The Graph Structure of the Anisotropic Wavelet Packet Transform

Rade Kutil

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

Abstract. The sub-bands of a wavelet packet transform can be arranged in a so called decomposition tree, where each sub-band is calculated from its parent sub-band by the application of a low- or high-pass filter (in each dimension for multidimensional transforms). Anisotropic wavelet packets are a generalisation of wavelet packets where the constraint of equal decomposition depths in each dimension is dropped. Such sub-bands cannot (at least not in a unique way) be arranged in a tree. This work develops the graph structure that underlies the set of sub-bands of anisotropic wavelet packet transforms and shows the benefits of this unique representation.

1 Introduction

Wavelets [2, 6] are a great tool for signal/image compression [11], feature extraction [8], classification algorithms [3, 4], telecommunication applications [5], numerical mathematics [7], and many more. The fast wavelet transform (FWT) uses a quadrature mirror filter pair (QMF) to decompose a signal into a high and a low frequency sub-band by convolution and downsampling by two. Higher dimensional data are transformed by consecutive filtering in all dimensions. This is repeated for the low frequency sub-band only in pyramidal decomposition and for arbitrary sub-bands in wavelet packet decomposition [10]. See Figure 1.

If we consider bases of wavelet packets whose scales are different in different directions (anisotropic wavelet packets [9]), then the number of sub-bands and the number of bases increases substantially. Additionally, the method of representing specific bases by a decomposition tree is not applicable any more. As filtering steps can be interchanged almost arbitrarily, more than one *decomposition path* can be found for most sub-bands. To avoid ambiguous representations of equal bases and redundant calculations, we introduce a new graph structure for wavelet packet decompositions that is a generalisation of trees. Because of its interwoven nature, we will call this structure *bush*.

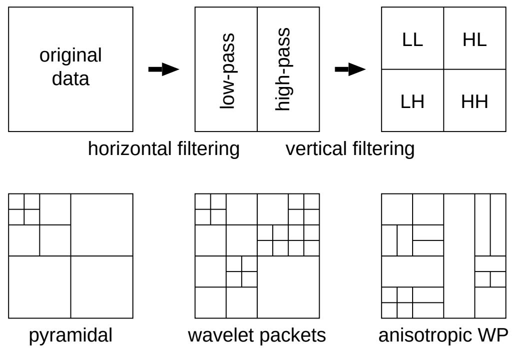


Fig. 1. The construction of 2-D wavelet transforms. Traditional wavelet packet transforms always apply horizontal and vertical low- and high-pass filtering on sub-bands to generate new sub-bands (in the pyramidal case only on the approximation sub-band (LL); in the wavelet packet case on arbitrary sub-bands), while the anisotropic wavelet packet transform possibly performs the filtering in only one dimension

2 Bushes

In general, we introduce a bush as a special coloured graph $B = (V, E, C)$ where V is a set of vertices (or nodes), C is a set of colours and $E \subseteq V \times V \times C$ is the set of edges which connects two vertices and is associated with a certain colour. We will call the number of colours *dimension*. We write $a \overset{c}{\sim} b$ for $(a, b, c) \in E$. The edges should be non-directional: $a \overset{c}{\sim} b \Rightarrow b \overset{c}{\sim} a$.

To be a bush, the coloured graph B has to satisfy some conditions. Before we state these conditions, we have to introduce a mapping (projection) of a coloured graph to a colourless graph that contracts every two vertices to a single vertex that are connected by a path that does not include a specific colour.

Definition 1. *Let $B = (V, E, C)$ be a coloured graph. Let $G = (V', E')$ be a complete uncoloured graph with at least as many vertices as B. For each colour $c \in C$ a "bush projection" is a mapping $f_c : V \longrightarrow V'$ which satisfies*

$$a \overset{d}{\sim} b \wedge d \neq c \Rightarrow f_c(a) = f_c(b) \, .$$

Now, we can define the projection of B:

$$f_c(E) := \{(f_c(a), f_c(b)) \,|\, a \overset{c}{\sim} b\}$$
$$f_c(B) := (f_c(V), f_c(E))$$

This means that $f_c(B)$ is an "orthogonal" projection of the coloured graph. Now, we will define the conditions for a bush: Firstly, the projections must be trees. Secondly, if all projections of two vertices a, b are equal then a and b are equal (while the reverse is trivial).

Definition 2. *A bush $B = (V, E, C)$ is a coloured graph such that for each colour c there exists a bush projections f_c with:*

- $\forall c \in C : f_c(B)$ *is a tree and*
- *if* $\forall c \in C : f_c(a) = f_c(b)$ *then* $a = b$.

The second condition states that the projections of a vertex are sufficient to localise the vertex. The question is now if and how whole bushes can be characterised by their projected trees. The following theorem gives the answer. Before that, we need two definitions, though.

Definition 3. *Let $T_1, \ldots, T_m$ be trees, where $T_i = (V_i, E_i)$. The product of trees is*

$$T_1 \times T_2 \times \ldots \times T_m := (V, E, C)$$

where

$$V := V_1 \times \ldots \times V_m \quad C := \{1, 2, \ldots, m\}$$

$$E := \{(a, b, c) \,|\, a, b \in V \,, \forall c' \neq c : a_{c'} = b_{c'}, (a_c, b_c) \in E_c\}$$

Definition 4. *Let $B_1 = (V_1, E_1, C), B_2 = (V_2, E_2, C)$ be coloured graphs. We write $B_1 \subseteq B_2 :\Leftrightarrow V_1 \subseteq V_2 \wedge E_1 \subseteq E_2$.*

Theorem 1. *B be a connected coloured graph with dimension m. Then B is a bush if and only if there exist m trees $T_1, \ldots, T_m$ such that $B \subseteq T_1 \times T_2 \times \ldots \times T_m$.*

An important class of trees are rooted trees and n-ary trees. We will now introduce similar classes for bushes.

Definition 5. *A rooted bush B is a bush with a single special vertex r that is called root and:*

- $\forall c \in C : f_c(r)$ *is the root of* $f_c(B)$,
- $\forall b \neq r \exists a, c : a \overset{c}{\sim} b \wedge f_c(a) = \mathrm{parent}(f_c(b))$.

The second condition assures the existence of a parent for each vertex. This means that there is a path from the root to each vertex which maps to a path in all $f_c(B)$.

Definition 6. *An n-ary bush is a rooted bush where*

1. $\forall c \in C : f_c(B)$ *is an n-ary tree,*
2. *if* $\exists b_1, b_2, c_1 \neq c_2 : \mathrm{child}(a, c_1, k_1) = b_1 \wedge \mathrm{child}(a, c_2, k_2) = b_2$
   then $\exists d : \mathrm{child}(b_1, c_2, k_2) = \mathrm{child}(b_2, c_1, k_1) = d$,

3. *for all colours* c_1, c_2*,* $c_1 \neq c_2$*, and* $m = 1 \ldots n$*:*
   if $\forall k = 1 \ldots n : \exists d = \mathrm{child}(\mathrm{child}(a, k, c_1), m, c_2)$
   then $\exists e = \mathrm{child}(a, m, c_2)$*,*

where, similar to n-ary trees, we define

$$\mathrm{child}(a, c, k) = b :\Leftrightarrow a \overset{c}{\sim} b \wedge \mathrm{child}(f_c(a), k) = f_c(b)$$
$$\mathrm{parent}(b, c) = a :\Leftrightarrow a \overset{c}{\sim} b \wedge \mathrm{parent}(f_c(b)) = f_c(a)$$

where $\mathrm{child}(a, k)$ *and* $\mathrm{parent}(a)$ *give the k-th child and the parent of a vertex respectively in an n-ary tree.*

There are incorrect n-ary bushes which only satisfy the first condition (the projections are n-ary trees) but break at least one of the two other conditions. See Figure 2 for a visualisation of correct and incorrect bushes.

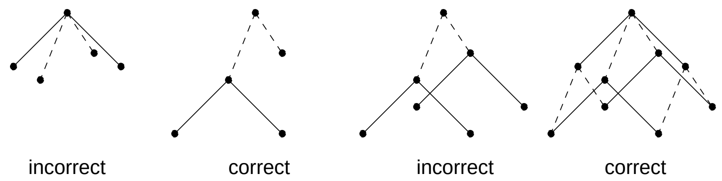


Fig. 2. Correct and incorrect binary bushes with dimension 2

The first example is incorrect because it breaks the second condition. The third example breaks the third condition.

Theorem 2. *Let* B *be a bush,* $a, b \in B$*. If* $\forall c : f_c(b) \in \mathrm{desc}(f_c(a)) \vee f_c(b) = f_c(a)$ *then* $b \in \mathrm{desc}(a) \vee b = a$*. (*$\mathrm{desc}(a)$ *denotes the set of descendants of a vertex a.)*

A corollary of this theorem is that if $f_c(b) = \mathrm{child}(f_c(a), k)$ and $\forall c' \neq c : f_{c'}(b) = f_{c'}(a)$ then $b = \mathrm{child}(a, c, k)$ (and vice versa, of course). Another corollary is the following theorem.

Theorem 3. *Let* $B_1 = (V, E_1, C)$ *and* $B_2 = (V, E_2, C)$ *be two n-ary bushes which share one set of projection functions* f_c*. Then* $E_1 = E_2$ *and* $B_1 = B_2$*.*

Definition 7. *A "full n-ary bush" is an n-ary bush where each vertex has either 0 or exactly n children referring to a single colour. We write* $\mathrm{haschildren}(a, c) :\Leftrightarrow \exists\, \mathrm{child}(a, c, 1) \Leftrightarrow \cdots \Leftrightarrow \exists\, \mathrm{child}(a, c, n)$*.*

Full n-ary bushes are an important class of n-ary bushes not only because anisotropic wavelet packet sub-bands can be arranged in full binary bushes (as we will see), but also because full bushes are uniquely determined by the set of leaf vertices. The following theorems shows this.

Theorem 4. *Let* $T_1, \ldots, T_m$ *be n-ary trees and* $B_1, B_2 \subseteq T_1 \times \ldots \times T_m$ *two full n-ary bushes. Then* $\mathrm{leaves}(B_1) = \mathrm{leaves}(B_2) \Rightarrow B_1 = B_2$*.*

Unfortunately and *not* according to trees, a similar theorem cannot be stated for non-full n-ary bushes.

The following theorem shows how to construct a tree as a subgraph of an n-ary bush which shares its leaves with the bush. This is especially useful to implement a walk (tree-walk) through a bush that visits each leaf vertex exactly once.

Theorem 5. *Let* B *be a bush. For each non-leaf vertex* a *we choose an arbitrary colour* c_a *such that* $\mathrm{haschildren}(a, c_a)$*.* T *be a the n-ary tree with the same root as* B *and* $\mathrm{child}(a, k) := \mathrm{child}(a, c_a, k)$*. Then* $\mathrm{leaves}(T) = \mathrm{leaves}(B)$*.*

An important question is how easy it is to implement a bush structure in a computer program. The first condition for n-ary bushes together with Definition 3 shows that the vertices can be identified by a vector of tree vertices. (How to implement trees is well known.) Fortunately, the other conditions for n-ary bushes are *local* conditions, i.e. local modifications of a bush (e.g. inserts and removals of vertices) can be handled locally, which avoids unnecessary computational complexity.

3 Bushes of Anisotropic Subbands

To show that the graph that connects all possible sub-bands is a correct binary bush, we choose the dimensions as the set of colours. Now, we have to show that the conditions of Definition 2 are satisfied. Clearly, the first condition is satisfied. If a sub-band is filtered in two distinct directions (high- and low-pass), then four sub-bands are created, all filtered in both directions. This is the behaviour formalised in the second condition for n-ary bushes. The third condition is not strictly necessary but plausible. It means that if both sub-bands produced by filtering in one direction are filtered in another direction, the result could as well be achieved if the order of filtering would be exchanged. Thus, we are dealing here with full binary bushes. Figure 3 shows an example of a bush of anisotropic wavelet packets.

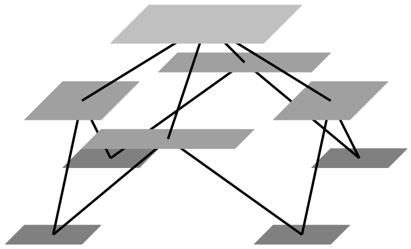


Fig. 3. A complete bush of anisotropic wavelet packets with decomposition depth 1 in each dimension

4 A Best Basis Algorithm

To choose the best basis (optimal bush structure) for a given purpose, we have to find some criterion to evaluate the bases. This is done by a cost function cost : $L^2(\mathbb{Z}) \to \mathbb{R}$ that determines a measure of (negative) usability of a given data sequence with respect to a given purpose. An example is the *entropy* for data compression.

As the number of possible bushes is large, it would mean a huge amount of work to test all bases to find the one that minimises the cost of the transformed data. For isotropic wavelet packet *trees*, there exists an efficient *best basis algorithm* [1]. It uses *additive* cost functions, i.e. for two data sets A, B: $\mathrm{cost}(A \cup B) = \mathrm{cost}(A) + \mathrm{cost}(B)$. The following is an equivalent algorithm for anisotropic wavelet packets:

- First, we have to calculate the cost for each sub-band.
- Then, we determine for each sub-band the the best cost and the best decomposition dimension by:

$$\mathrm{bestcost}(S,k) := \begin{cases} \mathrm{cost}(S) & k = 0 \\ \mathrm{bestcost}(\mathrm{child}(S,k,1)) + & \\ \quad \mathrm{bestcost}(\mathrm{child}(S,k,2)) & k = 1 \ldots n \end{cases}$$

$$\mathrm{bestdim}(S) := \operatorname*{argmin}_{k=0\ldots n} \mathrm{bestcost}(S,k)$$

$$\mathrm{bestcost}(S) := \mathrm{bestcost}(S, \mathrm{bestdim}(S))$$

  This is a recursive procedure ending at leaf sub-bands (sub-bands at maximum decomposition level).
- The best basis decomposition is then performed based on original data by applying a tree-walk on a spanning tree as in Theorem 5. Whereas in Theorem 5 an arbitrary branching dimension c_a is selected, we have to choose bestdim(a) here (0 indicates a leaf node).

5 Encoding the Bush's Structure

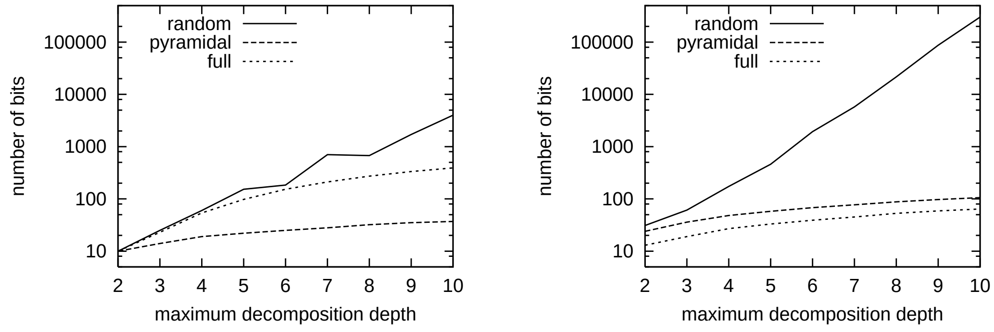
To encode a decomposition structure optimally (with a minimum number of bits) it is essential to start with a unique representation. Otherwise, the encoded bits necessarily contain redundancy. Fortunately, a bush *is* a unique representation of the decomposition structure. However, this does not mean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

Encode (S, dobranch, nobranch) :=
  for all dimensions d
    if not nobranch[d] and not dobranch[d]
      Encode (haschildren(d))
  if a dimension d exists with haschildren(S, d)
    for all dimensions e
      $D_{1,2}[e] \leftarrow \text{haschildren}(S, e) \wedge d \neq e$
    for all dimensions e with $\neg\,\text{haschildren}(S, e)$
      $D_1[e] \leftarrow \text{haschildren}(\text{child}(S, d, 1), e)$
      $D_2[e] \leftarrow \text{haschildren}(\text{child}(S, d, 2), e)$
      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)

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. Figure 4 shows the encoding results compared to a similar but much simpler algorithm for isotropic wavelet packets. Note that in both cases regular structures such as the pyramidal and the full decomposition can be encoded very efficiently. The higher bit-rates for the anisotropic case are a consequence of the fact that the number of possible bushes is much higher than the number of trees with the same maximum depth.

Acknowledgements

The author was partly supported by the Austrian Science Fund FWF, project no. P13903. The author also thanks the Austrian Science and Research Liaison Office in Košice for the conference presentation support.



(a) isotropic WP (b) anisotropic WP

Fig. 4. Number of bits necessary to encode a 2-D isotropic wavelet packet tree structure (a) and a 2-D anisotropic wavelet packet bush structure (b). The numbers for the random decomposition represent the maximum of ten randomly generated decompositions.

References

1. R.R. Coifman and M.V. Wickerhauser. Entropy based methods for best basis selection. *IEEE Transactions on Information Theory*, 38(2):719–746, 1992.
2. Ingrid Daubechies. *Ten Lectures on Wavelets.* Number 61 in CBMS-NSF Series in Applied Mathematics. SIAM Press, Philadelphia, PA, USA, 1992.
3. A. Laine and J. Fan. Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(15):1186–1191, 1993.
4. R.E. Learned and A.S. Willsky. A wavelet packet approach to transient signal classification. *Applied and Computational Harmonic Analysis*, 2:265–278, 1995.
5. A.R. Lindsey and J.C. Dill. Digital transceiver implementation for wavelet packet modulation. In H.H. Szu, editor, *Wavelet Applications V*, volume 3391 of *SPIE Proceedings*, pages 255–264. SPIE, 1998.
6. S. Mallat. *A wavelet tour of signal processing.* Academic Press, 1997.
7. L. Bacchelli Montefusco. Semi-orthogonal wavelet packet bases for parallel least-squares approximation. *Journal of Computational and Applied Mathematics*, 73:191–208, 1996.
8. N. Saito and R.R. Coifman. Local discriminant bases. In A.F. Laine and M.A. Unser, editors, *Wavelet Applications in Signal and Image Processing II*, volume 2303 of *SPIE Proceedings*, pages 2–14, 1994.
9. M.V. Wickerhauser. INRIA lectures on wavelet packet algorithms. Lecture notes, INRIA, 1991.
10. M.V. Wickerhauser. *Adapted wavelet analysis from theory to software.* A.K. Peters, Wellesley, Mass., 1994.
11. Zixiang Xiong, Kannan Ramchandran, and Michael T. Orchard. Wavelet packet image coding using space-frequency quantization. *IEEE Transactions on Image Processing*, 7(6):892–898, June 1998.