEG2000 style.
3-D DWT Levenstein	3-D DWT codec implementation with bit-plane entropy encoder based on Levenstein codes

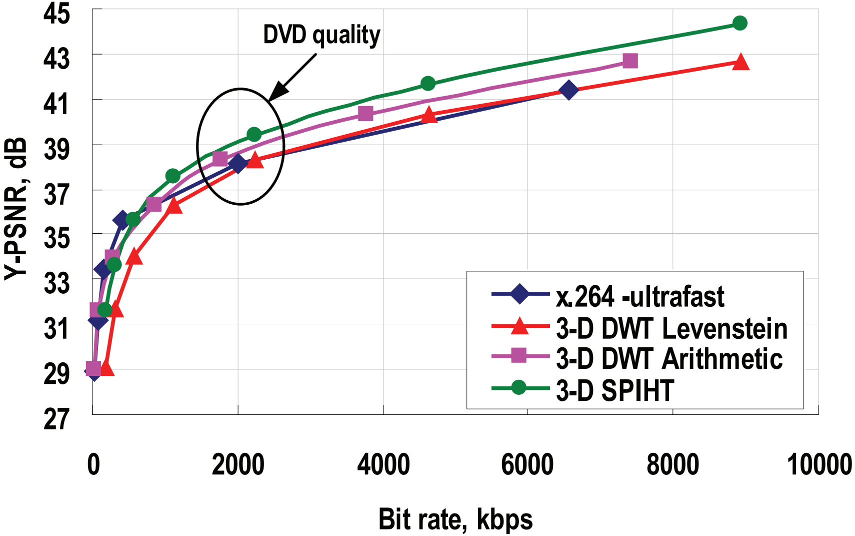
[1] http://www.videolan.org/developers/x264.html

[2] http://compression.ru/video/codec_comparison/

[3] http://ipl.rpi.edu/research/SPIHT/spiht3.html

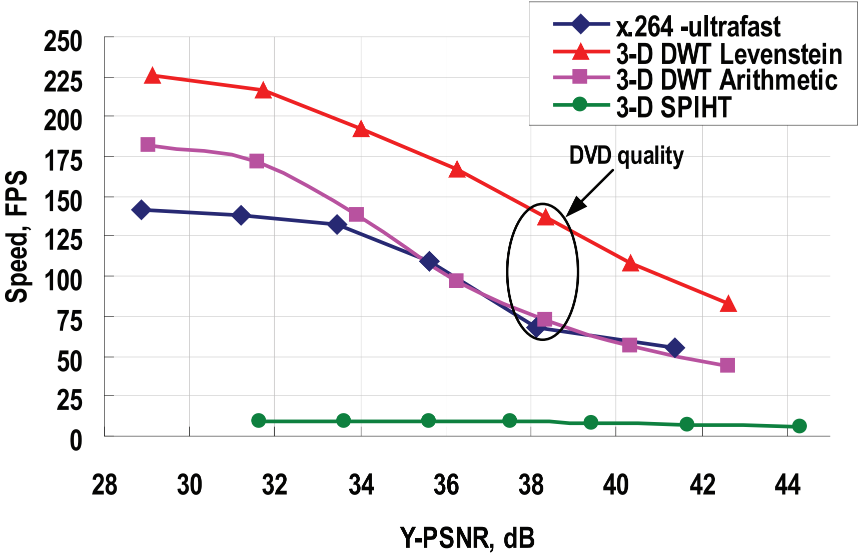
[4] B.J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)", 2000.

Rate-Distortion comparison[1]



[1]Test video sequence: vassar_0, 640×480, 30 fps.

Computation complexity comparison[1]



[1] All results were obtained for C-code software implementations without assemblers, GPU optimization etc. for processor Intel Core 2 Duo 3.6 GHz