Transformation Techniques for Future Video Coding

Kenneth Vermeirsch, Stijn Notebaert, Jan De Cock, Peter Lambert, Rik Van de Walle
Ghent University – IBBT
Department of Electronics and Information Systems – Multimedia Lab
Gaston Crommenlaan 8 bus 201, 9050 Ghent-Ledeberg, Belgium
{ kenneth.vermeirsch | stijn.notebaert | jan.decock | peter.lambert | rik.vandewalle } @ugent.be

ABSTRACT

In this paper we present a number of advanced concepts with respect to the transformation of residual blocks in video coders, which go beyond what is incorporated in today's video coding standards. Some of these concepts will undoubtedly be adopted as part of future standards. We discuss directional transforms for extrapolation-based prediction schemes, shape-adaptive transformation for object-based coding, and large transform sizes. For the latter we provide in-depth coding efficiency results which clearly illustrate the potential benefit, especially for high definition source material, which will dominate the requirements of tomorrow's video coding standards. The compression efficiency gains that can be achieved with large block transforms range from 6 to 25 %. Finally we also comment on the paradigm of coupling partition and transform areas, a principle which can be applied to block transforms as well as to shape-adaptive transforms for object-based coding.

Categories and Subject Descriptors

E.4 [**Data**]: Coding and Information Theory

Keywords

Video coding, transform, DCT, partitioning, coding efficiency

1. INTRODUCTION

Today is an exciting time in the realm of video coding. As standards bodies are starting to evaluate whether the time is right to start a new standardization effort, many old ideas are being reevaluated, and completely new techniques are being invented.

In this paper we will focus on the aspect of transformation, and present a number of advanced transform coding schemes which we feel may play an important role in the development of future video coding standards. Specifically, we will discuss directional transforms, shape-adaptive transforms, and large-size DCT transforms, with the main focus on the latter.

As future coding standards for broadcast applications will place a greater focus on high-definition material (1080p and up), it is expected that larger transform sizes will become important. In Section 4 we discuss the use of larger-sized DCT transforms and supply experimental evidence in support of doing so.

Finally, in Sect. 5 we discuss the issue of maintaining a tighter coupling between partition and transform sizes. Doing so reduces the signaling overhead that is required and applies to shape-adaptive transforms as well as to large transforms.

Mobimedia'09, September 7-9, 2009, London, UK.

2. DIRECTIONAL TRANSFORMS

The DCT transform kernel may not be the most appropriate choice in some cases. If the residual signal shows a strong correlation in some direction and much weaker correlation in others, a directional transform can perform much better. If the direction of the strongest correlation is known in advance, a Karhunen-Loève transform kernel can easily be defined.

As it turns out, the spatial prediction mechanism used for intracoded blocks in H.264/MPEG-4 AVC satisfies exactly these conditions. It is based on an extrapolation of previously coded pixels into the region to be coded, and nine different directions of extrapolation can be chosen by the encoder [15]. Since the prediction signal will be highly directional, so will the residual signal. This correlation behavior is illustrated in Figs. 1 and 2. Note that the horizontal and vertical correlations are similar but exhibit a 90° phase shift. At the time of transformation the direction is known, so specialized transform kernels can be constructed for each of the intra prediction directions.

The use of directional transforms for H.264/MPEG-4 AVC-style intra prediction has been proposed in ITU-T [17], where it has shown compression gains of around 10 %, and will be a useful technique in future video coding when used in conjuction with spatial extrapolation-based predictors.

3. SHAPE-ADAPTIVE TRANSFORMATION

3.1 Discussion

Shape-adaptive transformation techniques in general allow arbitrarily-shaped regions in images to be transformed separately. This is a key enabler for *object based coding,* which leads to many interesting possibilities, such as the

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7990
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7990

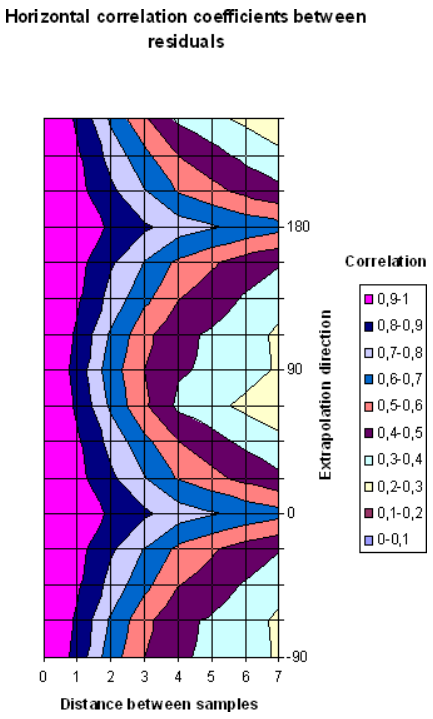


Figure 1: Horizontal correlation coefficients between residual samples vs. extrapolation direction w.r.t. the horizontal image axis. As measured in the City sequence.

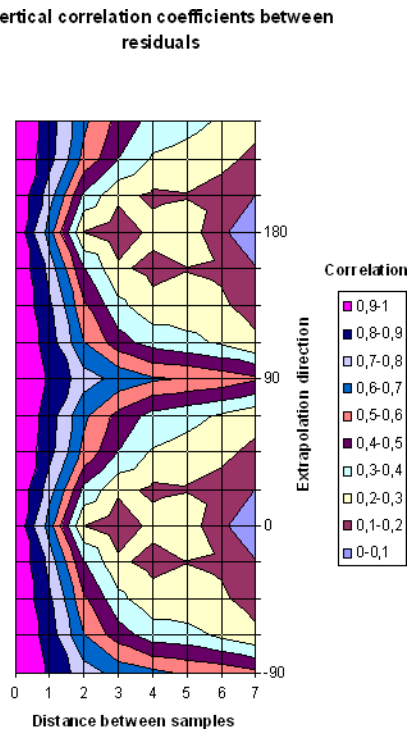


Figure 2: Vertical correlation coefficients between residual samples vs. extrapolation direction w.r.t. the horizontal image axis. As measured in the City sequence.

ability to define very precise regions of interest, which can then be coded at a higher quality. Also, shape-adaptive transforms will help reduce ringing artifacts at object boundaries.

During standardization of MPEG-4 Visual an attempt was made to introduce object-based coding. However, the main obstacle consists in finding a good and reasonably fast segmentation algorithm. The difficulty in finding this has left MPEG-4 object coding largely unused in practice.

In the future we may see a re-appreciation of object-based coding approaches. However, the problem of finding a good segmenter remains. This will require a substantial research effort.

Concerning the actual transformation of arbitrarily-shaped regions, there are numerous techniques to be found in the literature; we will now briefly review these.

3.2 State of the art

The best-known transform for arbitrary regions is the shape-adaptive separable DCT (SA-DCT) as defined [12] by Sikora. It is popular because of its simplicity and its good performance in terms of coding gain, and it was chosen for standardization in MPEG-4 object-based coding.

Aside from the Sikora transform, there are other shape-adaptive transformations that use alternative approaches. Gilge et al. [7] derived a generalized orthogonal transform (GOT) suitable for image coding applications. Stiller and Konrad [13] derived basis functions from a stochastic model. Desai [4] used so-called cosine filling to decompose the shape into an over-complete set. Chang and Messerschmitt [2] evaluated different 'padding' methods to apply a block DCT to arbitrary shaped segments. Kaup and Aach [8] used 2-D shape-independent basis functions defined on a rectangle circumscribing the given image segment. Donescu et al. described a generalized formulation of SA-DCT in [6].

Li et al. redesigned the shape-adaptive DCT specifically for H.264/MPEG-4 AVC integer transform of intra blocks, showing that shape-adaptivity can be ported to H.264 with minimal impact on the transform design [9].

4. LARGE TRANSFORM SIZES

4.1 State of the art

Successful video coding standards in the past have always used an 8×8 DCT for coding the residual signal; in the H.264/MPEG-4 AVC standard, a 4×4 integer approximation of DCT [10] is used in the Baseline, Main and Extended profiles, while the High Profile offers adaptive 4×4 / 8×8 integer transforms [16]: in macroblocks containing no partitions smaller than 8×8 a flag is included in the bitstream which indicates the size of the inverse transform that should be used in the decoder.

In general a larger transform provides a better decorrelation of signals, while smaller transform sizes can help reduce block artifacts. The main argument against large transforms, however, has always been one of computational complexity. In newer video standards, however, entropy coding,

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7990
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7990

I/O and rate-distortion optimization are the performance bottleneck rather than the transformation. Moreover, the increasing prevalence of high definition in broadcast video formats brings with it a greater smoothness (on average) of the signal within blocks of a given size.

The idea of adaptive transform sizes has been applied to still image coding in the past. A quadtree-based image coder using square DCTs of various sizes was presented in 1989 by Chen [3]. Similarly, Dinstein et al. [5] employ a bottom-up clustering approach using nine different rectangular block sizes ranging from 8×8 to 32×32. They report better preservation of fine detail and a reduced blocking effect as a result of using larger transform sizes.

Most recently, the use of larger block sizes — for motion compensation as well as for the transform — has garnered increasing levels of interest within the exploration activities of video standardization bodies. To illustrate the benefit of using larger transforms, we present results based on an extension of H.264/MPEG-4 AVC where we allow the encoder to choose larger transforms instead of the default 8×8 DCT.

4.2 Adaptive-size DCT coder

This coder allows either a large DCT or 8×8 square DCT to be chosen for each macroblock. We avoid transforming multiple partitions at once, so the larger transform will be 16×16, 16×8, or 8×16 samples in size, depending on the partition mode with which the macroblock is coded. In this coder, an extra flag has to be included in the macroblock header; this consumes some additional bit rate. The flag is entropy coded using the CABAC context-based adaptive binary arithmetic coder of H.264/MPEG-4 AVC [11]. This coder is able to exploit causal spatial correlation when coding this flag, reducing the rate overhead.

In this investigation only partition sizes of 8×8 luminance samples and greater are considered since smaller partitions are of little importance, especially in future applications which will be dominated by high definition and beyond-HD source material. Hence, the transform sizes that need to be provided are 8×8, 16×8, 8×16 and 16×16. We use a separable DCT, so we really only need an 8-point and a 16-point 1-D DCT implementation to perform all necessary 2-D transform operations. The DCT employed in this investigation is the well-known orthogonalized Type-II DCT with kernel matrix

$$(D_N)_{mn} = A \cdot \cos \frac{\pi(m+\frac{1}{2})n}{N}, \quad m,n = 0,\ldots,N-1, \quad (1)$$

with $N \in \{8,16\}$, and $A = \sqrt{\frac{1}{N}}$ if $n = 0$ and $A = \sqrt{\frac{2}{N}}$ otherwise. Using these 1-D DCTs, the transformation of a two-dimensional residual signal $\mathbf{X}_{P\times Q}$ of a partition of size $P\times Q$ is performed in two steps ('row' and 'column' transforms):

$$\mathbf{Y} = \mathbf{D_P^T} \cdot \mathbf{X} \cdot \mathbf{D_Q} \quad (2)$$

After transformation of the residual signal, the coefficients $\mathbf{Y} \in \mathbb{R}^{P\times Q}$ are quantized using a deadzone uniform quantizer and then compressed using the context-based, adaptive binary arithmetic CABAC coder [11]. New progressive zigzag scan patterns were defined for run-length coding of the transform coefficients, as illustrated in Fig. 3 for 16×8 blocks.

If desired, H.264-style integer approximations could easily be defined for these new transform sizes [10, 16].

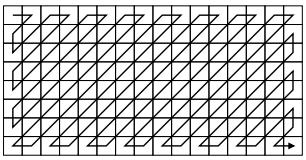


Figure 3: Scan pattern for 16×8 blocks.

4.3 Experimental results

We compare our coder against the 8×8 H.264/MPEG-4 AVC transform and quantizer (as described in [16]). In all cases the H.264/MPEG-4 AVC loop deblocking filter is switched off, since it is not designed for transform sizes other than 4×4 and 8×8. An open GOP structure (IPP...) is used. Rate and distortion points are taken at four different quality parameters (QPs): {23, 28, 33, 38}.

Results for a number of test sequences are gathered in Table 1, under "Adaptive." The numbers indicated are averages; they are derived by fitting a third-order polynomial to the R-D curves and, by integration, calculating the area between the proposed and anchor curve fits. Δ BR expresses this area as an average bit rate change; ΔQ as an *equivalent* average improvement in objective reconstruction quality (PSNR) of the luminance channel. More information on this procedure may be found in [1].

Average bit rate reductions range from 5 to 18 %. Though it is not evident from the averages, it can be seen in the example R-D curves in Fig. 4 that the highest gain for many sequences is obtained at medium quality; this is explained as follows. At low quality (strong quantization), many partitions will be coded without residual data (only motion vectors are sent), which obviously leads to the same result regardless of transform type. At higher quality settings, and for some sequences (e.g., 'City'), the rate-distortion optimizer will be more in favor of smaller partitions (i.e., 8×8), again leading to almost equal results (8×8 DCT vs. 8×8 H.264/MPEG-4 AVC transform). Hence, the new transforms will be used most frequently when medium-quality settings are selected.

It can be clearly seen from the averages in Table 1 that larger-sized transforms work better at higher picture sizes.

The complexity increase of the transforms themselves is minimal compared to the normal 8×8 DCT, especially because of the fact that they are separable. All known optimizations and parallelizations for DCT can be used.

5. COUPLING TRANSFORM AND PARTITION SIZES

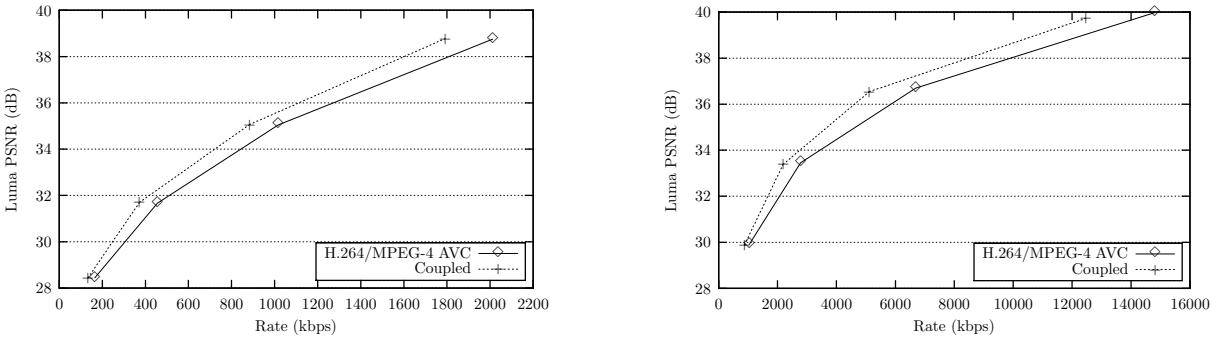
In video coding, when considering the residual signal after motion compensation, it is a logical choice to take entire partitions as the transform support, since the statistical characteristics of the residual signal are uniform throughout the partition. When using several smaller N×N transforms on that same partition, this uniformity cannot be fully exploited.

Partition sizes in older coding standards were very limited, e.g. only 16×16 luminance samples (as in MPEG-2) or adaptive 16×16 / 8×8 (MPEG-4 Part 2, VC-1); this immediately limits the usefulness of the proposed approach. In H.264/MPEG-4 AVC, however, many more partition types are defined; this is illustrated in Fig. 5. This figure shows

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7990
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7990

Table 1: Coding Efficiency Results.

Sequence name	Format	No. of frames	Adaptive		Coupled	
			ΔBR (%)	ΔQ (dB)	ΔBR (%)	ΔQ (dB)
Coastguard	CIF	299	-9.51	0.416	-15.88	0.715
Foreman	CIF	299	-5.37	0.209	-8.25	0.326
Paris	CIF	299	-6.15	0.336	-6.06	0.325
Silent Voice	CIF	299	-8.99	0.416	-10.21	0.476
Stefan	CIF	299	-6.78	0.366	-9.45	0.512
CIF average	CIF	1495	-7.36	0.349	-9.97	0.471
BigShips	720p	599	-7.56	0.231	-10.14	0.312
City	720p	899	-10.59	0.308	-12.38	0.356
Crew	720p	599	-8.18	0.251	-10.40	0.321
Harbour	720p	599	-18.27	0.762	-25.83	1.138
Night	720p	459	-9.35	0.346	-9.38	0.342
ShuttleStart	720p	599	-11.37	0.384	-13.84	0.475
720p average	720p	2556	-10.89	0.380	-13.66	0.491



(a) Coastguard CIF, 15.88 % avg gain

(b) Harbour 720p, 25.83 % avg gain

Figure 4: Rate-distortion curves for coding efficiency.

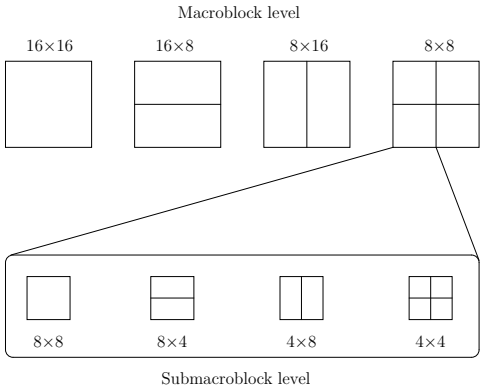


Figure 5: The H.264/MPEG-4 AVC quadtree of partition sizes.

the so-called "partition quadtree" of H.264/MPEG-4 AVC, which is a two-level tree rooted at the 16×16 partition size and with the lower level starting in the 8×8 node. There are a total of seven different sizes available. In this paper, however, we only consider partitions of 8×8 luma samples and larger.

In this section we eliminate the adaptivity in the encoder regarding the transform size. The DCT sizes must now strictly follow the partition sizes that were used during motion compensation. As an immediate benefit, signaling overhead is no longer incurred.

Interestingly, this coder has better performance almost everywhere, despite the loss of coding options. Numbers are provided in Table 1 under "Coupled." In fact, it appears that the coder we discussed in Sect. 4.2 will almost always select the large transform when partitions larger than 8×8 are used. Therefore, the bit rate expended to the additional flag is mostly wasted.

5.1 Shape-adaptive transformation

A coupling of partitions and transform areas also makes sense in the field of shape-adaptive transforms, as this is a natural extension of object-based coding in which not only the transform but also the motion compensation process is object-based. This will require a generalization of the motion partitioning e.g. as we described in [14]. The benefit consists of a better temporal prediction along boundaries of moving objects.

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7990
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7990

6. CONCLUSION

There is much room for improvement beyond the range of transform coding tools that are deployed in video coding standards today. In this paper we highlighted just a few promising new tools and ideas that are currently being investigated by researchers and standards committees.

We have examined the concept of using larger transform sizes in greater detail, and have shown that a bit rate reduction of 5 to 18 % can be achieved when the encoder can choose, for each macroblock, between a regular 8×8 transform and a larger one. However, by applying the idea of coupling partition and transform areas, the signaling of the best transform mode can be omitted and the bit rate reduction can thus be increased to 6 to 25 %.

Coupling of partitioning and transformation can also be applied to shape-adaptive transformation, thereby extending the object-based coding approach from transformation and quantization into motion compensation and estimation.

7. ACKNOWLEDGMENTS

The research activities described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research–Flanders (FWO–Flanders), and the European Union.

8. REFERENCES

[1] G. Bjøntegaard. VCEG-M33: Calculation of average PSNR differences between RD-curves. ITU-T Q.6/16 VCEG, http://wftp3.itu.int/av-arch/video-site/0104_Aus/, Apr. 2001.

[2] S. F. Chang and D. G. Messerschmitt. Transform coding of arbitrarily-shaped image segments. In *Proc. ACM Int. Conf. on Multimedia*, pages 83–90, 1993.

[3] C.-T. Chen. Adaptive transform coding via quadtree-based variable block size DCT. In *Proc. IEEE Int. Acoustics, Speech, Signal Processing ICASSP*, volume 3, pages 1854–1857, May 1989.

[4] U. Y. Desai. DCT and wavelet based representations of arbitrarily shaped image segments. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 1995.

[5] I. Dinstein, K. Rose, and A. Heiman. Variable block-size transform image coder. *IEEE Trans. Commun.*, 38(11):2073–2078, Nov. 1990.

[6] I. Donescu, O. Avaro, and C. Roux. A shape-adaptive transform for object-based coding. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sept. 1996.

[7] M. Gilge, T. Engelhardt, and R. Mehlan. Coding of arbitrarily shaped image segments based on a generalized orthogonal transform. *Signal Processing: Image Communication*, 1(2):153–180, 1989.

[8] A. Kaup and T. Aach. Coding of segmented images using shape-independent basis functions. *IEEE Trans. Image Process.*, 7(7):937–947, July 1998.

[9] X. Li, E. Edirisinghe, and H. Bez. Shape adaptive integer transform for coding arbitrarily shaped objects in H.264/AVC. In *Visual Communications and Image Processing 2006*, volume 6077. SPIE, 2006.

[10] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky. Low-complexity transform and quantization in H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):598–603, July 2003.

[11] D. Marpe, H. Schwarz, and T. Wiegand. Context-based Adaptive Binary Arithmetic Coding in the H.264/AVC video compression standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):620–636, July 2003.

[12] T. Sikora and B. Makai. Shape-adaptive DCT for generic coding of video. *IEEE Trans. Circuits Syst. Video Technol.*, 5(1):59–62, Feb. 1995.

[13] C. Stiller and J. Konrad. Region-adaptive transform based on a stochastic model. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 1995.

[14] K. Vermeirsch, J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle. Increased flexibility in inter picture partitioning. In *Proc. of Int. Symp. on Multimedia Sig. Proc. (MMSP)*, Oct. 2008.

[15] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):560–576, 2003.

[16] M. Wien. Variable block-size transforms for H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7):604–612, July 2003.

[17] Y. Ye and M. Karczewicz. VCEG-AG11r1: Improved intra coding. ITU-T Q.6/16 VCEG, http://wftp3.itu.int/av-arch/video-site/0710_She/, Oct. 2007.

Digital Object Identifier: 10.4108/ICST.MOBIMEDIA2009.7990
http://dx.doi.org/10.4108/ICST.MOBIMEDIA2009.7990