ANISOTROPIC 3-D WAVELET PACKET BASES FOR VIDEO CODING

Rade Kutil

Salzburg University, Dept. of Scientific Computing
Jakob Haringer-Str. 2, A-5020 Salzburg, Austria
rkutil@cosy.sbg.ac.at

ABSTRACT

Video coding algorithms based on 3-D wavelet transforms exist which have a competitive rate-distortion performance and are very fast because motion compensation is not needed. The compression efficiency of these algorithms can be increased by the use of more general wavelet packet transforms. Anisotropic wavelet packets are an even more generalized type of wavelet transform, where multidimensional wavelets do not necessarily have equal dilations in each dimension. This property is advantageous especially for 3-D video coding because video data has different characteristics in the time and spatial dimensions. This paper developes methods to adapt the SMAWZ codec to anisotropic 3-D wavelet packet bases and investigates algorithm complexity and the resulting rate-distortion performance. It can be shown that there exists a fixed anisotropic wavelet packet basis that is able to outperform MPEG-4 by up to 4dB especially for video sequences with low motion.

1. INTRODUCTION

Most video compression algorithms rely on 2-D based schemes employing motion compensation techniques. On the other hand, rate-distortion efficient 3-D wavelet based algorithms exist which are able to capture temporal redundancies in a more natural way (see e.g. [1, 2]). Wavelet packet based compression methods have been developed which outperform the most advanced wavelet coders (e.g. JPEG2000, SPIHT) significantly for textured images in terms of rate-distortion performance. A similar development in the area of video compression may be observed for 3-D wavelet packet [3] video coding.

Reasons for the use of wavelet packets in video coding, apart from those in image coding, are that cyclic events such as rotating wheels produce regular patterns in the time domain which can efficiently be represented by wavelet packets, and the fact that the behaviour of video data in the spatial domain differs from that in the time domain. The latter brings up anisotropic wavelet packets which are able to represent shapes and features with different properties in different directions (in terms of frequency components).

This paper deals with the application of the SMAWZ codec developed in [4] on the 3-D wavelet transforms of a video sequence. The pyramidal wavelet transform will be used as well as conventional and anisotropic wavelet packets. The results are compared against MPEG-4.

2. ANISOTROPIC WAVELET PACKETS

The fast wavelet transform divides a data set into a low- and a high-frequency sub-band by the application of a pair of quadrature mirror filters. In the 2-D case consecutive filtering of rows and columns produces four sub-bands (eight in the 3-D case). Whereas in the pyramidal decomposition only the low-pass sub-band is decomposed the wavelet packet decomposition allows all sub-bands to be decomposed. If sub-bands may be filtered in only one dimension, one gets wavelet packets whose scales are different in different directions – anisotropic wavelet packets. In this case, the number of sub-bands and the number of bases increases substantially. Additionally, the method of representing specific bases by a decomposition tree (including all intermediate sub-bands) is not applicable any more. To avoid ambiguous representations of equal bases and redundant calculations, a graph structure called *bush* was introduced in [5] as a generalisation of trees.

To determine the optimal basis for compression, a cost function is minimised globally by the best-basis algorithm that was adopted to anisotropic wavelet packets in [6]. The L_1-norm turned out to be a good choice for the cost function and is, therefore, used in section 4.

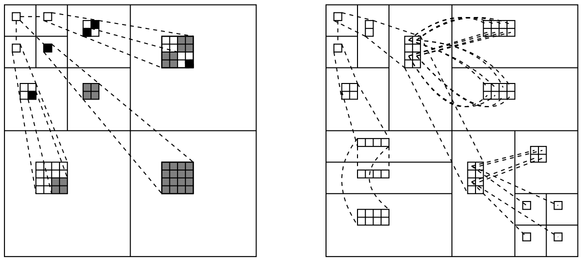
3. THE SMAWZ ALGORITHM

3.1. Zero-Trees and Significance Maps

Zero-tree based algorithms arrange the coefficients of a wavelet transform in a tree-like manner (see Fig. 1 (a)). All coefficients are connected by trees that are rooted in the approx-

The author was supported by the Austrian Science Fund FWF, project no. P13903FWF.

0-7803-7750-8/03/$17.00 ©2003 IEEE.



(a) pyramidal decomposition

(b) anisotropic WP

Fig. 1. Coefficient trees in zero-tree coding (2-D case). The dark areas are insignificant coefficients. Black coefficients are zero-tree roots

imation sub-band's coefficients (the sub-band that is low-pass filtered only). Furthermore, a zero-tree is a sub-tree which entirely consists of insignificant coefficients. The significance of a coefficient is relative to a threshold which is used in several coding passes to successively refine the image quality. Zero-trees can be encoded by a single symbol or bit.

To build trees of coefficients is more complicated when we face arbitrary wavelet packet decompositions (see Fig. 1 (b)). Coefficient trees should connect coefficients of approximately equal spatial positions in a way so that bigger coefficients (usually the low-pass filtered ones) are next to the root of the trees and the significance (absolute value) of parent and child coefficients is statistically related (similarity condition). To achieve this, we connect similar sub-bands by similarity trees. While the similarity tree is fixed and simple for the pyramidal basis, rules have to be developed to construct such trees for wavelet packet bases and the even more complicated case of anisotropic wavelet packets. See [6] for details.

Once the similarity trees are constructed, the SMAWZ algorithm [4] can be applied without major modifications.

3.2. Arithmetic Coding

SPIHT uses arithmetic coding to improve rate-distortion performance by about 0.5 dB. Four bits of four neighbouring coefficients (the direct offspring of a single parent coefficient) are grouped together and fed into the arithmetic coder as a single symbol. In the case of wavelet packet decompositions (and similarity trees), the direct offspring of a coefficient does not necessarily consist of exactly four coefficients. This makes the above approach not applicable. Looking for another context-based arithmetic coding approach for coefficient significance, we find EBCOT [7] which JPEG2000 is based upon. However, in the 3-D case arithmetic coding is more complicated since a straight-forward extension of the EBCOT contexts would produce too many contexts. This would undermine the adaptiveness of the arithmetic coding algorithm. Note that 26 neighbours together with 8 sub-band types would produce about $5 \cdot 10^8$ contexts.

As in EBCOT, the significance bits of neighbouring coefficients are summed (see Fig. 2). There are two first order neighbours in each of the three dimensions. The corresponding significance sums h, v and t can take values from 0 to 2. The second order neighbours are summed in hv, vt and th which take values from 0 to 4. The third order neighbours are summed in d which takes values from 0 to 8.

To reduce the number of contexts, only a few cases for first and second order neighbours are considered. There are eight cases for each type which are represented by a and b (see Fig. 2). a and b, together with d and the sub-band type are reduced further to 26 different contexts. This is the result of lengthy experiments with significance statistics.

Adopting the context model for signs to the 3-D case leads to the same problem of too many contexts. As in EBCOT, h, v and t are set to the sign of neighbouring coefficients in horizontal, vertical and temporal directions. h, v and t are set to $+1$ (-1) if at least one neighbour has positive (negative) sign. They are set to zero if either both neighbours are insignificant (not encoded yet) or the neighbours have differing signs.

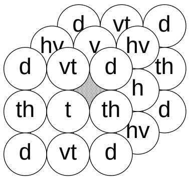
So the number of contexts is $3^3 \cdot 8 = 216$. The contexts (h, v, t) and $(-h, -v, -t)$ have the same probabilities for $+1$ and -1 respectively, so the number of contexts can be halved. Nevertheless, the result is still too much.

Experiments have shown that the context (h, v, t) for a HLx sub-band can be used as (v, h, t) context for a LHx sub-band as well because of approximately equal probabilities. Additionally, (h, v, t) and (v, h, t) can be identified for a LLx or a HHx sub-band. Unfortunately, the t-dimension behaves differently in this respect, so that e.g. (h, v, t) and (h, t, v) cannot be identified. Fig. 3 shows the resulting context assignment. Context types are listed in the header line. Two context types in the same column (e.g. 0+- and 0-+) indicate that the same context number is used but -1 is predicted instead of 1 for the second context type. Likewise, a negative context number in the table indicates the same. The author admits that 49 contexts is still a high number but is convinced that this is more or less unavoidable.

Regarding anisotropic wavelet packet decompositions, another problem arises. These sub-bands are not inherently of an LLL, ..., HHH type. Since this type is used to determine the contexts for arithmetic coding, an equivalent has to be found. This can be done by finding the type of the last filter (high- or low-pass) for each dimension that was used to produce the sub-band. Together, these types form the desired sub-band type.

h	0	> 0	0	0	0	> 0	> 0	> 0
v	0	0	> 0	0	> 0	0	> 0	> 0
t	0	0	0	> 0	> 0	> 0	0	> 0
a	0	1	2	3	4	5	6	7

vt	0	> 0	0	0	0	> 0	> 0	> 0
th	0	0	> 0	0	> 0	0	> 0	> 0
hv	0	0	0	> 0	> 0	> 0	0	> 0
b	0	4	5	6	1	2	3	7



a	0	0	0	0	0	> 0	> 0
b	0	0	0	0	> 0	= a	≠ a
d	0	0	> 0	> 0			
sub-band	xxL	xxH	xxL	xxH			
context	0	1	2	3	b+3	a+10	a+18

Fig. 2. Contexts for arithmetic coding of 3-D significance information

hvt	000	00+ 00-	0+- 0-+	0+0 0-0	0++ 0--	+-- -++	+-0 -+0	+-+ -+-	+0- -0+	+00 -00	+0+ -0-	++- --+	++0 --0	+++ ---
LLL	48	48	48	48	48	48	48	48	48	48	48	48	48	48
HHH	0	1	2	1	3	4	2	-4	2	1	3	4	3	5
HLL	0	6	7	9	11	13	15	-14	8	10	12	16	17	18
LHL	0	6	8	10	12	14	-15	13	7	9	11	16	17	18
HLH	0	19	20	22	24	26	28	-27	21	23	25	29	30	31
LHH	0	19	21	23	25	27	-28	-26	20	22	24	29	30	31
LLH	0	32	33	34	35	36	0	-36	33	34	35	37	38	39
HHL	0	40	41	42	43	44	0	-44	41	42	43	45	46	47

Fig. 3. Contexts for arithmetic coding of 3-D sign information

4. EXPERIMENTAL RESULTS

Experiments were conducted on many well-known video sequences. Representative results are shown for *claire* and *foreman*. All sequences are in the QCIF format (176×144 pixel). The first 64 frames (Y-component only) of each sequence were selected for the 3-D wavelet transform.

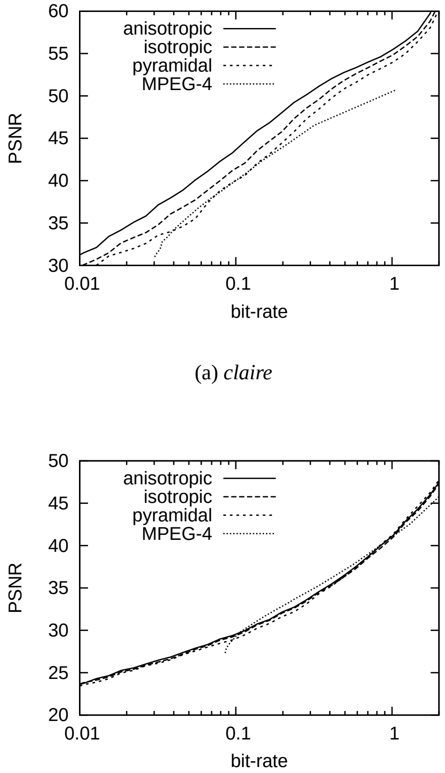
As MPEG-4 codec, the implementation of the verification model from December 1999 was used. A frame group with three P-frames and 2 B-frames per P-frame was chosen, which results in IBBPBBPBBP as encoding scheme. Because we only consider monochrome (grey) videos, the MPEG-4 codec was modified not to encode any colour data, i.e. only Y- but no U- and V-components were encoded.

Equal quantisation parameters were used for all frame types in order to have all frames compressed at approximately equal quality. This was done for reasons of fair comparison since this is also the case in 3-D SMAWZ. Of course, quantisation matrices are different for I-, P- and B-frames – the defaults are used here. The PSNR is calculated for whole video sequences by means of the mean squared error of the 3-D set of video pixels.

While in image coding the gain in rate-distortion performance achieved by arithmetic coding is usually about 0.5 dB, we get 1 to 4 dB difference here (not shown). The reason might be that 3-D contexts deliver more information for the prediction of significance and sign bits than 2-D contexts. Additionally, this effect is greater for sequences with low motion content.

As a result, 3-D SMAWZ is a highly competitive video codec as can be seen in Fig. 4. The (high motion) *foreman* sequence is the most hard to compress for 3-D SMAWZ. Isotropic as well as anisotropic wavelet packets are not able to improve the compression significantly. However, SMAWZ is only about 1dB below MPEG-4 in the worst case. On the other hand, low motion sequences such as *claire* result in a performance gain of up to 4 dB.

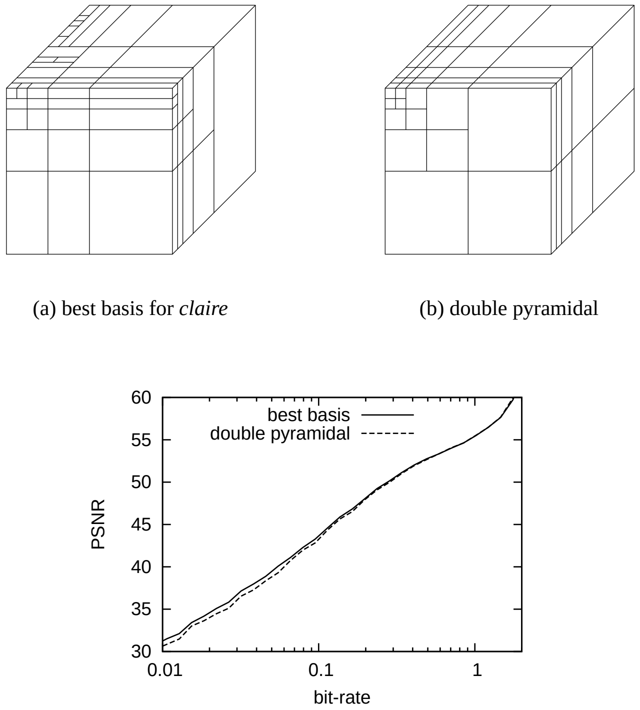
Fig. 5(a) shows the decomposition structure generated by the best-basis algorithm. For some reason, this basis is very close to a special anisotropic wavelet packet basis shown in Fig. 5(b) which is generated by a 1-D temporal pyramidal plus a 2-D spatial pyramidal decomposition. This basis is, therefore, called *double-pyramidal*. Fig. 5(c) shows that this basis produces nearly as good results as the best basis. All other sequences tested by the author show similar results. Additionally, the complexity of the double-pyramidal decomposition ($\frac{14}{3}fn \approx 4.7fn$, where f is the filter length and n is the data size) is not significantly higher than that of the pyramidal decomposition ($\frac{24}{7}fn \approx 3.4fn$). Compared to the complexity of the best-basis algorithm ($O(n(\log n)^3)$), the double-pyramidal basis should be the basis of choice for 3-D wavelet video coding. Note that many 3-D video codecs actually use this basis because constraints of the system delay require small block sizes,



(a) *claire*

(b) *foreman*

Fig. 4. Rate-distortion performance of 3D-SMAWZ with pyramidal WT, isotropic and anisotropic wavelet packets compared to MPEG-4 for the *claire* and *foreman* sequences



(a) best basis for *claire*

(b) double pyramidal

(c) performance comparison

Fig. 5. The anisotropic decomposition structure for the *claire* sequence produced by the best basis algorithm using the *norm* as cost function (a) compared to the double pyramidal decomposition (b). (c) compares the compression performance

shorter filters and lower decomposition depths in the time dimension. Therefore, the temporal decomposition is separated from spatial decomposition.

5. REFERENCES

[1] S.J. Choi and J.W. Woods, “Three-dimensional subband/wavelet coding of video with motion compensation,” in *Visual Communications and Image Processing ’97*, J. Biemond and E.J. Delp, Eds., San Jose, Feb. 1997, vol. 3024 of *SPIE Proceedings*, pp. 96–104.

[2] B.J. Kim and W.A. Pearlman, “An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT),” in *Proceedings Data Compression Conference (DCC’97)*. Mar. 1997, pp. 251–259, IEEE Computer Society Press.

[3] W.L. Hsu and H. Derin, “Video compression using adaptive wavelet packets and DPCM,” in *Video Data Compression for Multimedia Computing*, H.H. Li, S. Sun, and H. Derin, Eds., pp. 55–94. Kluwer Academic Publishers Group, 1997.

[4] R. Kutil, “A significance map based adaptive wavelet zerotree codec (SMAWZ),” in *Media Processors 2002*, S. Panchanathan, V. Bove, and S.I. Sudharsanan, Eds., Jan. 2002, vol. 4674 of *SPIE Proceedings*, pp. 61–71.

[5] R. Kutil, “The graph structure of the anisotropic wavelet packet transform,” in *Proceedings of the 7th international scientific conference devoted to the 25th anniversary of civil engineering faculty and 50th anniversary of technical university Kosice*, May 2002, pp. 41–47.

[6] R. Kutil, *Wavelet Domain Based Techniques for Video Coding*, Ph.D. thesis, Department of Scientific Computing, University of Salzburg, Austria, July 2002.

[7] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158 – 1170, 2000.