# Methods for the Anisotropic Wavelet Packet Transform

Rade Kutil and Dominik Engel
Department of Computer Sciences
University of Salzburg
Jakob-Haringer-Str. 2, A–5020 Salzburg
{rkutil, dengel}@cosy.sbg.ac.at
Tel.: +43 662 8044/6303, /6347, Fax: /172

**Abstract**

Anisotropic wavelet packets present a flexible transform with interesting properties and applications. While certain aspects of this transform have been investigated in conjunction with applications, this paper aims at providing a basic theoretical framework for working with anisotropic wavelet packets. Random decompositions are developed which have distributions with different average decomposition depths and degrees of anisotropy. They can be used in cryptographic applications or to test other algorithms. For the uniform distribution, it is necessary to determine the number of possible bases for all decomposition depths. A best basis algorithm for anisotropic decompositions is developed. A graph theoretical representation of the anisotropic decomposition structure is presented, which is unique for each decomposition and, thus, free of redundancy, which is important for compression purposes. A compression algorithm based on these techniques is developed and tested on random decompositions.

*Keywords: wavelet packets, anisotropic, random, best basis, graph, compression*

This is a revised preprint. The published article is:

R. Kutil and D. Engel, "Methods for the anisotropic wavelet packet transform," *Applied and Computational Harmonic Analysis*, 25(3):295–314, Nov. 2008, doi:10.1016/j.acha.2007.12.001.

# Contents

# 1 Introduction

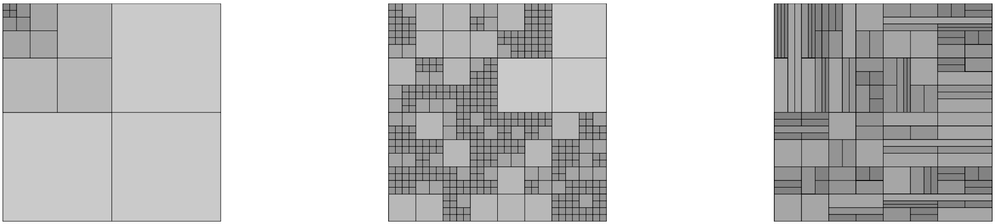
The wavelet transform [1, 2] has successfully been employed for signal/image compression [3], feature extraction [4], classification algorithms [5, 6], telecommunication applications [7], numerical mathematics [8], and many other fields. The wavelet transform consists of decomposition steps in which a pair of orthogonal quadrature mirror filters (QMF) or biorthogonal filters are applied on the data, producing one low-pass- and one high-pass-subband which are subsampled by a factor of 2. For higher dimensional data, the filtering is applied in each dimension to produce $2^d$ subbands, where $d$ is the number of dimensions. In this work we concentrate on two-dimensional data ($d = 2$), although most results can be extended to higher dimensions easily. In the classical pyramidal wavelet transform, the decomposition step is recursively applied only to the approximation subband, i.e. the subband that was low-pass filtered in each dimension. The other subbands are called detail subbands and are included in the output of the transform.

Isotropic and anisotropic wavelet packets are a generalization of the pyramidal wavelet transform. Examples are given in Figure 1. The wavelet packet transform [9] presents an overcomplete library of bases suitable for energy compaction in the frequency domain for visual data. Other than in the case of the pyramidal wavelet transform, in the wavelet packet transform also the detail subbands are subject to further decomposition. The multitude of bases provided by the wavelet packet transform allows for adaption in the frequency domain. In the context of image compression, wavelet packets have an advantage for images with oscillatory patterns in image space that result in more energy in the highpass subbands. The best basis algorithm [9] performs adaptive optimization of the wavelet packet decomposition structure to suit a source signal. Other methods use fixed decomposition structures to compact the energy of a class of signals, e.g. fingerprints [10].

The anisotropic wavelet packet transform [11] is a generalization of the isotropic case: whereas in the latter, horizontal and vertical wavelet decomposition are always applied in pairs for each subband to be decomposed, this restriction is lifted for anisotropic wavelet packets. Therefore, anisotropic wavelet packets offer an even more flexible adaption in the frequency domain. Anisotropic wavelet packets have been proposed for the compression of image [12, 13] and video [14] data. A rate-distortion optimized basis selection for 3-D anisotropic wavelet packets has been proposed for the compression of hyperspectral image [15]. Randomized selection of wavelet packet bases has been used to provide lightweight security for visual data, for isotropic [16, 17] as well as for anisotropic wavelet packets [18, 19].

This work aims at providing a basic theoretical framework for working with anisotropic wavelet packets: random and optimal generation, efficient representation of anisotropic bases, and analysis of some basic properties. Work published so far in this area mainly focused on image data and especially the compression thereof. Here we take a more holistic perspective. First, we investigate distributions for randomly selected anisotropic wavelet packet bases. Each of the discussed distributions has a different focus in selection. This can be useful for empirically investigating features of the anisotropic wavelet packet transform, compression performance being only one of them. Random decompositions are also very important for cryptographic purposes.

The number of expected subbands in a randomly selected basis will be determined for each distribution and also for random isotropic bases. This gives a measure for the



(a) Pyramidal Wavelet Decomposition

(b) Example Isotropic Wavelet Packet Decomposition

(c) Example Anisotropic Wavelet Packet Decomposition

Figure 1: Wavelet Decomposition Structure

average depth of a decomposition. Also, the degree of anisotropy for different distributions is a property that we will investigate in detail. This measures, how much an average decomposition will deviate from the isotropic case.

A second major topic of this work is the efficient representation of anisotropic wavelet packet decomposition structures. The representation as a decomposition tree that is commonly used for isotropic wavelet packets is not suitable for anisotropic wavelet packets. We discuss a suitable graph representation, the “bush” structure, that is unique for each basis. The uniqueness of the representation removes redundancy and is, therefore, a necessary condition for optimal coding. To evaluate the coding gain this representation warrants, we compare it to a straightforward representation which does not exhibit the property of uniqueness. A suitable method of representation of anisotropic wavelet packet decompositions is important for many applications, especially if they are situated in the area of data compression. The advantage of an anisotropic basis, tailored to a target image, for example by the use of the best basis algorithm presented in Section 5, would be small if the coding gain was eaten up by a redundant representation of the wavelet packet basis. The same is true for cryptographic applications. In Section 2 we discuss the large key-space that can be generated with anisotropic wavelet packets. For an encryption scheme that uses randomized anisotropic wavelet packets to work in practice, an explicit (encrypted) representation of the used basis in the metadata is of advantage: the decoder does not have to run through the process of random generation (which it would have to do if only the parameters for the generation process were transmitted), but can directly use the transmitted representation of the basis. It is obvious that such a representation should be as efficient as possible.

This paper is organized as follows: In the next section we investigate the total number of anisotropic wavelet packets for a certain maximum decomposition depth. In Section 3 we discuss the construction of random decompositions. We turn to the suitable representation of anisotropic wavelet packet structures in Section 4. Section 5 presents a best-basis algorithm for anisotropic wavelet packets. Efficient coding of the graph structure is discussed in Section 6 and results are presented that compare the coding gain achieved by the sophisticated representation to a straightforward representation. Section 7 concludes.

# 2 Number of Anisotropic Bases

The discussion of the number of anisotropic bases forms the foundation for the distributions which we construct later. We investigate the number of anisotropic wavelet packets for two cases: *joint* and *separate*. For the first, there is a joint maximum decomposition depth $j$ for horizontal and vertical decomposition, meaning that the number of horizontal plus vertical decomposition steps must not exceed $j$. For the second, there is a maximum decomposition depth for each dimension: $j$ for vertical decomposition and $k$ for horizontal decomposition.

## 2.1 Joint Decomposition Depth

We determine $A_j$, the number of bases of joint horizontal and vertical decomposition level up to $j$, recursively by adapting the method put forward by Xu and Do [13]. The root node may not be decomposed, or it may be decomposed either horizontally or vertically, forming two subtrees of $A_{j-1}$ possible decompositions in each case, leading to $1 + 2A_{j-1}^2$ possible bases. There exist, however, some decompositions that result in the same basis: a horizontal decomposition (**r**) followed by two vertical decompositions (**c**) on the resulting subtrees is equivalent to the case in which the vertical decomposition is done first followed by two horizontal decompositions. The total number of these decompositions is equal to $2A_{j-2}^4$, as illustrated in Figure 2. As the resulting bases should be counted only once, half the number is subtracted, leading to the formula:

$$A_j = 1 + 2 \cdot A_{j-1}^2 - A_{j-2}^4 \tag{1}$$

where $A_0 = 1$, $A_1 = 3$, and $A_j = 0$ for $j < 0$.

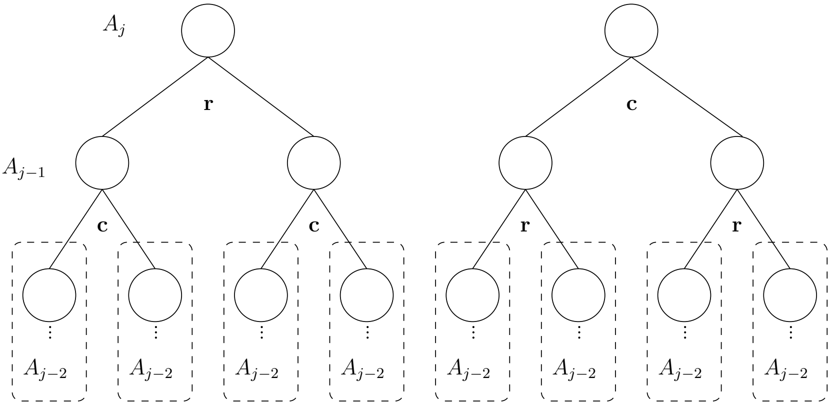


Figure 2: Number of equivalent AWP bases

## 2.2 Separate Decomposition Depth

Separate decomposition depths have been investigated by Xu and Do [13]. They give $A_{j,k}$, the number of bases with horizontal decomposition level up to $j$ and vertical decomposition level up to $k$, as follows

$$A_{j,k} = 1 + A_{j-1,k}^2 + A_{j,k-1}^2 - A_{j-1,k-1}^4 \tag{2}$$

with $A_{j,0} = 1 + A_{j-1,0}^2$, $A_{0,k} = 1 + A_{0,k-1}^2$ and $A_{0,0} = 1$. Furthermore, $A_{j,k} = 0$ for $j < 0 \vee k < 0$.

# 3 Random Decomposition

In this section we suggest some representative random decompositions and investigate some of their properties such as expected number of subbands and the distribution of their subbands' degree of anisotropy.

Two types of random decompositions will be presented. For the first type, each possible decomposition, constrained by maximum decomposition depth, has the same probability. The second type assigns a higher probability to smaller decomposition depths in order to produce approximate scale invariant decompositions.

Each of the types of decompositions comes in three forms. First, we consider the isotropic case. Second, there is only a joint maximum decomposition depth $j$ which the total number of horizontal plus vertical decomposition steps must not exceed. Third, there are separate maximum decomposition depths $j$ and $k$ for horizontal and vertical decompositions.

If the original data is not decomposed, this corresponds to a decomposition of depth 0, i.e. a decomposition consisting of a single subband. On the other hand, a full decomposition of depth $j$ (or depths $j$ and $k$) consists of a maximum of $2^j$ (or $2^{j+k}$) subbands. In the isotropic case the number would be $4^j$. An average decomposition lies somewhere in between and the expected number of subbands $N_j$ (or $N_{j,k}$) gives a measure of how deep a random decomposition is, depending on the distribution of the decompositions.

An isotropic subband is produced by an equal number of horizontal and vertical decomposition steps. We say that an isotropic subband has a zero degree of anisotropy. Following this idea, a subband produced by $j$ horizontal and $k$ vertical decomposition steps has a degree of anisotropy $a = j - k$. Thus, the anisotropy is positive if there are more horizontal decomposition steps and negative if there are more vertical. Each distribution owns a certain distribution of anisotropy, i.e. for a random coefficient of a random decomposition there is a certain probability $P_j^a$ (or $P_{j,k}^a$) that it has a degree $a$ of anisotropy.

To determine the overall degree of anisotropy of a distribution, the second non-central moment is used

$$\bar{\bar{a}}^2 = \sum_a P_j^a a^2 \,, \tag{3}$$

which represents the mean squared deviation from $a = 0$. This measure can be estimated from a single decomposition $D$ by averaging the squared anisotropy of all (non-decomposed) subbands $S \in D$, weighted by the size of the subbands:

$$\bar{\bar{a}}^2(D) = \sum_{S \in D} 2^{-j(S)-k(S)} (j(S) - k(S))^2 \,, \tag{4}$$

where $j(S)$ and $k(S)$ are the horizontal and vertical decomposition levels of the subband $S$.

The remainder of this section will develop recurrence equations for the expected number of subbands and for the distribution of anisotropy and investigate their convergence.

### 3.1 Uniform Distribution of Isotropic Decompositions

For the purpose of comparison we will first have a look at isotropic decompositions. An isotropic decomposition of depth $j$ consists of $j$ decomposition steps. Each step in turn consists of a horizontal and a vertical filtering step. In each step, four subbands are created. The total number of possible decompositions is

$$\tilde{A}_j = 1 + \tilde{A}_{j-1}^4 \,. \tag{5}$$

Of course, $\tilde{A}_0 = 1$. To produce a random decomposition in a way that each possible decomposition has the same probability, each subband must have a probability of $\frac{1}{\tilde{A}_j}$ that it is not decomposed, where $j$ is the maximum decomposition depth of that subband. A subband at the next level has $j$ reduced by 1.

**Expected Number of Subbands** Let $\tilde{N}_j$ be the expected number of subbands in an isotropic decomposition, assuming a uniform distribution of decompositions. $\tilde{N}_j$ can be split into the case of no decomposition, where we have only one subband, and the case of a decomposition depth of at least one, where there are four subbands which are on average decomposed into $\tilde{N}_{j-1}$ subbands. The first case has a probability of $\frac{1}{\tilde{A}_j}$. Thus, we get

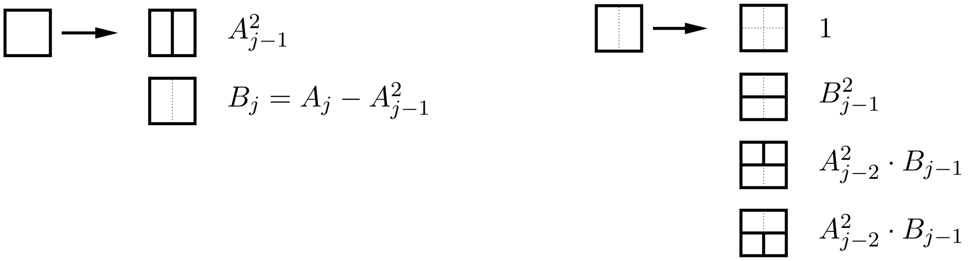
$$\tilde{N}_j = \frac{1}{\tilde{A}_j}(1 + \tilde{A}_{j-1}^4 4\tilde{N}_{j-1}) \,. \tag{6}$$

The maximum number of subbands is $2^{2j}$. To see how deep an average decomposition becomes compared to a full decomposition, we build the quotient and see that it converges quickly, so that for $j \geq 4$ we have

$$\frac{\tilde{N}_j}{2^{2j}} \approx 0.036994 \,. \tag{7}$$

### 3.2 Uniform Distribution with Joint Decomposition Depth

For a uniform distribution of all possible anisotropic bases up to a maximum decomposition level $j$, we split the decomposition decisions into mutually exclusive cases and assign probabilities based on the number of subtrees for each case. We introduce restrictions in decomposition to handle isotropic bases. Without loss of generality, we define the admissible decompositions for a subband without restriction as (a) horizontal decomposition with further processing, or (b) further processing with restriction for horizontal decomposition. Case (b) leads to the restricted case in which horizontal decomposition is forbidden for at least one of the resulting subbands. The number of the possible bases in case (b) is $A_{j-1} - A_{j-2}^2$, which we define as $B_j$. Figure 3 shows the possible decomposition decisions and lists the number of bases for each case. The probability for any case to be chosen is the ratio of the number of bases contained in the case to the total number of bases ($A_j$ for unrestricted processing, $B_j$ for restricted processing).



(a) Unrestricted Processing ($A_j$)

(b) Restricted Processing ($B_j$)

Figure 3: Case selection for uniform distribution of randomized anisotropic wavelet packet bases for joint maximum horizontal and vertical decomposition depth

Note that this distinction into mutually exclusive cases also presents a slightly different way to determine the number of available bases without using the definition of $A_j$ or $B_j$ as given above. This makes the resulting recursion more flexible, which is useful if additional restrictions are applied in selecting a suitable anisotropic bases (such as restriction that enhance compression performance, see [18]). The alternative definitions are given by:

$$A_j = \begin{cases} B_0 & \text{for } j = 0 \\ A^2_{j-1} + B_j & \text{else} \end{cases} \tag{8}$$

$$B_j = B_j^{(1)} + B_j^{(2)} + B_j^{(3)} + B_j^{(4)} \tag{9}$$

$$B_j^{(1)} = 1 \tag{10}$$

$$B_j^{(2)} = \begin{cases} 0 & \text{for } j = 0 \\ B^2_{j-1} & \text{else} \end{cases} \tag{11}$$

$$B_j^{(3)} = B_j^{(4)} = \begin{cases} 0 & \text{for } j = 0 \vee j = 1 \\ A^2_{j-2} \cdot B_{j-1} & \text{else.} \end{cases} \tag{12}$$

The numbers $A_j$ and $B_j$ become very large very quickly. More precisely, they grow as $K\beta^{2^j}$. Numerical experiments show that $\beta \approx 2.353017462$. The following probabilities have to be calculated while performing a random decomposition for each decomposition level $j$:

$$p_j^{(A1)} = \frac{A^2_{j-1}}{A_j}, \quad p_j^{(A2)} = \frac{B_j}{A_j}, \quad p_j^{(B1)} = \frac{1}{B_j}, \quad p_j^{(B2)} = \frac{B^2_{j-1}}{B_j}, \quad p_j^{(B3)} = \frac{A^2_{j-2}B_{j-1}}{B_j} \tag{13}$$

Since $A_j$ and $B_j$ might overflow the number presentation even for moderate depths $j$, alternative ways to calculate the probabilities $p_j^{(\cdot\cdot)}$ are necessary. There are several possibilities. The first one is very simple. The probabilities converge very quickly ($p^{(A1)} \rightarrow 0.618034, p^{(B1)} \rightarrow 0, p^{(B2)} \rightarrow 0.236068$). Therefore, only a minor error is introduced by substituting the limit for the probabilities, starting at a high enough depth $j$.

The second possibility is slightly more elaborate. Instead of $A_j$ and $B_j$ the values $A'_j = \beta^{-2^j} A_j$ and $B'_j = \beta^{-2^j} B_j$ are calculated. This is very easily done by adding $\beta^{-2^j}$ instead of 1 for the non-decomposition case in (1) or (10). All other equations remain the same, as can easily be checked. This possibility has the disadvantage that small errors in the representation of $\beta$ make the recursion diverge to $\infty$ or 0 for high enough $j$.

The third possibility is the most complicated. A whole new set of recurrence equations for the probabilities $p_j^{(\cdot\cdot)}$ are developed. Assuming that $p_j^{(B1)}$ is neglectably small above a certain depth $j_0$, the following equations hold:

$$p_j^{(T1)} = 2p_{j-1}^{(A1)} + p_{j-1}^{(A2)}, \quad p_j^{(T2)} = p_j^{(T1)} p_{j-1}^{(A2)}, \tag{14}$$

$$p_j^{(A1)} = \frac{1}{1+p_j^{(T2)}}, \quad p_j^{(A1)} = \frac{p_j^{(T2)}}{1+p_j^{(T2)}}, \quad p_j^{(B2)} = \frac{p_{j-1}^{(A2)}}{p_j^{(T1)}}, \quad p_j^{(B2)} = \frac{p_{j-1}^{(A1)}}{p_j^{(T1)}} \tag{15}$$

The starting values are set by calculating $p_j^{(\cdot\cdot)}$ from $A_j$ and $B_j$ for $j \leq j_0$.

**Expected Number of Subbands** Let $N_j$ be the expected number of subbands in an anisotropic decomposition, assuming a uniform distribution of decompositions. The subband we start with can either be decomposed horizontally, vertically or not at all. The latter case gives a number of subbands of 1 at a probability of $\frac{1}{A_j}$. The other cases give $2N_{j-1}$ subbands at a probability of $\frac{A_{j-1}^2}{A_j}$. However, some decompositions are counted twice, so they have to be subtracted once. We obtain

$$N_j = \frac{1}{A_j}(1 + 2A_{j-1}^2 2N_{j-1} - A_{j-2}^2 4N_{j-2}), \tag{16}$$

The maximum number of subbands is $2^j$ and for $j \geq 5$ we have

$$\frac{N_j}{2^j} \approx 0.7828\,. \tag{17}$$

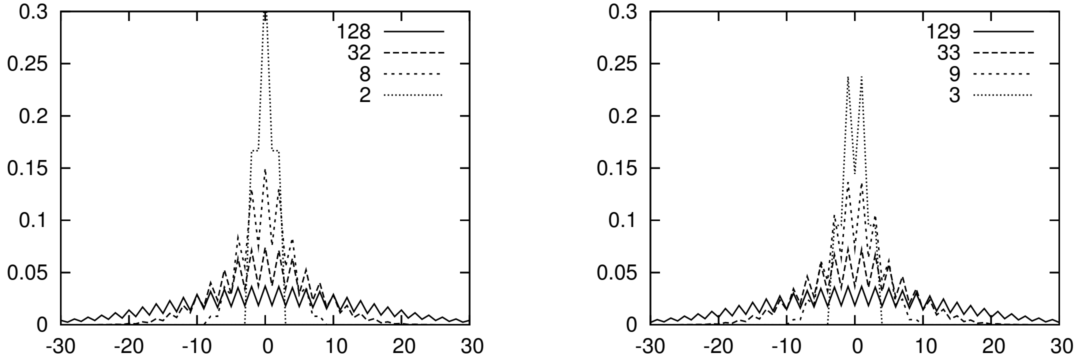
This means that anisotropic decompositions have on average about 21 times as many subbands as isotropic decompositions.

**Degree of Anisotropy** The distribution of anisotropy is expressed by probabilities $P_j^a$ that a random coefficient of a random decomposition with maximum depth $j$ lies in a subband with a degree $a$ of anisotropy. Again, a recurrence equation can be found for $P_j^a$ by considering cases similar to the computation of $N_j$. If the original subband is not decomposed the anisotropy is 0, which means that $P_j^a = \frac{1}{A_j}$ if $a = 0$ and $P_j^a = 0$ otherwise. In case of horizontal or vertical decomposition, the probabilities $P_{j-1}^{a-1}$ and $P_{j-1}^{a+1}$ are inherited. By correcting decompositions that are counted twice, we obtain

$$P_j^a = \frac{1}{A_j}(\chi(a=0) + A_{j-1}^2(P_{j-1}^{a-1} + P_{j-1}^{a+1}) - A_{j-2}^4 P_{j-2}^a), \tag{18}$$

where the characteristic function $\chi(e) = 1$ if $e$ is true and $\chi(e) = 0$ otherwise. The starting values are simply $N_j^a = 0$ for $j \leq 0$ except for the case $N_0^0 = 1$. Note that of course $\sum_{a=-j}^{j} P_j^a = 1$.

Figure 4 shows the distribution of anisotropy for several maximum decomposition depths $j$. As expected, subbands with a high degree of anisotropy are less probable than the more isotropic. The most surprising fact in Figure 4 is that there is a significant contrast between odd and even values of $j$ and $a$. If the maximum decomposition depth $j$ is even, then even values $a$ are more probable than odd values and vice versa.



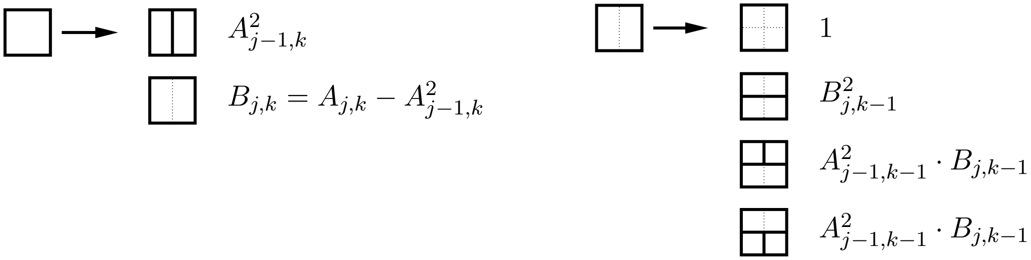
(a) even maximum decomposition depth $j$

(b) odd maximum decomposition depth $j$

Figure 4: Distribution of anisotropy for uniform distribution with joint maximum decomposition depth.

### 3.3 Uniform Decomposition with Separate Decomposition Depths

The procedure can be easily extended to take separate maximum decomposition depths for horizontal and vertical decomposition into account. Figure 5 lists the corresponding cases and the number of bases contained in each.


$A_{j-1,k}^2$
$B_{j,k} = A_{j,k} - A_{j-1,k}^2$
1
$B_{j,k-1}^2$
$A_{j-1,k-1}^2 \cdot B_{j,k-1}$
$A_{j-1,k-1}^2 \cdot B_{j,k-1}$

(a) Unrestricted Processing

(b) Restricted Processing

Figure 5: Case selection for uniform distribution of randomized anisotropic wavelet packet bases for separate maximum horizontal and vertical decomposition depth

Again, the numbers $A_{j,k}$ and $B_{j,k}$ grow very quickly as $K\beta^{2^{j+k}}$, where this time $\beta \approx 1.7381342497927$. The following probabilities have to be calculated for all decomposition levels $j$ and $k$ while performing a random decomposition:

$$p_{j,k}^{(A1)} = \frac{A_{j-1,k}^2}{A_{j,k}}, \quad p_{j,k}^{(A2)} = \frac{B_{j,k}}{A_{j,k}},$$

$$p_{j,k}^{(B1)} = \frac{1}{B_{j,k}}, \quad p_{j,k}^{(B2)} = \frac{B_{j,k-1}^2}{B_{j,k}}, \quad p_{j,k}^{(B3)} = \frac{A_{j-1,k-1}^2 B_{j,k-1}}{B_{j,k}} \qquad (19)$$

As we try to apply the same three possibilities to calculate these probabilities without having to calculate $A_{j,k}$ and $B_{j,k}$ as in Section 3.2, we find that the first one – substitution

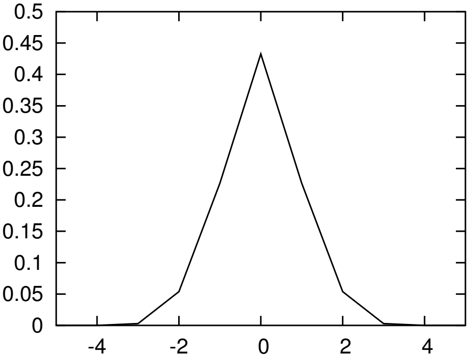


Figure 6: Distribution of anisotropy for uniform distribution with separate maximum decomposition depths.

of limits – does not work here because the limits depend on the quotient of $j$ and $k$. The second possibility – normalization by $\beta^{-2^{j+k}}$ – does work. It owns the same problems of instability, though. The third possibility turns out to be slightly more complicated than in the case of joint decomposition depths. The following equations hold:

$$p^{(T1)}_{j,k} = \left(\frac{p^{(A1)^2}_{j,k-1}}{p^{(T1)^2}_{j-1,k}} + 2p^{(A2)}_{j,k-1}p^{(A1)}_{j,k-1} + p^{(A2)^2}_{j,k-1}\right)^{-1}, \quad p^{(T2)}_{j,k} = p^{(A2)}_{j,k-1}(2-p^{(A2)}_{j,k-1})p^{(T1)^2}_{j-1,k}, \tag{20}$$

$$p^{(A1)}_{j,k} = \frac{p^{(A2)^2}_{j,k-1}}{p^{(T2)}_{j,k} + p^{(A2)^2}_{j,k-1}}, \quad p^{(A2)}_{j,k} = 1 - p^{(A1)}_{j,k}, \quad p^{(B2)}_{j,k} = \frac{p^{(A1)}_{j,k-1}}{1+p^{(A1)}_{j,k-1}}, \quad p^{(B3)}_{j,k} = \frac{p^{(A2)}_{j,k-1}}{1+p^{(A1)}_{j,k-1}} \tag{21}$$

**Expected Number of Subbands** Let $N_{j,k}$ be the expected number of subbands. We obtain

$$N_{j,k} = \frac{1}{A_{j,k}}(1 + A^2_{j-1,k}2N_{j-1,k} + A^2_{j,k-1}2N_{j,k-1} - A^4_{j-1,k-1}4N_{j-1,k-1})\,. \tag{22}$$

The maximum number of subbands is $2^{j+k}$ and for $j \geq 3, k \geq 3$ we have

$$\frac{N_{j,k}}{2^{j+k}} \approx 0.6064\,. \tag{23}$$

**Degree of Anisotropy** The distribution of anisotropy can be calculated analogously to the case of joint decomposition depths.

$$P^a_{j,k} = \frac{1}{A_j}(\chi(a=0) + A^2_{j-1,k}P^{a-1}_{j-1,k} + A^2_{j,k-1}P^{a+1}_{j,k-1} - A^4_{j-1,k-1}P^a_{j-1,k-1})\,, \tag{24}$$

These probabilities converge quickly and for $j = k \geq 3$ we get a distribution as in Figure 6.

### 3.4 Scale-Invariant Distribution with Joint Decomposition Depth

Because average decompositions are so deep in the uniform case, we may want to have a distribution of decompositions where bigger subbands are more likely to occur. A first approach would be to apply a fixed probability $p$ of non-decomposition on every subband. Such a distribution is scale-invariant (self-similar) in that each subband is statistically equal in terms of decomposition structure, except of course for the maximum decomposition depth, which is smaller for smaller subbands. The expected number of subbands $N_j$ is

$$N_j = p + (1-p)2N_{j-1}\,. \tag{25}$$

The convergence of this recurrence equation shows the following behavior.

$$\begin{aligned} 0 \leq p < \frac{1}{2} \quad &\Rightarrow \quad \lim_{j\to\infty} N_j(2(1-p))^{-j} = \frac{1-p}{1-2p} \\ p = \frac{1}{2} \quad &\Rightarrow \quad N_j = 1 + \frac{j}{2} \\ \frac{1}{2} < p \leq 1 \quad &\Rightarrow \quad \lim_{j\to\infty} N_j = \frac{p}{2p-1} \end{aligned}$$

The interesting case is $0 \leq p < \frac{1}{2}$ where the number of subbands grows exponentially with the maximum decomposition depth, but with a base $\alpha = 2(1-p)$ in the range $1 < \alpha \leq 2$. The case $\alpha = 2$ ($p = 0$) is a full decomposition. In the case $\alpha \to 1$ ($p \to \frac{1}{2}$) the behavior should approach a constant, but the convergence becomes very slow.

A better approach is therefore to demand that

$$N_j = \alpha^j \qquad 1 \leq \alpha \leq 2\,, \tag{26}$$

and to choose the probability of non-decomposition $p_j$ so that this is fulfilled. Note that this leads to a distribution that is not exactly scale-invariant any more. Examining the quotient

$$\frac{N_j}{N_{j-1}} = \alpha = \frac{p_j + (1-p_j)2N_{j-1}}{N_{j-1}} = p_j\frac{1-2N_{j-1}}{N_{j-1}} + 2 \tag{27}$$

enables us to find such a $p_j$ as

$$p_j = \frac{\alpha - 2}{N_{j-1}^{-1} - 2} = \frac{2-\alpha}{2-\alpha^{1-j}}\,. \tag{28}$$

Note that for $\alpha = 2$ we get a full decomposition, since $p_j = 0$, and for $\alpha = 1$ there will be no decomposition at all, since $p_j = 1$.

**Degree of Anisotropy** The distribution of anisotropy can be calculated with a recurrence equation similar to the one for the uniform distribution.

$$P_j^a = p_j\chi(a=0) + \frac{1-p_j}{2}(P_{j-1}^{a-1} + P_{j-1}^{a+1}) \tag{29}$$

These probabilities converge to the distribution in Figure 7.

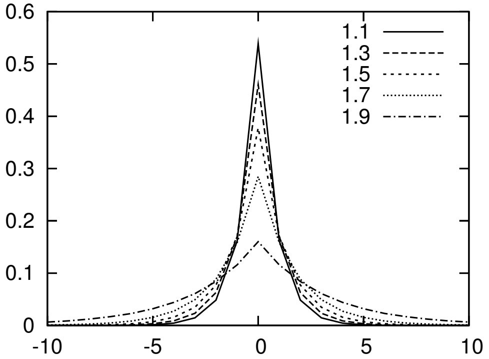


Figure 7: Distribution of anisotropy for scale-invariant decompositions for several values of $\alpha$.

### 3.5 Scale-Invariant Distribution with Separate Decomposition Depths

The case of separate maximum decomposition depths for horizontal and vertical dimensions is handled easily by demanding that $N_{j,k} = \alpha^{j+k}$. The probability of non-decomposition is simply $p_{j,k} = p_{j+k}$. The probabilities $q_{j,k}$ and $r_{j,k}$ of horizontal and vertical decomposition can be chosen so that $p_{j,k} + q_{j,k} + r_{j,k} = 1$ is fulfilled, as well as $q_{0,k} = 0$ and $r_{j,0} = 0$. It is reasonable to let $q_{j,j} = r_{j,j}$. The most natural way to ensure all this is to set

$$q_{j,k} = (1 - p_{j,k})\frac{j}{j+k}, \quad r_{j,k} = (1 - p_{j,k})\frac{k}{j+k}\,. \tag{30}$$

Another possibility would be to simply set $q_{j,k} = r_{j,k} = \frac{1}{2}(1 - p_{j,k})$ for $j > 0$ and $k > 0$. However, this leads to rather unnatural distributions of anisotropy that heap up at the "borders" $\pm j$ or $\pm k$.

**Degree of Anisotropy** The distribution of anisotropy is calculated in this way:

$$P^a_{j,k} = p_{j,k}\chi(a = 0) + q_{j,k}P^{a-1}_{j-1,k} + r_{j,k}P^{a+1}_{j,k-1} \tag{31}$$

These probabilities converge to exactly the same distribution as in the case of joint decomposition depths, see Figure 7. However, for $\alpha$ near 2, the convergence is slow. The reason is that, in such a case, the decomposition is almost total and the anisotropy is equal to $j - k$ (isotropy if $j = k$) with high probability. Only for high decomposition depths, significant amounts of subbands with lower decomposition levels and greater anisotropy can emerge.

### 3.6 Distribution with Additional Constraints

For some applications, random selection of a basis is desired that follows a distribution with additional constraints. For example, if the target data is visual data, a certain level of energy compaction may be desired. This has been employed for lightweight encryption of visual data (for isotropic [16] and anisotropic wavelet packets [18]): For this approach, the decomposition structure is selected randomly and kept secret. To ensure competitive compression performance, the set of admissible bases is restricted to those that achieve high energy compaction in the transform domain.

A number of parameters can be used to apply additional constraints. They are listed in Table 1.

| | |
|---|---|
| $n$ | Minimum decomposition depth of the approximation subband |
| $m$ | Maximum decomposition depth of the approximation subband |
| $e$ | Minimum decomposition depth of the detail subbands |
| $d$ | Maximum decomposition depth of the detail subbands |
| $q$ | Maximum degree of anisotropy for approximation subband |
| $r$ | Maximum degree of anisotropy for detail subbands |
| $s$ | Seed for pseudo-random number generator |

Table 1: Parameters for Additional Constraints

The first four parameters, $n, m, e, d$, determine the maximum and minimum decomposition depths for the approximation and the detail subbands. They influence both compression performance and number of admissible bases. Only the joint decomposition depths are listed in Table 1, separate decomposition depths for horizontal and vertical decomposition could also be used.

The maximum degree of anisotropy for approximation subband and detail subband, $q$ and $r$, respectively, can be used to prevent subbands from being decomposed into a single direction excessively, as this would lead to inferior energy compaction for natural images in the frequency domain for the other direction. For applications that require good compression performance whilst providing a large keyspace, it is sensible to define separate parameters for the maximum degree of anisotropy for the approximation and the detail subbands. The maximum degree of anisotropy has an impact on both compression performance and the number of admissible bases.

The number of available bases, expected number of subbands and the degree of anisotropy depends on the values used for the parameters. The degree of anisotropy is of course related to the setting of parameters $q$ and $r$, the number of subbands and number of bases is influenced by maximum and minimum decomposition depths ($n$, $m$, $e$, $d$). Generally, parameters set for the approximation subband ($n$, $m$, $q$), while possibly important for image compression, do not influence these properties too much. Restricting the degrees of freedom for the detail subbands (by setting parameters $r$, $e$, $d$), on the other hand, has a major influence on the number of bases.

The number of available bases for working with additional constraints can be determined by adapting the recurrence equation. As an example, we develop the formula for joint decomposition depths. Furthermore, we allow arbitrary degrees of anisotropy for the detail subbands, and only restrict the maximum degree of anisotropy for the approximation subband. Such a setup reflects the needs of a lightweight encryption scenario [19]. We use the previously given definition for the degree of anisotropy

$$Q(h, v) = v - h \tag{32}$$

where $h$ and $v$ are the decomposition depths in horizontal and vertical direction, respectively. Furthermore, to reflect the minimum decomposition depth $n$ we define $N$ as

$$N(j,a) = \begin{cases} 1 & \text{for } a = 0 \\ 0 & \text{for } a = 1 \wedge j < n \\ 1 & \text{for } a = 1 \wedge j \geq n \end{cases} \tag{33}$$

where $a$ defines if the current subband is in the approximation tree ($a = 1$) or in the detail tree ($a = 0$). We can then determine the number of bases taking additional constraints into consideration. Subscript $j$ refers to the total number of possible decompositions available. Subscripts $h$ and $v$ refer to the number of horizontal and vertical decompositions, respectively, that have been performed to produce the current subband. Finally, subscript $a$ determines if a subband is in the subtree of the approximation subband or if it is a detail subband.

$$A_{j,h,v,a} = \begin{cases} 1 & \text{for } h+v+1 > d \\ B_{0,h,v,a} & \text{for } j = 0 \\ B_{j,h,v,a} & \text{for } a = 1 \wedge \\ & |Q(h+1,v)| > q \\ A_{j-1,h+1,v,a} \cdot A_{j-1,h+1,v,0} & \\ +B_{j,h,v,a} & \text{else} \end{cases} \tag{34}$$

$$B_{j,h,v,a} = B_j^{(1)} + B_j^{(2)} + B_j^{(3)} + B_j^{(4)} \tag{35}$$

$$B^{(1)}_{j,h,v,a} = N(j,a) \tag{36}$$

$$B^{(2)}_{j,h,v,a} = \begin{cases} 0 & \text{for } j = 0 \\ 0 & \text{for } h+v+1 > d \\ 0 & \text{for } a = 1 \wedge \\ & |Q(h,v+1)| > q \\ B_{j-1,h,v+1,a} \cdot B_{j-1,h,v+1,0} & \text{else} \end{cases} \tag{37}$$

$$B^{(3)}_{j,h,v,a} = \begin{cases} 0 & \text{for } j = 0 \vee j = 1 \\ 0 & \text{for } h+v+1 > d \\ 0 & \text{for } a = 1 \wedge \\ & |Q(h,v+1)| > q \\ A_{j-2,h+1,v+1,a} \cdot A_{j-2,h+1,v+1,0} \cdot & \\ B_{j-1,h,v+1,0} & \text{else} \end{cases} \tag{38}$$

$$B^{(4)}_{j,h,v,a} = \begin{cases} 0 & \text{for } j = 0 \vee j = 1 \\ 0 & \text{for } h+v+2 > d \\ 0 & \text{for } a = 1 \wedge \\ & |Q(h,v+1)| > q \\ B_{j-1,h,v+1,a} \cdot A^2_{j-2,h+1,v+1,a} & \text{else.} \end{cases} \tag{39}$$

This formula reflects the minimum decomposition depth $n$ of the approximation subband by checking $N$. The maximum degree of anisotropy is handled by comparing $Q$ to the minimum degree of anisotropy $q$. The maximum decomposition of the detail subband is handled by checking against $e$.

The choice which additional constraints to set as well as the optimal settings of the additional constraints will depend on the target application. For the lightweight encryption scheme mentioned above, which seeks a balance between high energy compaction and a large number of admissible bases, suitable settings have been determined empirically [18]. As an illustration, Table 2 gives the number of bases for different settings of three additional constraints: $n$, $d$, and $q$. It can be seen that setting $d$ has a major impact on the number of available bases. Settings that pertain only to the approximation tree obviously have a much smaller impact on the number of bases. For the lightweight encryption scheme this is advantageous, as setting a sufficient minimum decomposition depth and a low maximum degree of anisotropy for the approximation subband produces favorable compression results, while still maintaining a number of bases that is sufficient for providing lightweight security. A detailed discussion of this point can be found in [19]. A lightweight encryption scheme that relies on anisotropic wavelet packet bases has the nice property of providing a generous keyspace, while at the same time only a small amount of data needs to be encrypted. There are two possibilities how to transmit the used basis: (a) send the parameters and the seed used for random generation in the header (encrypted with a traditional cipher) or (b) insert a description of the used basis into the header (encrypted with a traditional cipher). Option (b) has the important advantage of saving the decoder the need to run through the process of random generation. Instead the description can be used directly. Of course, the representation of the basis should be as efficient as possible. We propose the bush structure as a possible representation in the next section.

|  | Additional Constraints |  |  |  | No Constraints |
|---|---|---|---|---|---|
| $m$ | $n$ | $d$ | $q$ | #Bases | #Bases |
| 6 | 0 | 6 | 1 | $2^{75}$ | $2^{78}$ |
| 12 | 6 | 12 | 1 | $2^{5048}$ | $2^{5055}$ |
| 12 | 0 | 8 | 1 | $2^{364}$ | $2^{5055}$ |
| 12 | 6 | 8 | $\infty$ | $2^{371}$ | $2^{5055}$ |
| 12 | 6 | 8 | 1 | $2^{364}$ | $2^{5055}$ |

Table 2: Number of Bases with Additional Constraints

# 4 Graph Representation of Decomposition Structure

The lightweight encryption scheme discussed above is only one example for which an efficient representation of anisotropic wavelet packet decomposition structures is important. Another example is image and video compression. A possible gain in compression performance that can be achieved by using a basis that is tailored to the target visual data should not be cancelled out by a redundant representation of this basis.

Subbands of isotropic wavelet packet decompositions are created by a unique succession of filtering steps. Each step consists of one horizontal and one vertical application of a filter pair. While these two sub-steps may be interchanged, the filtering steps cannot. Therefore, the graph structure of a decomposition is a tree with subbands as nodes and filtering steps as edges, and a unique decomposition path from the root node, i.e. original data, to each leaf node, i.e. non-decomposed subband.

On the other hand, the filtering steps of an anisotropic decomposition may be interchanged to some degree. Therefore, there can be several paths between subbands, which means that the underlying graph structure is not a tree. Because of its interwoven nature, we will call this structure *bush*.

Bushes are introduced as general graphs by Kutil [20, 21]. Bushes as used in 2-D anisotropic wavelet packet decompositions are special cases, i.e. 2-D binary bushes. Such graphs can more easily be introduced as subgraphs of the Cartesian product of two binary trees.

Let $T_1$ and $T_2$ be two binary trees, where $T_i = (V_i, E_i)$. $V_i$ is the set of vertices and $E_i \subseteq V_i \times V_i$ the set of edges. The Cartesian product of these trees is defined as

$$\begin{aligned} T_1 \times T_2 := (V, E) \quad \text{where} \quad V := V_1 \times V_2, \quad \text{and} \\ E := \{(a, b, c) \,|\, a, b \in V, (a_c, b_c) \in E_c, a_{\bar{c}} = b_{\bar{c}}\}\,, \end{aligned} \tag{40}$$

where $\bar{c} = 2$ if $c = 1$ and $\bar{c} = 1$ if $c = 2$. This means that the tree product contains the Cartesian product of the trees' vertices as its set of vertices, and each two vertices are adjacent if the corresponding vertices of one tree are adjacent and the ones of the other are equal. Each edge is additionally associated with a "color" (or dimension) $c$ that specifies in which tree the corresponding vertices are unequal and thus adjacent. In other words, a position in the tree product is determined by a position in each of the two trees, and one can move to another position only by moving the position in one tree and remaining stationary in the other.

As with (rooted) trees we can now introduce parent-child relationships in a (rooted) bush. The root vertex $r$ of a bush is of course the vertex that is the pair of the two trees' roots. Parents and children of vertices, however, are not unique and depend on the color (or dimension).

$$\text{child}(a, c, k) = b \quad :\Leftrightarrow \quad a \overset{c}{\sim} b \wedge \text{child}(a_c, k) = b_c \tag{41}$$

$$\text{parent}(b, c) = a \quad :\Leftrightarrow \quad a \overset{c}{\sim} b \wedge \text{parent}(b_c) = a_c \tag{42}$$

Note that $a \overset{c}{\sim} b$ is a short notation for $(a, b, c) \in E$. child$(a, k)$ and parent$(b)$ denote the usual tree-operators for the $k$-th child and the parent.

A 2-D binary bush is a subgraph $B \subseteq T_1 \times T_2$ of the Cartesian product of two binary trees $T_1$ and $T_2$ which contains all possible paths from the root to each vertex. Under certain assumptions, this makes the bush structure unique for a given set of leaf vertices, i.e. the desired decomposition, which is important for efficient coding, as we will see later. However, this condition is a *global* condition and therefore difficult to implement. Fortunately, it can also be expressed in the following *local* conditions, which axiomatize a 2-D binary bush.

1. $\forall b \neq r \exists a, c : a = \text{parent}(b, c)$ (43)
2. if $\exists b, b', c \neq c' : \text{child}(a, c, k) = b \wedge \text{child}(a, c', k') = b'$
   then $\exists d : \text{child}(b, c', k') = \text{child}(b', c, k) = d$ (44)
3. for all colors $c \neq c'$ and $m = 1, 2$ :
   if $\forall k = 1, 2 : \exists\, \text{child}(\text{child}(a, c, k), c', m)$ then $\exists\, \text{child}(a, c', m)$ (45)

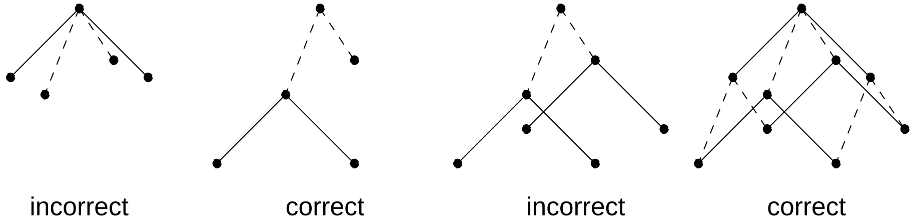


Figure 8: Correct and incorrect 2-D binary bushes.

The first condition assures the existence of a parent for each vertex. The second condition states that if a vertex has children in both dimensions, then each child in one dimension has to have children in the other. This represents the case of a subband being decomposed horizontally and vertically, producing four subbands. The third condition works reverse: if both children of a vertex in one dimension have children in the other, then the vertex must also have children in the other dimension.

There are incorrect binary bushes which violate at least one of the two latter conditions. See Figure 8 for a visualization of correct and incorrect bushes. The first example is incorrect because it violates the second condition. The third example violates the third condition.

A *full* binary bush is one where each vertex has either 0 or exactly 2 children in each dimension, i.e. $\exists\, \mathrm{child}(a, c, 1) \Leftrightarrow \exists\, \mathrm{child}(a, c, 2)$. Full binary bushes are important not only because anisotropic wavelet packet subbands can be arranged in full binary bushes, but also because full bushes are uniquely determined by the set of leaf vertices and vice versa. The proof can be found in [21]. Note that this also shows that the bush structure is able to represent all possible anisotropic wavelet packet decompositions.

The structure of an anisotropic wavelet packet decomposition is appropriately represented by full 2-D binary bushes. Each subband is created by a sequence of horizontal and a sequence of vertical filtering steps. These two sequences can be arbitrarily interleaved, so there are possibly many different ways to produce a subband. However, each subband has, of course, to be derived from the original data set, which forms the root node of the bush. Thus, the first condition is fulfilled. If a subband is filtered in both horizontal and vertical directions, then four subbands are created, all filtered in both directions. This behavior is formalized by the second condition. Finally, the third condition is not strictly necessary but plausible. It reflects the freedom of how to perform the decomposition by representing all possible ways to do so. Figure 9 shows an example of a bush of anisotropic wavelet packets.

As for the implementation of such a graph in a computer program, we remember that an $n$-ary tree can be implemented as an array, where the root is placed at position $p(r) = 0$ and the children of a node $a$ are placed at positions $np(a)+1, \ldots, np(a)+n$. Non-existent nodes leave holes in the array. Now, since a complete 2-D binary bush is the Cartesian product of two complete binary trees, a bush can be implemented as a 2-D array, where horizontal and vertical positions $j(a), k(a)$ of a bush-node $a$ are determined by positions $j(a) = p(a_1), k(a) = p(a_2)$ in the horizontal and vertical binary trees that build up the binary bush.

There are several applications where the bush structure is useful. One example is

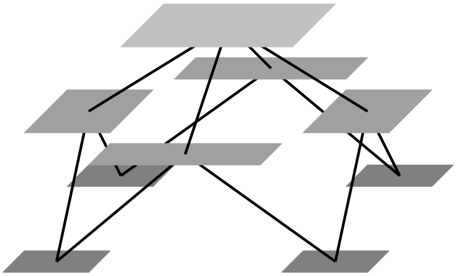


Figure 9: A complete bush of anisotropic wavelet packets with decomposition depth 1 in each dimension

the computation of the number of subbands common to two different decompositions as a means of measuring the similarity of the two decompositions. In the isotropic case this can be done by a simple tree-walk that skips branches that are missing in one of the two trees of subbands, and by counting common leaves. In the anisotropic case, however, a subband in the second decomposition that corresponds to a subband in the first decomposition would have to be searched by starting at the root node and then following those branches that lead in the direction of the desired subband. This had to be repeated for every subband. By using bushes, such a subband can be accessed directly by its two coordinates $j(a), k(a)$.

Another application is presented in Section 6.

# 5 Best Bases

To construct a decomposition basis not randomly but optimally, the so-called "best basis" has to be found. Several methods can be used to find the best wavelet packet basis for a given purpose [22, 23, 24]. Cost functions can be used that determine a measure of (negative) usability of a given data sequence for a given purpose. The well-known best-basis algorithm [22] for wavelet packet bases is an efficient way to calculate the best basis which minimizes the overall cost function value. The cost function must be additive, i.e. the cost of a set of subbands is equal to the sum of the subbands' costs.

The best basis algorithm relies on a tree-like decomposition structure. Therefore, it has to be modified to work with bushes. In the isotropic case the algorithm has to decide for each subband whether to decompose it or not. In the anisotropic case it can either decompose it horizontally, vertically, both or not at all. To be able to decide this, the best basis of the horizontally and vertically decomposed subbands and the minimized cost function values have to be compared against the subband's own cost function value. These values have to be determined recursively. The smallest has to be chosen.

In the isotropic case it is possible to include the actual calculation of the transform according to the best basis into the determination of the best basis. This avoids a second decomposition run and, thus, saves computation time. However, this is very complicated in the anisotropic case and requires an enormous amount of memory. Additionally, it does not pay because the actual decomposition has a lower complexity than the determination of the best basis (see below). Therefore, the conventional second-run approach is chosen here.

```
BestDecomp ($S$) :=
    if bestdim($S$) > 0
        Filter ($S$, bestdim($S$))
        BestDecomp (child($S$, bestdim($S$), 1))
        BestDecomp (child($S$, bestdim($S$), 2))
```

Figure 10: Best basis decomposition algorithm based on a spanning tree walk. Leaf-vertices are indicated by bestdim(S)=0

- First, we have to calculate the cost for each subband. This is done by generating all subbands as in Figure 11 and calculating the cost for each subband immediately after it is created.

- Then, we determine for each subband the best cost and the best decomposition dimension by:

$$\begin{aligned}
\mathrm{bestcost}(S,k) &:= \begin{cases} \mathrm{cost}(S) & k=0 \\ \mathrm{bestcost}(\mathrm{child}(S,k,1)) + & \\ \quad \mathrm{bestcost}(\mathrm{child}(S,k,2)) & k=1\dots n \end{cases} \\
\mathrm{bestdim}(S) &:= \operatorname*{argmin}_{k=0\dots n} \mathrm{bestcost}(S,k) \\
\mathrm{bestcost}(S) &:= \mathrm{bestcost}(S,\mathrm{bestdim}(S))
\end{aligned}$$

  This is a recursive procedure ending at leaf subbands (subbands at maximum decomposition level) where bestdim($S$) is set to 0 and bestcost($S$) = bestcost($S$, 0) = cost($S$).

- The best basis decomposition is then performed based on original data by applying a tree-walk on a spanning tree. This leads to the algorithm in Figure 10.

**Complexity** As opposed to isotropic wavelet packets, performing a full decomposition is not the same task as calculating all possible subbands for anisotropic wavelet packets. This is, by the way, a consequence of the fact that bushes are not cycle free. To perform a full decomposition, i.e. calculating all subbands with maximum decomposition level in each direction, is just as much work as for isotropic wavelet packets, i.e. $O(n \log n)$. This is because the resulting subbands actually contain isotropic wavelet packets. Accordingly, the decomposition to a specific basis as in Figure 10 has the same complexity.

To see how much work it is to calculate all possible subbands, look at Figure 11. Each coefficient is calculated by the inner product of a filter and another subband. So we have to count the number of coefficients. Following the construction of Figure 11 we get

$$\mathrm{complexity} = n((j+1)^d - 1) = O(n(\log n)^d) \tag{46}$$

where $n$ is the data size, $d$ is the dimension ($d = 2$ in our case) and $j$ is the decomposition depth. Here we assume that $j = \log_2 \sqrt[d]{n} - c$, which means that the decomposition depth is increased by 1 if the data size is increased by a factor of 2 in each dimension.

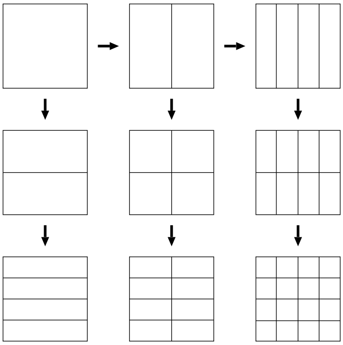


Figure 11: All subbands of a full anisotropic wavelet packet decomposition

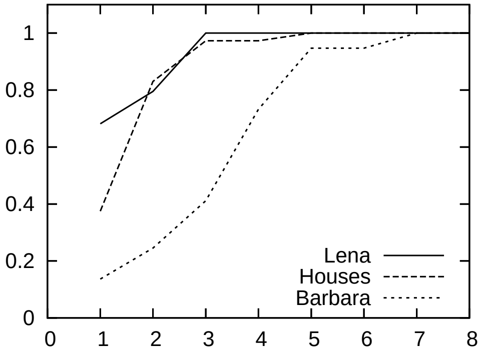


Figure 12: Accuracy of the truncated best basis algorithm: the ratio of leaf subbands in the best basis solution that are also present in truncated best basis algorithm solution. Results for three standard images are plotted over the truncation depth $j'$. Maximum decomposition depth is 8 in each dimension.

**Truncated Best Basis Algorithm** Since the complexity of the best basis algorithm is worse than in the isotropic case ($O(n \log n)$), we consider a heuristic approximation to the best basis algorithm in order to reduce the complexity. The idea is to decompose data not to maximum depth but to a fixed (truncated) depth $j'$ and find the decomposition decisions (bestdim($S$)) in the usual way. We then accept this result for the root subband only and proceed with its child subbands in the same way.

The first step in this approach has a complexity of $n((j'+1)^d-1)$. However, consecutive steps only have to produce complementary data if the data of previous steps is kept in memory, which means significantly increased memory demand. If implemented this way, the total complexity is $n((j'+1)^d - 1 + (j'+1)^{d-1}(j-j')d)$ in the worst case.

This heuristic assumes that the decision how to decompose a subband does not depend on subbands at much higher decomposition depth. Whether this assumption is justified, can be seen in Figure 12, where decompositions of actual images based on the best basis and the truncated best basis algorithms are compared in terms of the number of leaf subbands common to both decompositions (see end of Section 4). Especially for images with high frequency content (such as the "Barbara"-image), the truncation depth $j'$ should be chosen large enough in order to avoid loss of accuracy in the application, since the decomposition would differ significantly from the optimum otherwise.

# 6 Efficient Coding of the Decomposition Structure

Independent of the application in which anisotropic wavelet packets are used, it may be appropriate to store the decomposition basis, i.e. its graph structure, with a minimum amount of bits. We discuss two approaches to encode anisotropic wavelet packet structures: a straightforward method and a method that utilizes the bush structure to achieve a better representation.

## 6.1 Simple Coding

The straightforward method codes a structure depth-first from root to leaves. It is based on the way the decomposition structure is created: on the basis of the decomposition decision, each node $S$ is conceived to have offsprings in only one dimension $\dim(S)$ . For each node a symbol $\texttt{node}_d$ is emitted that denotes the dimension $d = \dim(S)$ in which the node is decomposed. The symbol `leaf` denotes a leaf. The algorithm is given in figure Figure 13. Of course, this approach is not very efficient: for an isotropic decomposition step, the direction is coded three times, while with an efficient representation no direction needs to be coded.

```
Encode $(S)$ :=
    if $S$ is leaf
        Emit symbol leaf
    else
        let $d = \dim(S)$
        Emit symbol $\texttt{node}_d$
        Encode $(\text{child}(S, d, 1))$
        Encode $(\text{child}(S, d, 2))$
```

Figure 13: A straightforward algorithm to encode the decomposition structure of an anisotropic wavelet packet transform

## 6.2 Bush Coding

To encode a decomposition structure optimally (with a minimum number of bits) it is essential to start with a unique representation. Otherwise, the encoded bits necessarily contain redundancy. Fortunately, a bush *is* a unique representation of the decomposition structure. However, this does not mean that it is trivial to encode this structure without redundancy. The algorithm in Figure 14 does this.

It takes into account that branches of a node imply certain branches of descendant nodes: First, if the bush branches at a node in dimension $d$, then all child nodes in any dimension different from $d$ have to branch in dimension $d$. Second, if the bush does not branch at a node in dimension $d$, then at least one child node in a dimension different from $d$ does not branch in dimension $d$. The algorithm, therefore, keeps track of dimensions in

```
Encode ($S$, $D$, $N$) :=
    for all dimensions $d$
        if not $N[d]$ and not $D[d]$
            Encode (haschildren($S, d$))
    if a dimension $d$ exists with haschildren($S, d$)
        for all dimensions $e$
            $D_{1,2}[e] \leftarrow$ haschildren($S, e$) $\wedge\, d \neq e$
        for all dimensions $e$ with $\neg$ haschildren($S, e$)
            $D_1[e] \leftarrow$ haschildren(child($S, d, 1$), $e$)
            $D_2[e] \leftarrow$ haschildren(child($S, d, 2$), $e$)
            Encode $D_1[e]$ and $D_2[e]$ together (note: at least one is *false*)
            $N_1[e] \leftarrow \neg D_1[e]; N_2[e] \leftarrow \neg D_2[e]$
        Encode (child($S, d, 1$), $D_1$, $N_1$)
        Encode (child($S, d, 2$), $D_2$, $N_2$)
```

Figure 14: An algorithm to encode the decomposition structure of an anisotropic wavelet packet transform without redundancy

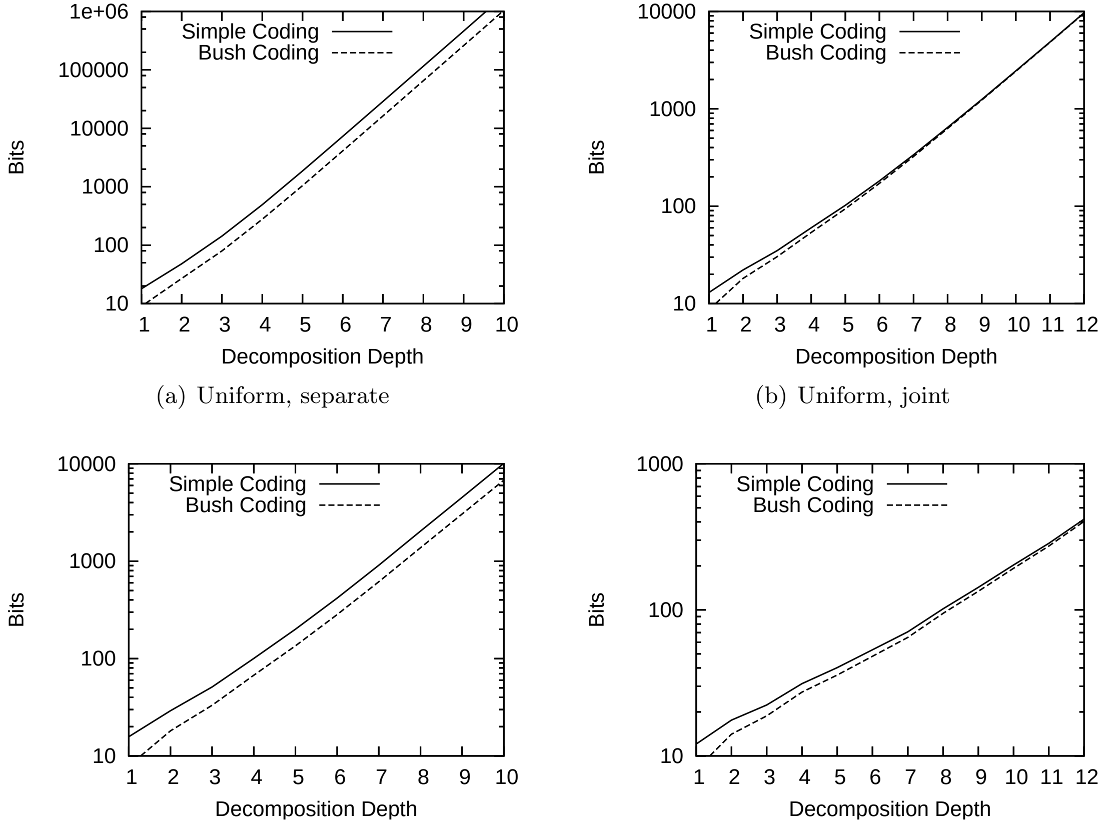
which nodes must or must not branch ($D$ means "do branch" and $N$ means "no branch") and does not encode this information. This removes all redundancy from the bush's structure.

The initial call is "Encode (root ($B$), $\emptyset$, $\emptyset$)". Additionally, nothing has to be encoded if the maximum decomposition depth is reached (if it is known). At one point, two Boolean values are encoded together where only three states are possible. To encode this without redundancy, an arithmetic coder is necessary. This coder also exploits probabilistic redundancies in the structure.

Now let us compare the two coding methods for the different distributions discussed above. Both, the simple and the bush representation, are entropy coded with an arithmetic coder and number of required bits is recorded. Figure 15 shows the average number of bits needed to code 10000 random samples for each decomposition depth. Figure 17 shows the coding gain as the ratio between the average number of bits needed for the bush representation and the simple representation.

It can be seen that the coding gain that can be obtained with the bush representation depends on the properties of the decomposition structure. For the uniform distribution, which produces decompositions that are much deeper the decompositions produced by the scale-invariant distribution, the coding efficiency between the simple and sophisticated representation differs substantially for joint and separate decomposition depths. Separate decomposition depths produce more balanced decomposition trees that are well suited for the bush representation. Randomly selected trees with a joint decomposition depth do not exhibit this balance and the bush representation cannot achieve much of a coding gain for higher maximum decomposition levels.

For the scale-invariant distribution with parameter $\alpha = 1.5$, the coding gain of the sophisticated approach is substantial for both, joint and separate decomposition depths, as can be seen in Figure 15(a) and Figure 15(c). Figure 16 shows the performance for different values of $\alpha$ (with maximum separate and joint decomposition depths of 10 and 12,



(a) Uniform, separate

(b) Uniform, joint

(c) Scaleinvariant ($\alpha = 1.5$), separate

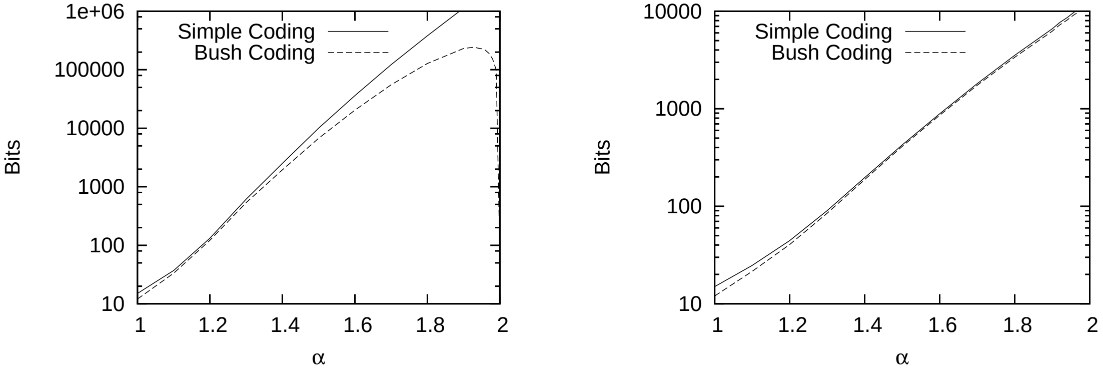
(d) Scaleinvariant ($\alpha = 1.5$), joint

Figure 15: Coding Efficiency by Decomposition Depth for Different Distributions

respectively) . Higher values of $\alpha$ produce deeper decompositions. In the case of separate decomposition depths, the resulting decomposition trees are more balanced in vertical and horizontal dimension. In this case, the performance of bush coding is significantly better than simple coding. For separate decomposition depths, $\alpha = 2$ produces a full wavelet packet decomposition. Because a full decomposition can be described very efficiently by the bush representation, the number of bits decreases dramatically for this case. In the case of joint decomposition depth the produced decompositions are much less balanced and the coding gain obtained by the bush representation is marginal, even for high values of $\alpha$.

# 7 Conclusion

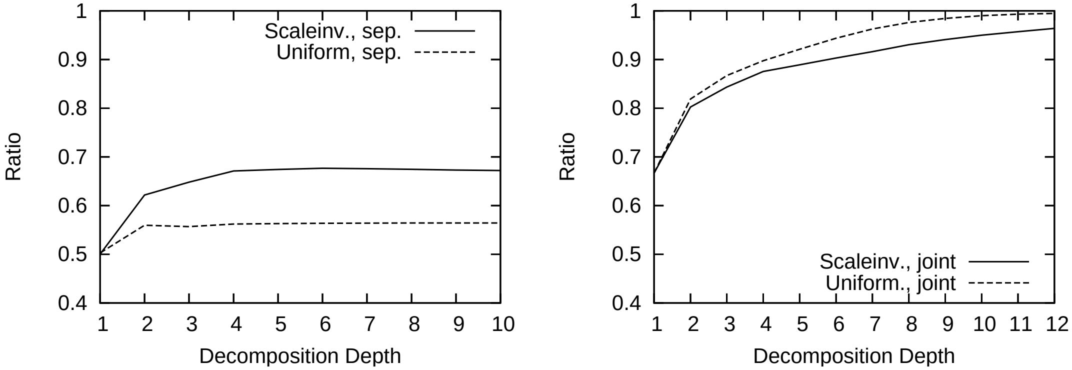
To work with the anisotropic wavelet packet transform, one has to construct and to represent its decomposition structure. For the first part, i.e. construction, this work develops a number of random decomposition with specific distributions, as well as a best basis algorithm for application driven optimized transforms. For the second part, i.e. representation, a novel graph structure is introduced, which has several essential advantages.



(a) Scaleinvariant, separate (max. depth = 10)

(b) Scaleinvariant, joint (max. depth = 12)

Figure 16: Coding Efficiency by $\alpha$ for Scaleinvariant Distributions



(a) Separate Depths

(b) Joint Depths

Figure 17: Ratio Bush/Simple Coding

As a first result, we see that the number of possible decompositions is much higher than in the isotropic case. As a consequence, a random decomposition from a uniform distribution is, on average, even deeper than in the isotropic case since decompositions with lower depth are rarer compared to the total number of decompositions. Therefore, another type of distribution is developed where the number of subbands grows exponentially with the maximum decomposition depth, and the base of the growth can be chosen. This leads to an approximately scale invariant decomposition. Furthermore, additional constraints that may be imposed by applications have an impact on the properties of the distributions, in particular the number of available bases is reduced.

When analyzing the properties of these random decompositions, it turns out that decompositions with a joint maximum decomposition depth for all dimensions produce a wider range of anisotropy of the resulting subbands, while those with a separate maximum decomposition depth tend to remain closer to isotropy.

To construct not only random decompositions but ones that are optimized with respect to a certain application, the well-known best basis algorithm for isotropic wavelet packet transforms is adapted to the anisotropic case. Although the hierarchic nature of this algorithm must be broken to do so, the optimal complexity of the algorithm is retained, i.e. the complexity necessary to compute all possible subbands.

All considerations show that the anisotropic wavelet packet transform is more complicated than the isotropic transform because the underlying graph structure is not a tree but a certain kind of cyclic but hierarchic graph. This graph is developed as a graph theoretic definition. It serves as the foundation to all algorithms in this work and turns out to be of high value for implementations of the anisotropic wavelet packet transform. Its main advantage is that it offers a unique representation of the decomposition structure. Thus, it is a necessary prerequisite for coding the decomposition structure without redundancy. A compressing algorithm based on this graph to code the structure information is developed and shows that there is a significant reduction of bits especially for decompositions with separate maximum decomposition depths.

# References

[1] I. Daubechies, *Ten Lectures on Wavelets.* No. 61 in CBMS-NSF Series in Applied Mathematics, Philadelphia, PA, USA: SIAM Press, 1992.

[2] S. Mallat, *A wavelet tour of signal processing.* Academic Press, 1997.

[3] 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.

[4] N. Saito and R. Coifman, “Local discriminant bases,” in *Wavelet Applications in Signal and Image Processing II* (A. Laine and M. Unser, eds.), vol. 2303 of *SPIE Proceedings*, pp. 2–14, 1994.

[5] A. Laine and J. Fan, “Texture classification by wavelet packet signatures,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 15, pp. 1186–1191, 1993.

[6] R. Learned and A. Willsky, “A wavelet packet approach to transient signal classification,” *Applied and Computational Harmonic Analysis*, vol. 2, pp. 265–278, 1995.

[7] A. Lindsey and J. Dill, “Digital transceiver implementation for wavelet packet modulation,” in *Wavelet Applications V* (H. Szu, ed.), vol. 3391 of *SPIE Proceedings*, pp. 255–264, SPIE, 1998.

[8] L. B. Montefusco, “Semi-orthogonal wavelet packet bases for parallel least-squares approximation,” *Journal of Computational and Applied Mathematics*, vol. 73, pp. 191–208, 1996.

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

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

[11] M. Wickerhauser, “INRIA lectures on wavelet packet algorithms.” Lecture notes, INRIA, 1991.

[12] R. Kutil, “Zerotree image compression using anisotropic wavelet packet transform,” in *Visual Communications and Image Processing 2003 (VCIP’03)* (T. Ebrahimi and T. Sikora, eds.), vol. 5150 of *SPIE Proceedings*, (Lugano, Switzerland), pp. 1417–1427, SPIE, July 2003.

[13] D. Xu and M. N. Do, “Anisotropic 2-D wavelet packets and rectangular tiling: theory and algorithms,” in *Proceedings of SPIE Conference on Wavelet Applications in Signal and Image Processing X* (M. A. Unser, A. Aldroubi, and A. F. Laine, eds.), vol. 5207 of *SPIE Proceedings*, (San Diego, CA, USA), pp. 619–630, SPIE, Aug. 2003.

[14] R. Kutil, “Anisotropic 3-D wavelet packet bases for video coding,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP’03)*, (Barcelona, Spain), Sept. 2003.

[15] E. Christophe, C. Mailhes, and P. Duhamel, “Best anisotropic 3-D wavelet decomposition in a rate-distortion sense,” in *Proceedings of 31st International Conference on Acoustics, Speech, and Signal Processing, ICASSP ’06*, vol. II, (Toulouse, France), pp. 17–20, May 2006.

[16] A. Pommer and A. Uhl, “Selective encryption of wavelet-packet encoded image data — efficiency and security,” *ACM Multimedia Systems (Special issue on Multimedia Security)*, vol. 9, no. 3, pp. 279–287, 2003.

[17] D. Engel and A. Uhl, “Secret wavelet packet decompositions for JPEG 2000 lightweight encryption,” in *Proceedings of 31st International Conference on Acoustics, Speech, and Signal Processing, ICASSP ’06*, vol. V, (Toulouse, France), pp. 465–468, May 2006.

[18] D. Engel and A. Uhl, “Lightweight JPEG2000 encryption with anisotropic wavelet packets,” in *Proceedings of International Conference on Multimedia & Expo, ICME ’06*, (Toronto, Canada), pp. 2177–2180, July 2006.

[19] D. Engel and A. Uhl, “An evaluation of lightweight JPEG2000 encryption with anisotropic wavelet packets,” in *Security, Steganography, and Watermarking of Multimedia Contents IX* (E. J. Delp and P. W. Wong, eds.), Proceedings of SPIE, (San Jose, CA, USA), SPIE, Jan. 2007. to appear.

[20] 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*, pp. 41–47, May 2002.

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

[22] R. Coifman and M. Wickerhauser, “Entropy based methods for best basis selection,” *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 719–746, 1992.

[23] K. Ramchandran and M. Vetterli, “Best wavelet packet bases in a rate-distortion sense,” *IEEE Trans. on Image Process.*, vol. 2, no. 2, pp. 160–175, 1993.

[24] N. M. Rajpoot, R. G. Wilson, F. G. Meyer, and R. R. Coifman, “A new basis selection paradigm for wavelet packet image coding,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP’01)*, (Thessaloniki, Greece), pp. 816–819, Oct. 2001.