Dominik Engel, Andreas Uhl (2003).
Adaptive image compression of arbitrarily shaped objects using wavelet packets. In Proceedings of the 23rd International Picture Coding Symposium 2003 (PCS'03), pp. 283–288.

ADAPTIVE IMAGE COMPRESSION OF ARBITRARILY SHAPED OBJECTS USING WAVELET PACKETS

Dominik Engel and Andreas Uhl

University of Salzburg, Department of Scientific Computing,
Jakob Haringer Strasse 2, Salzburg
{dengel, uhl}@cosy.sbg.ac.at

ABSTRACT

With the emergence of objectbased image coding standards and the successful use of adaptive image coding techniques for various applications, the combination of both fields in objectbased adaptivity is only logical, as it promises not only better compression efficiency, but also increased functionality.

In this work, we present approaches of objectbased and adaptive image compression using wavelets. We discuss recent propositions how to adapt established wavelet techniques to arbitrarily shaped objects and compare different methods and implementations. We show how objectbased adaptivity can be achieved and how well it performs compared to methods in a non objectbased framework.

1. INTRODUCTION

Image and video coding with wavelet methods have been paid much attention in the scientific community in recent years. Since the finalization of the JPEG-2000 standard [1] at the latest, wavelet methods are now well established in image compression. The MPEG-4 standard makes use of the wavelet transform in its visual texture coding (VTC) module [2]. Especially for low bitrates, wavelet methods outperform longer established techniques like the discrete cosine transform (DCT) [3].

The wavelet transform itself is not a single algorithm, but a whole set of different approaches, ranging from the classical pyramidal Mallat decomposition to adaptive transformations, like the wavelet packet transform (WPT) using the Best Basis Algorithm (BBA) [4], each with its area of application. The WPT, for example, has been successfully adopted for the compression of oscillatory signals.

There are two reasons why objectbased compression may prove to be of advantage.

- Increased functionality
- Increase compression performance

Whereas increased functionality is inherently achieved be the objectbased approach, increased compression performance by adaptive objectbased coding is not necessarily achieved. The idea here is to segment an image into objects of different characteristics and then adaptively encode each object separately, thus increasing compression performance. Even if this aim is not met and compression efficiency remains at least constant, the approach has to be considered a success if the overhead introduced by the increased functionality can be balanced by using adaptivity for increased compression efficiency.

This paper is organized as follows. In sec. 2, we will discuss important foundations for objectbased adaptivity and wavelet packets. We will discuss the necessary extensions for using zerotree coding with wavelet packets and arbitrarily shaped objects in sections 3 and 4, respectively. In sec. 5 we present the framework developed for adaptive objectbased coding and discuss the related results. Sec. 6 concludes the paper.

This work was supported by the Austrian Science Fund (FWF), project no. P13732

2. SHAPE ADAPTIVE WAVELET PACKET TRANSFORM

2.1. Wavelet Packets

The wavelet packet transformation (WPT) is a generalization of the pyramidal wavelet decomposition. In WPT the approximation subband is not the only subband to be decomposed further. Thus, for a specific level of maximum decomposition depth, there are many possible WP-structures – the WPT is an *overcomplete* library of bases. Each structure represents a wavelet basis and therefore a specific subband decomposition. Fig. 1 shows a possible WP-decomposition structure.

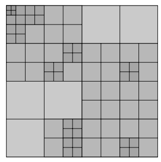


Fig. 1. WPT structure

The choice of one particular decomposition can be adapted to the properties of the image to be transformed. We use the best basis algorithm (BBA) [4] to select the best suited decomposition. In BBA, first a full wavelet packet decomposition for the target level is generated. BBA uses costfunctions to decide where the decomposition structure should be modified. Various costfunctions can potentially be used – we limit our testruns to additive costfunctions. The BBA always produces an optimal basis with regard to the used measure for orthogonal filters. For biorthogonal filters, to which we also apply the same algorithm, exclusive optimality is not guaranteed. However, from a pragmatic perspective, the results are quite acceptable.

In this way, WPT has been successfully used for compressing images with oscillatory patterns, like the famous testimage "*Barbara*" [5, 6, 7] or fingerprint images [8]. However, as optimality is only achieved with regard to the addressed measure, PSNR-performance varies for different costfunctions and different classes of images.

A tree representation is well suited to describe a single decomposition. The *"decomposition tree"* is a quadtree representing the structure of the wavelet decomposition. Each of its nodes is associated with a subband and represents the binary decision whether this subband is decomposed further or not. Note that it makes sense to distinguish the tree representing the decomposition structure from the actual coefficients. In our implementation the coefficients are stored separately from the decomposition information. We will generically call the structure holding the coefficients *"coefficient tree"* (although the actual implementation does not necessarily have to be a tree). When we discuss zerotree coding for WPT, we will introduce another tree, the *"similarity tree"*, which orders the subbands in the decomposition tree for zerotree coding.

2.2. Shape-adaptive Wavelet Transform

There are several propositions on how to perform the transformation of arbitrarily shaped objects. Since the wavelet transformation of rectangular regions has been very well researched, the use of padding techniques would seem rewarding. By finding the bounding box of an object and padding values into the region that is not contained in the object-mask, the problem of arbitrary shapes is reduced to a rectangular transformation. However, because in this approach more pixels than contained in the original object have to be coded, it is not efficient. Additionally, perturbation of the frequency pattern of the objects is artificially introduced and shows negative impact on the performance of the BBA. For better efficiency a way has to be found to deal with arbitrary shapes directly and to transform them without producing overhead pixels.

In our test setup we use the shape adaptive DWT (SA-DWT) proposed by Li [9] to produce WP-transformations of arbitrarily shaped objects. In the following we will outline the most eminent features of SA-DWT and introduce important terminology. For a more detailed description see [9]. SA-DWT produces exactly the required amount of coefficients and preserves coefficient positions at the same time. It has been adopted for the MPEG-4 Visual Texture Coding (VTC) module for coding of still textures. Transforming objects of arbitrary shape always involves the problem of having to transform oddsized segments. When biorthogonal filters are used, oddsized segments can be transformed without any splitting in SA-DWT.

Subsampling in SA-DWT can be done at even or odd position. Following [9], we will call subsampling at even positions "*even* subsampling strategy" and subsampling at odd positions "*odd* subsampling strategy". Because odd-sized biorthogonal filters introduce a phaseshift during the analysis step, different subsampling strategies have to be used for lowpass and highpass subbands, i.e. if the lowpass subband is subsampled at even positions, the highpass subband has to be subsampled at odd positions and vice versa. We denote the subsampling strategy for this kind of filters as "*even-odd* subsampling" and "*odd-even* subsampling", where the first part refers to the strategy used for the lowpass subband and the second part refers to the strategy for the highpass subband. For orthogonal filters, the same subsampling strategy is applied for both subbands.

We further distinguish two subsampling *modes*, namely *"global"* and *"local"*. For global subsampling the local subsampling strategy depends on the start of the signal relative to the left and upper border (for horizontal and vertical transformation, respectively) of the bounding box of the object shape. In this case, subsampling is always done at even or odd offsets, respectively, from the border. From a local perspective, this may result in local even or local odd subsampling.

3. EXTENDING ZTC TO WAVELET PACKETS

The extension of zerotrees to wavelet packets is not a trivial task. Some of the constraints that are given in the pyramidal decomposition are missing in a wavelet packet decomposition. The hypothesis that the magnitude of coefficients decays with frequency is to be questioned [10]. Furthermore, a more sophisticated definition of the parent-child relationship between subbands has to be used, as the relationship is not as clear as in the pyramidal case. With wavelet packets it is possible that a child subband is at a coarser scale than its parent, i.e. a coefficient in the child subband has multiple possible parents, resulting in a *parenting conflict* [11].

Xiong et al. [6] propose to avoid this parenting conflict in the first place by considering it in the selection of the best basis and merging children that have parents at a coarser scale. However, this approach severely limits the flexibility of the best basis algorithm to adapt to a given signal.

Rajpoot et al. [11] propose a zerotree structure called *"compatible zerotree"* to resolve the parenting conflict. For the initial creation of a compatible zerotree, the parent-child relationship is defined between whole subbands (unlike the spatial orientation trees of SPIHT, where the relationship is defined between single coefficients, see [12]). Accordingly, a node in a compatible zerotree refers to a whole subband. After the initial tree has been created, its nodes are reordered to resolve parenting conflicts. The prerequisite for the use of compatible zerotrees is that the approximation subband is at the coarsest scale.

Our own tests are based on SMAWZ [13], which uses a zerotree like structure called *"similarity trees"*. The similarity tree resembles the compatible zerotree in that is defines a parent-child relationship between whole subbands. However, the parental conflict is not resolved by moving coarse scale subbands up the tree. If a node C has multiple possible parents $P_i,\ i = 0, \dots 3$, only P_0 (i.e. the subband of lowest frequency range) is chosen as parent, while the others are marked as childless. Note that, other than with compatible zerotrees, no restriction as to the scale of the approximation subband needs to be taken into account.

By following a set of rules (cf. [13]), a hierarchical representations of the subbands in the decomposition tree is produced. Even though the resulting similarity tree only connects leaf subbands of the decomposition tree (see sec. 2), the rules also use nodes that are not leaves in intermediate stages. It is important to point out that even though the whole structure of the decomposition tree is used, there is never the need to access the data contained in the coefficient tree. The algorithm for creating similarity trees is purely structural.

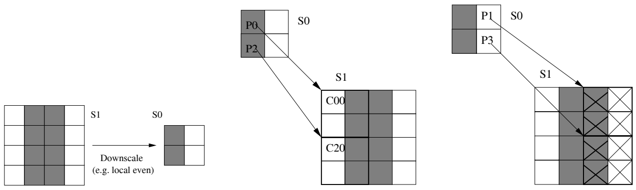
The resulting similarity tree, which connects the leaf subbands of the decomposition tree, is used to encode the actual wavelet coefficients with a significance map based approach and finally arithmetic coding.

4. EXTENSIONS TO ZTC FOR ARBITRARY SHAPES

4.1. Extensions for Bitmasks

Let S_0, S_1 be subbands, where S_1 is one scale finer than S_0, $|S_0| = 4$ and $|S_1| = 16$. Let B_1 be the bitmask of an arbitrary shape in S_1. Let B_0 be the bitmask of the same arbitrary shape in S_0 that is constructed according to the chosen downscaling strategy and mode (e.g. *local-even*, Fig. 2.a).

Further let $P_i : i = 0, \dots, 3, P_i \in S_0$ be the coefficients contained in S_0 and $C_{i,j} : i = 0, \ldots, 3, \;\; j = 0, \ldots, 3$ be coefficients contained in S_1 with $parent(C_{i,j}) = P_i$. Assume that P_i significant $\forall\, P_i \in B_0$ and $C_{i,j}$ significant $\forall\, C_{i,j} \in B_1$.



(a) Downscale (b) OK (c) Problem

Fig. 2. Bitmasks and zerotrees

In Fig. 2.b the zerotree relationship is shown for P_0 and P_2. Both, P_0 and P_2 are contained in B_0. Even though not all child-coefficients at a finer scale are contained in B_1 no problem occurs for zerotree coding. However, in Fig. 2.c a case is shown where special treatment is necessary. Neither P_1 or P_3 are contained in B_0, but for $i = 1, 3$ the following equation holds

$$\exists C_{i,j} : \; parent(C_{i,j}) = P_i, \; C_{i,j} \in B_1, \; C_{i,j} \text{ sig.} \qquad (1)$$

This is a situation that cannot be deduced by the decoder, if the parent coefficients that are not contained in the bitmask are not encoded at all. A possible solution is to encode the parent coefficients with a special symbol in this case, denoting a position not contained in the bitmask but with significant child-coefficients. However, this approach lowers the efficiency of entropy coding.

With regard to PSNR performance, an approach is preferable that allows the decoder to deduce the positions in such a case without the need for additional symbols. This requires some preprocessing and thus has an impact on coding time. In our test setup, which is aimed at gains in coding efficiency and not at gains in coding time, we chose the following approach that is performed by encoder and decoder alike prior to the actual encoding process:

Positions of coefficients with children fulfilling eq. (1) are marked recursively, starting in the approximation subband.

For a coefficient P_i:

- If P_i is not a leaf node, process all of its children $C_{i,j}$ first.
- If the marking process returned false for all children, return *false*. Otherwise, return *true* and if P_i is not contained in the bitmask associated with its subband, mark P_i.
- If P_i is a leaf node, return *true* if P_i is significant, *false* otherwise.

In the encoding process the marks are used to determine the children of which positions have to be processed even though the positions themselves are not contained in the bitmask. Note that no information regarding the marked coefficients is encoded at all. In the decoding process the marks are used to determine positions that have significant children, but are not contained in the bitmask (and thus not in the bitstream) themselves. In both, encoding and decoding, the marks are not needed for the actual transformation.

4.2. Impact of Downscaling Strategy on Efficiency of ZTC

According to the zerotree hypothesis [10], the magnitude and with it the importance of wavelet coefficients is indirectly proportional to the frequency, i.e. coefficients in subbands of low frequency are important with a higher probability than coefficients in subbands of high frequency. So a higher amount of coefficients in the lowpass subbands has a good chance of producing a gain in the efficiency of zerotree coding (cf. [9]).

With the choice of subsampling mode and strategy the number of coefficients in lowpass and highpass portions of a decomposition can be influenced to a certain degree. Using *local* subsampling mode in conjunction with *even-odd* subsampling strategy guarantees that the number of coefficient in the lowpass subband are equal or (one) more than in the highpass subband for one step of 1D-decomposition.

Fig. 3. High energy patterns in local subsampling, ratio 90

While this is an intriguing approach on a theoretical level, in practice *local* subsampling does have a problem with artificial high frequency components. Constellations exist, where this affects coding quality in a high degree. Take a pair of subsequent horizontal segments which overlap vertically, with start indices i_1, i_2. If $i_1 \bmod 2 = 1$ and $i_2 \bmod 2 = 0$, the phases of the two lines are not compatible, high energy patterns are artificially introduced and PSNR quality is severely affected (see fig. 3 for a reconstructed image). In our testruns, a gain for zerotree processing through local subsampling could only be achieved for a limited set of shapes, where lines as well as columns had a pairwise offset of $2i$, $i \in \mathbb{N}$.

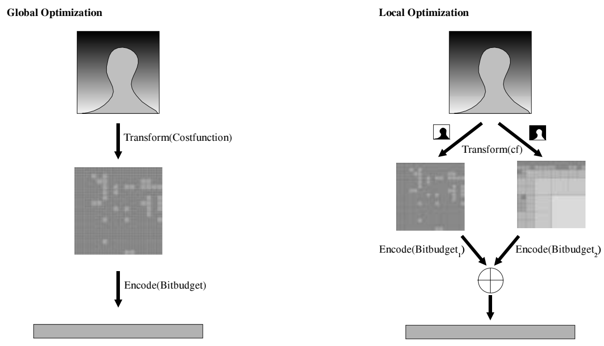
In order to preserve image quality and favor a gain in PSNR, the global approach is to be preferred in such a case, because all pairs of lines are of the same phase here. Note that for odd-length segments there is more than one possibility where to put the smaller of the two resulting segments. For optimal PSNR performance, the positions have to be chosen so that the transformation in the other direction has coefficients of the same phase in each of its segments. Whether globally *even-odd* or *odd-even* subsampling (referring to the strategy of lowpass and highpass subband) produces better results, depends much on the shape used.

5. ADAPTIVE OBJECTBASED IMAGE COMPRESSION

Taking the superior performance of the adaptive WPT for oscillatory patterns, objectbased adaptive coding seems a rewarding approach. The basic idea is segmenting an image into objects on distinction of frequential patterns and then transforming each object on its own. Thus, the wavelet bases are adapted to each object separately instead of the image as a whole.

We will not deal with the segmentation process, but assume an image to be already segmented into objects of different frequential patterns. In fact, we construct test images fitting the above requirements to test whether the adaptive objectbased approach is able to produce better compression results and whether segmentation on the criteria of frequential properties would make sense in the first place. The used testimages are constructed from textures for which the adaptive WP-decomposition produced superior results compared to the pyramidal decomposition.

A schematic comparison between global, i.e. non-objectbased (*"global optimization"*), and objectbased adaptivity (*"local optimization"*), is shown in figures 4.a and 4.b. In the former case, the costfunction is used on the image as a whole, which leads to a single decomposition. The transform coefficients are then encoded and written to the target bitstream. Opposed to this, local optimiza-



(a) Global Optimization

(b) Local Optimization

Fig. 4. Adaptive objectbased image coding

tion does not work on the image as a whole, but optimizes the information cost for each object individually. It is not even necessary to use the same costfunction for all objects. Thus, a decomposition structure is produced for each object in the image. A challenging question is how to distribute the target bitbudget between the objects. The choice of distribution has considerable impact on the coding performance, as will be seen. Based on the allocated bitbudgets, individual bitstreams are produced for the objects, which then have to be merged in some way, producing the final bitstream. There is a range of possibilities how to perform the merging operation, ranging from simple concatenation to more sophisticated methods for producing an embedded bitstream or including ROI coding.

5.1. Bitbudget Allocation

As far as PSNR performance is concerned, a good distribution of the available bitbudget should take the rate-distortion characteristics of each object into account and perform a "cost-benefit analysis". For some objects it is possible to concentrate the energy in just a few transform coefficients, i.e. the object is well suited for compression. Each of these coefficients adds greatly to the quality of the object when transmitted, but further coefficients do not raise the PSNR in a considerable degree anymore. For other objects, the set of coefficients that carry the bulk of the energy is much bigger, and only after a lot of these coefficients have been transmitted the same gain in PSNR is achieved as in the previous case. Therefore, a good method of bitbudget allocation will pay attention to what part of the entire bitbudget an object actually needs and how much is better spent on other objects.

5.1.1. Bitbudget costfunctions (BBCF)

The use of costfunctions for bitbudget allocation is fairly straightforward – basically the bitbudget is assigned to each object proportionally to the value a certain costfunction produces for this particular object.

Let $O_i,\ i = 0, \ldots, n-1$ denote the image objects with their associated object shapes $S_i,\ i = 0, \ldots, n-1$. By applying a costfunction to each object O_i, cost-values $c_i,\ i = 0, \ldots, n-1$ are produced. Depending on the measure that should be addressed, different costfunctions can be used. We will test the following costfunctions for bitbudget allocation: *Variance, Energy, Entropy*.

Let B be the total bitbudget for coding the entire image. B is split into bitbudgets $b_i,\ i = 0, \ldots, n-1$ to be assigned to the objects proportional to their respective costs

$$b_i = B \cdot c_i / \sum_{j=0}^{n-1} c_j . \tag{2}$$

5.1.2. Multipackettree (MPT)

Let $O_i,\ i = 0, \ldots, n-1$ again denote the image objects with their associated object shapes $S_i,\ i = 0, \ldots, n-1$. Further, let D_i be the wavelet packet decomposition structure that is obtained by applying a BBA to O_i. As discussed previously, trees are a common way of representing decomposition structures. We will thus also refer to D_i as *"decomposition tree"*. In parallel to each structural tree D_i, a *"coefficient tree"* exists for each object (see sec. 2).

For the MPT-approach, a similarity tree V_i is created for each object O_i from $D_i, i = 0, \ldots, n-1$, using the algorithm discussed in section on SMAWZ. The similarity trees are grouped to form a *"unified similarity tree" U* with multiple roots,

$$\bigcup_{i=0}^{n-1} V_i . \tag{3}$$

We use SMAWZ to code the unified similarity tree. The difference to the normal mode of operation of SMAWZ is that the coefficients can now be located in different coefficient trees. The ordering of the coefficients is done for all objects together. Thus, a rate-distortion oriented allocation of the bitbudget to the various objects is achieved implicitly, while at the same time an embedded bitstream is produced.

5.2. Common Structure Coding (CSC)

The intersection of the decomposition trees of the various objects in an image is never empty. Especially objects whose characteristics in frequency space do not differ in a high degree have considerable overlap in their decompositions. The idea of CSC is to make use of this fact and to code as much of the objects together as possible in a first step. The individual differences of the objects from the common structure are coded as "refinements" in a second step. Fig. 5 illustrates the analysis process, which is divided into three major steps.

Note that in step 3, the union of the object shapes of all the objects in an image I, does not always cover the whole area of I, depending on the used subsampling mode and strategy. In order to remedy this, the object shapes are enlarged to cover the whole of I.

5.3. Experimental Results

In the following we will compare the performance of global and objectbased adaptivity using the *Ganesh*++-codec. We also include results produced by JASPER (`http://www.ece.uvic.ca/~mdadams/jasper`), a reference implementation of the codec specified in the JPEG-2000 Part-1 standard (ISO/IEC 15444-1). We used the standard *biorthogonal 7,9* filter for both codecs.

For bitbudget costfunctions, as with the additive costfunctions used in BBA, there is no causal link between the measure the BBCF addresses and actual PSNR performance. Thus, as can be

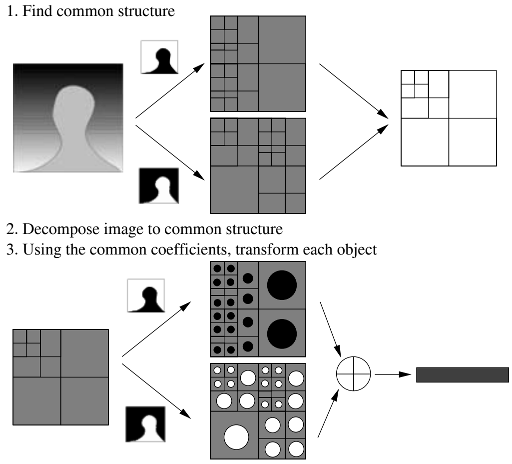
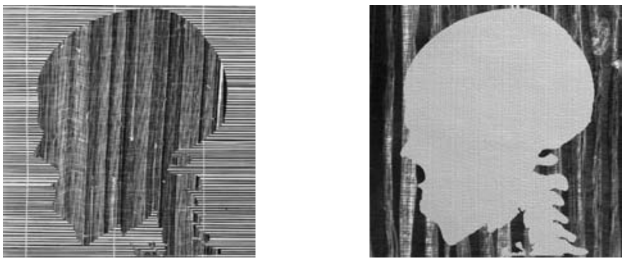


Fig. 5. Common Structure Coding - Analysis



(a) *D105 X D49* (b) *poly_linen X D50*

Fig. 6. Test Images

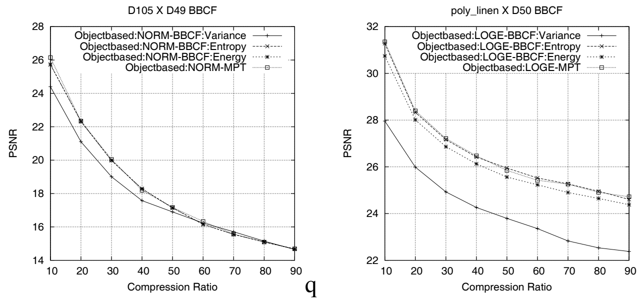
seen in fig. 7, there is not just one costfunction that always guarantees to distribute the bitbudget in an optimal way as regards PSNR performance, but the performance depends on the features of the testimages and how well they can be described using a certain measure.

For most of the tested images, MPT is about tied with the best-performing bit-budget costfunction (see fig. 7). However, the advantage of MPT is that it substitutes the computationally costly evaluation of a costfunction with the building of a unified similarity tree, which is cheap in terms of computational complexity.

In our testruns, CSC does not result in a major gain in PSNR-performance. On the contrary, the PSNR values are generally lower than without CSC. This may be due to the (necessary) enlarging of the object-shapes in the transform domain. But even though this problem could be solved by arranging the coefficients during the subsampling process, the possible gain in coding performance would hardly be worth the considerable overhead that is necessary to find the common structure.

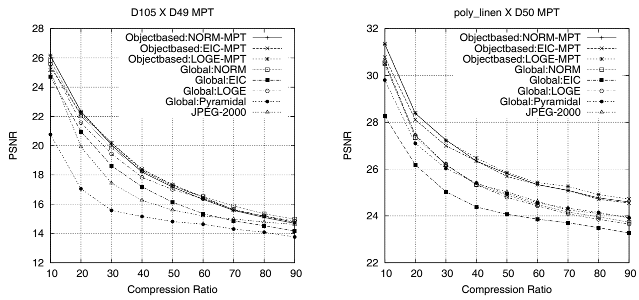
In fig. 8 we compare the performance of MPT to the approach using *Ganesh*++ with global optimization and to JPEG-2000. As expected, for all of the tested images, *Ganesh*++ using pyramidal decomposition and JPEG-2000 are both outperformed by the adaptive methods. However, the difference between the globally adaptive approach and the objectbased adaptive approach are quite insubstantial for most of the images (like in the case of fig. 8.a).

A possible explanation is the fact that for these images the



(a) BBCF: Results for fig. 6.a (b) BBCF: Results for fig. 6.b

Fig. 7. BBCF: Results for fig. 6



(a) MPT: Results for fig. 6.a (b) MPT: Results for fig. 6.b

Fig. 8. MPT: Results for fig. 6

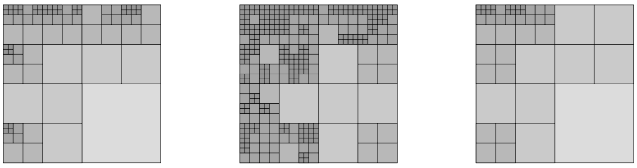
globally adaptive approach, due to the local preservation property of the wavelet transform, implicitly performs a segmentation in image space based on characteristics in frequency space. For our testimages, which are constructed from objects which exhibit these different characteristics, the coefficients for each object are almost disjointly divided into different frequency bands. This can best be illustrated by applying the algorithm to the artificial testimage shown in fig. 9.a, whose parts have high frequential patterns in orthogonal directions. Fig. 9.b shows the coefficient distribution over the subbands for the pyramidal decomposition[1]. As can be seen, the two objects are completely separated into different areas, and it is obvious that a segmentation based on frequency characteristics

[1]The pyramidal decomposition obviously does not achieve high energy compaction, but is well suited to illustrate the point.

(a) *Artificial test image* (b) *Distribution of coeff.*

Fig. 9. Distribution of coefficients in global optimization

prior to transformation will not yield much of an improvement.



(a) Global (b) Local - O_1 (c) Local - O_2

Fig. 10. Decomposition structures for fig. 6.b

However, global adaptivity does not outperform objectbased adaptivity for all testimages. In fig. 8.b, for example, the global approach is outperformed by more than 1db for most compression ratios. The global optimization in this case is not able to compact the energy as much as the objectbased approach. Furthermore, local optimization produced a much more complex decomposition for the foreground object O_1 (fig. 10.b) than is contained in the most successful decomposition structure in global optimization.

6. CONCLUSION

We can state that there are some special cases where the gain in PSNR-performance achieved by introducing objectbased adaptivity is substantial. For most images, the gain in compression efficiency is negligible or not present at all. However, considering the in crease in terms of functionality, the adaptive objectbased approach makes sense as long as compression performance does not degrade.

7. REFERENCES

[1] D. Taubman and M. Marcellin, *JPEG2000 — Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.

[2] I. Sodagar, H. J. Lee, P. Hatrack, and Y.-Q. Zhang, “Scalable wavelet coding for synthetic/natural hybrid coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 2, pp. 244–254, 1999.

[3] Z. Xiong, K. Ramchandran, M. T. Orchard, and Y.-Q. Zhang, “A comparative study of DCT- and wavelet-based image coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, pp. 692–695, Aug. 1999.

[4] M. Wickerhauser, *Adapted wavelet analysis from theory to software*. Wellesley, Mass.: A.K. Peters, 1994.

[5] D. Marpe, H. Cycon, and W. Li, “Complexity constrained best-basis wavelet packet algorithm for image compression,” *IEE Proceedings Vision, Image, and Signal Processing*, vol. 145, no. 6, pp. 391–398, 1998.

[6] Z. Xiong, K. Ramchandran, and M. T. Orchard, “Wavelet packet image coding using space-frequency quantization,” *IEEE Transactions on Image Processing*, vol. 7, pp. 892–898, June 1998.

[7] F. G. Meyer, A. Z. Averbuch, and J.-O. Strömberg, “Fast adaptive wavelet packet image compression,” *IEEE Trans. on Image Process.*, vol. 9, pp. 792–800, May 2000.

[8] T. Hopper, “Compression of gray-scale fingerprint images,” in *Wavelet Applications* (H. Szu, ed.), vol. 2242 of *SPIE Proceedings*, pp. 180–187, 1994.

[9] S. Li and W. Li, “Shape-adaptive discrete wavelet transform for arbitrarily shaped visual object coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 725–743, Aug. 2000.

[10] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. on Signal Process.*, vol. 41, pp. 3445–3462, Dec. 1993.

[11] N. M. Rajpoot, F. G. Meyer, R. G. Wilson, and R. R. Coifman, “On zerotree quantization for embedded wavelet packet image coding,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP'99)*, (Kobe, Japan), Oct. 1999.

[12] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–249, June 1996.

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