Image compression with adaptive Haar-Walsh tilings

Maj Lindberg[a] and Lars F. Villemoes[b]

[a]Department of Mathematics, Åbo Akademi University, Åbo, Finland.
[b]Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden.

ABSTRACT

We perform adaptive joint space and frequency tilings including all levels in the Haar-Walsh wavelet packet tree for two-dimensional signals. The method gives surprisingly good results in terms of nonlinear approximation. The visual quality of the compressed images with this method is the same as the quality using twice the number of coefficients for wavelets and standard wavelet packets when Haar filters are used. When all levels are allowed the cost for description of the location of the winning coefficients is not negligible. A tiling information vector is introduced for description of the chosen basis and the original image can be easily and quickly reconstructed using this information. For image compression this tiling information vector is compressed to only those nodes which correspond to kept coefficients, and this makes the adaptive scheme competitive.

Keywords: Anisotropic wavelet packets, best basis, space-frequency transform, Haar filter, Walsh functions, Image coding.

1. INTRODUCTION

Although the multiscale wavelet analysis is a powerful tool for image compression, it is well known that oscillatory or repetitive patterns are poorly represented and therefore lost at low bit rates. Adapted orthogonal bases such as wavelet packets, local cosines, and brushlets have been invented to address this issue.[845] In these schemes a partition of *either* the direct domain or the frequency domain is performed adapting to the given image. As the operations of windowing and filtering do not commute in general, former studies of joint space-frequency methods employ a very limited number of levels, for reasons of complexity.

We study complete space-frequency adaptability in the case of Haar-Walsh filters, because then the global best basis search is tractable. In each step of the transform, subimages are split in halves or Haar-transformed either horizontally or vertically. Complete adaptability signifies that we allow any of the four choices for any of the subimages at each level and that the process is continued through all levels until the subimages are of size one.

The goal of this research was to answer the following two questions.

1. Does full adaptability make a difference compared to restricted adaptability of wavelet packet type ?

2. Is the advantage lost in the additional price of coding the chosen transform ?

The surprising answer to (1) in terms of nonlinear approximation is that approximately the same visual quality and PSNR is obtained with half the number of kept coefficients compared to any of the before-mentioned restricted adaptability methods. For (2), we develop a new scheme for coding only the tree of filterings necessary to reconstruct the image. The result is that the joint space-frequency method still wins in terms of overall bitrates. The developed scheme for coding the tree of filterings could even be useful for more general joint space-frequency transforms.

Several gray scale images have been compressed using four different methods. The methods used are: Haar wavelets, adaptive anisotropic Haar-Walsh wavelet packets, local Walsh bases, and the method of adaptive joint space and frequency tilings. Only Haar-Walsh filters have been used. These filters have poor frequency resolution but they are easy to use and are free from the problems at edges. Joint space and frequency tilings are possible for these filters because of the disjoint supports of the corresponding wavelet functions. In these experiments the adaptive joint space and frequency segmentation method gave the best results for all test images at different compression ratios.

Correspondence: E-mail: fam.lindberg@pp.inet.fi, larsv@math.kth.se

The adapted joint space and frequency tiling method is based on the search algorithm for computing optimal Walsh packet basis of Thiele and Villemoes[6] and Herley et al.[2] The method gives the best segmentation of a given signal when the tilings are made in both space and frequency. In the studied Haar-Walsh setting, the anisotropic wavelet packet bases of Bennett[1] are described by a tiling with rectangles of the frequency plane only. In Herley et al.,[2] the wavelet packet tree was not allowed to grow to levels greater than 4. When the best basis choice is made among the levels less than 4 the cost of sending the location information is negligible.

The number of possible bases grows considerably when we allow tilings in space as well as in frequency. We have to choose a best basis among all these possibilities and one should think of the resulting basis as a tiling of a 4-dimensional space-frequency hypercube by boxes of unit volume and dyadic sides. We compare the cost of all pairs of space-neighbors with the cost for their frequency domain versions, and choose the tiling which gives the minimum cost for each of the pairs and keep track of the choices. The algorithm yields the best possible tiling that can be obtained by successive splitting. At each node we can split a two-dimensional signal in two halves by tiling in space in x- or y-direction or in frequency in x- or y-direction. We use four different marks to keep track of the type of split. For an image of size $N \times N$ the search will give a tiling information vector of length $N^2 - 1$, where each entry is one of the four marks used.

In practice we save only the largest coefficients and their locations. Then the tiling information vector can be compressed to only those nodes which have information about the kept coefficients. After this compression the location of the chosen basis can be given using two tiling information vectors and the significance map for the kept coefficients. All these vectors consist of ones and zeros and can be run-length encoded. For real images like Lena, this gives the total need of memory in bits per kept coefficient for the description of the location of about twice the amount needed for the significance map for wavelets when the same method of run-length encoding is used. See Figures 5, 6 and Table 1.

The total computational complexity of the method is of $\mathrm{O}\{(N \log N)^2\}$. The best basis search is expensive both in terms of time and memory consumption, but when we know the best basis tilings the computation of the coefficients can be made easily and quickly. Even reconstructions are easy to compute using the tiling information and the significance map. We just join the coefficients in space or in frequency depending on the marks in the tiling information vector and with each other or with zeros depending on the marks in the significance map. The freedom of choosing among all levels in the wavelet packet tree is really used as it can be seen on Figure 1.

The result of this paper is that we in the joint space and frequency segmentation method when it is combined with the new method for location information have an image compression method which clearly outperforms wavelet and other wavelet packet methods for gray scale images when Haar-Walsh filters are used.

2. HAAR-WALSH WAVELET PACKETS

Let $W_0(x) = 1$ for $0 \leq x < 1$ and zero elsewhere. Define $W_1, W_2, \ldots$ by the recursion

$$\begin{aligned} W_{2n}(x) &= W_n(2x) + W_n(2x-1), \\ W_{2n+1}(x) &= W_n(2x) - W_n(2x-1). \end{aligned} \tag{1}$$

This sequence of functions is equal to the Walsh system[7] on $[0,1[$ and zero outside. From this system we then form a larger collection by dyadic rescaling. A *Walsh atom* is defined by

$$w_p(x) = w_{j,k,n}(x) = 2^{-j/2} W_n(2^{-j}x - k),$$

where j, k, n are integers with $n \geq 0$. The corresponding *tile* p is the following dyadic rectangle of area one in the closed upper half plane,

$$p = I_p \times \omega_p = [\,2^j k, 2^j(k+1)\,[\times[\,2^{-j}n, 2^{-j}(n+1)\,[. \tag{2}$$

Then w_p vanishes outside I_p and ω_p is the frequency support of w_p in the Walsh sense. The most important property of Walsh atoms is that w_p and w_q are orthogonal functions on the line if and only if the tiles p and q are disjoint.[6] By a change of variable in the defining recursion (1), we get that if two pairs of tiles (l, r) and (d, u) cover the same dyadic rectangle of area two $l \cup r = d \cup u$, such that l is the left half, r the right half, d the lower half, and u the upper half, then the corresponding Walsh atoms are related by the Haar transform, involving the filters $h = (1,1)/\sqrt{2}$ and $g = (1,-1)/\sqrt{2}$.

$$\begin{pmatrix} w_d \\ w_u \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} w_l \\ w_r \end{pmatrix}. \tag{3}$$

We identify discrete signals of length $N = 2^L$ with functions on $[0, N[$ which are piecewise constant on the intervals $[0,1[, [1,2[, \ldots, [N-1, N[$. Then an orthonormal basis of Walsh atoms is obtained corresponding to each tiling of the space-"frequency" plane

$$S_N = [0, N[\times[0,1[$$

with pairwise disjoint tiles of the form (2). For each scale parameter $j = 0, 1, \ldots, L$ there are N tiles inside S, so there are $(L+1)N$ basis vectors to choose from.

In the same manner we can identify an image of size $N \times N$ with a function $f(x,y)$ on $[0,N[\times[0,N[$ which is piecewise constant on squares of size one. We consider orthonormal bases for the space of such functions consisting of tensor products of Walsh atoms,

$$w_{(p,q)}(x,y) = w_p(x) w_q(y), \qquad p \times q \subset S_N \times S_N.$$

There are $(L+1)^2 N^2$ such two-dimensional Walsh atoms. The corresponding inner products $c_{(p,q)} = \langle f, w_{(p,q)} \rangle$ with the given image are arranged in a $(L+1)N \times (L+1)N$ wavelet packet coefficient matrix P, using lexicographic ordering of the (j,k,n)-indices of tiles in each space coordinate. Observe that $w_{(p,q)}$ is a local Walsh function vanishing outside the rectangle $I_p \times I_q$.

Two Walsh atoms $w_{(p,q)}$ and $w_{(\widetilde{p},\widetilde{q})}$ are orthogonal if and only if either $p \cap \widetilde{p} = \emptyset$ or $q \cap \widetilde{q} = \emptyset$. This rule leads to a huge number of different tilings $\mathcal{T}$ of $S_N \times S_N$, each corresponding to an orthonormal basis in which the image can be expanded

$$f(x,y) = \sum_{(p,q) \in \mathcal{T}} c_{(p,q)} w_{(p,q)}(x,y).$$

We wish to find the best basis in the sense that an additive cost $\mathrm{Cost}(c) = \sum_{(p,q)\in\mathcal{T}} \mathrm{Cost}(c_{(p,q)})$ is minimized. In order to be able to do this with reasonable computational complexity the class of tilings is restricted. We consider only tilings produced by successive splits of $S_N \times S_N$ in halves in space or frequency directions in either the x- or y-component. In the first step, we have the four choices

$$S_N \times S_N = \begin{cases} ([0,N/2[\times[0,1[) \times ([0,N[\times[0,1[) \cup ([N/2,N[\times[0,1[) \times ([0,N[\times[0,1[), \\ ([0,N[\times[0,1/2[) \times ([0,N[\times[0,1[) \cup ([0,N[\times[1/2,1[) \times ([0,N[\times[0,1[), \\ ([0,N[\times[0,1[) \times ([0,N/2[\times[0,1[) \cup ([0,N[\times[0,1[) \times ([N/2,N[\times[0,1[), \\ ([0,N[\times[0,1[) \times ([0,N[\times[0,1/2[) \cup ([0,N[\times[0,1[) \times ([0,N[\times[0,1/2[). \end{cases}$$

Each of the resulting dyadic hyper-rectangles $R = R_x \times R_y$ correspond to a subspace spanned by the Walsh atoms $w_{(p,q)}$ with $(p,q) \subset R$. The best basis problem inside a hyper-rectangle with given size parameters (l,m), defined by $(\mathrm{Area}(R_x), \mathrm{Area}(R_y)) = (2^l, 2^m)$, can be solved assuming it is already solved for all hyper-rectangles with size parameters $(l-1,m)$ and $(l,m-1)$. This is the basic recursive observation. The algorithmic organization is described in the following sections.

By restriction to frequency splits only we obtain as special cases all anisotropic Haar-Walsh wavelet packet bases including the usual Haar basis. By restriction to space splits only we recover all anisotropic local Walsh bases.

3. BEST BASIS SEARCH FOR THE JOINT SPACE-FREQUENCY METHOD

The best basis selection is made among all levels in the wavelet packet tree. The number of possible bases grows considerably when we allow tilings in space as well as in frequency. At each node we have to choose one of four possible tilings as the most efficient tiling. We can split the signal in two halves by tiling in space in x- or y-direction or in frequency in x- or y-direction. We have to choose a set of best basis among all these possibilities. The cost function used for the best basis selection in this paper is the ℓ^1-norm of the set of coefficients, $\mathrm{Cost}(c) = \sum_{(p,q)} |c_{(p,q)}|$.

We start the best basis search by computing the wavelet packet tree to the maximum depth. For an image of size $N \times N$, $N = 2^L$, this will result in a $(L+1)N \times (L+1)N$ coefficient matrix consisting of all levels in the tree. The best basis search is made in this matrix. We search the information about how the tilings should be made at each node in order to give us the coefficients corresponding to the best basis. We start the search at the smallest scale and compute the costs for all possible pairs of two space-neighbors in the x-direction and also the cost for their frequency versions giving the cost for corresponding frequency neighbors. We choose between tiling in space and tiling in frequency. What we actually do is putting together two building blocks of size one element to a 1×2 block of space or frequency neighbors. The space-neighbors can be found in the first LN columns of the $(L+1)N \times (L+1)N$ coefficient matrix P. The corresponding frequency versions are found in the columns $(N+1), \ldots, (L+1)N$. We compute the space-cost and the frequency-cost for all 1×2 blocks, compare the costs, save the winning cost and keep track of the choice. We call the winning cost matrix $C_{1,0}$ and the tiling mark matrix for $M_{1,0}$, using the size parameters (l, m) introduced in the previous section.

The size of $C_{1,0}$ and $M_{1,0}$ is $(L+1)N \times LN/2$. In the matrix $C_{1,0}$ the number of submatrices decreases by one in the x- direction and all of these submatrixes will have one half of the previous width. Next we compute the costs for all pairs of 1×2 blocks resulting in 1×4 blocks in matrix $C_{1,0}$. The resulting winning costs are saved in the matrix $C_{2,0}$ and the tiling information in the matrix $M_{2,0}$. We continue with $C_{2,0}$ and compute the costs for all time neighbors and their frequency versions and keep track of the choices. We move in the x- direction and compute the costs for the space- and frequency neighbors and continue until the winning cost matrix has only one column.

After this we move in the y-direction from the coefficient matrix in a similar fashion. The cost of the winning choice is saved in $C_{0,1}$ and the information of the type of the winning tile in $M_{0,1}$. After computing the matrix $C_{0,1}$, we do not need the wavelet packet matrix any more. The space in memory can be used for $C_{0,1}$.

The normal search process, with four possibilities for each tile, can now start. For $C_{0,1}$ we move to the right and build blocks of space and frequency neighbors by adding the elements in $C_{0,1}$ in the x-direction. The resulting blocks have to be compared with the blocks we build by adding elements in $C_{1,0}$ in the y-direction. The winning cost is saved in $C_{1,1}$. After computing $C_{1,1}$ we do not need $C_{1,0}$ any more and can reuse the memory for storage of $C_{1,1}$.

Continue then by comparing space and frequency neighbors in x-direction in $C_{1,1}$ with space and frequency neighbors in y-direction in $C_{2,0}$. The resulting winning costs are in $C_{2,1}$ and the corresponding marks in $M_{2,1}$. Continue in this manner until the cost matrix consists of only one column. In $C_{l,m}$ and $M_{l,m}$, the indices l and m count the steps in x- respective y-direction. The coefficient matrix P has $l = m = 0$, and we proceed all the way to $l = m = L$.

We also have to look for the corresponding tiling information for the previous generations of tilings. When we compare the block built from the elements in $C_{1,0}$ and $C_{0,1}$, we can find the marks for the first generation of tiles in $M_{1,0}$ and $M_{0,1}$. We store the information from the previous tilings by expanding the mark matrices into arrays of dimension three. Thus, behind each element in $M_{1,1}$ we have two elements which tell us how the tiling was made in the first stage. At stage (l, m) we have to find two marks in $M_{l-1,m}$ or $M_{l,m-1}$ consisting of the tiling information for the previous generation of tilings. In the third dimension behind these marks we have the tiling marks for all other generations of tilings corresponding to the winning tile at stage (l, m).

For each comparison in the x-direction the width of the cost matrix will decrease and for each comparison in y direction the height of it will decrease. After L stages in both directions we will have the total cost in the cost matrix. At this stage the tiling mark array has only one element at the front. Its size is $1 \times 1 \times (N^2 - 1)$ and it has the information of the winning tiles corresponding to the best basis joint space and frequency selection. For a $N \times N$ image this best basis search results in a tiling vector of length $N^2 - 1$ whose elements are one of the four numbers $0, 1, 2, 3$. We use $0 =$ tiling in time x, $1 =$ tiling in frequency x, $2 =$ tiling in time y, and $3 =$ tiling in frequency y.

3.1. A 4×4 example

Let X be the two-dimensional signal

$$X = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 0 & -1 & 2 & 3 \\ 1 & -4 & 5 & 6 \end{pmatrix}.$$

We compute the cost for all pairs of space and frequency neighbors, compare them and save the winning costs in $C_{1,0}, C_{2,0}, C_{0,1}, C_{1,1}, \ldots$ and the corresponding tiling information in $M_{1,0}, M_{2,0}, M_{0,1}, M_{1,1}, \ldots$. The coefficient matrix P and obtained tiling information matrices (top layer of arrays) are arranged in the table below.

P

$$
\begin{array}{|c|c|c|}
\hline
\begin{matrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 0 & -1 & 2 & 3 \\ 1 & -4 & 5 & 6 \end{matrix} &
\frac{1}{\sqrt{2}}\begin{pmatrix} 3 & 7 & -1 & -1 \\ 11 & 15 & -1 & -1 \\ -1 & 5 & 1 & -1 \\ -3 & 11 & 5 & -1 \end{pmatrix} &
\begin{matrix} 5 & -2 & -1 & 0 \\ 13 & -2 & -1 & 0 \\ 2 & -3 & 0 & 1 \\ 4 & -7 & 2 & 3 \end{matrix} \\
\hline
\frac{1}{\sqrt{2}}\begin{pmatrix} 6 & 8 & 10 & 12 \\ 1 & -5 & 7 & 9 \\ -4 & -4 & -4 & -4 \\ -1 & 3 & -3 & -3 \end{pmatrix} &
\begin{matrix} 7 & 11 & -1 & -1 \\ -2 & 8 & 3 & -1 \\ -4 & -4 & 0 & 0 \\ 1 & -3 & -2 & 0 \end{matrix} &
\frac{1}{\sqrt{2}}\begin{pmatrix} 18 & -4 & -2 & 0 \\ 6 & -10 & 2 & 4 \\ -8 & 0 & 0 & 0 \\ -2 & 4 & -2 & -2 \end{pmatrix} \\
\hline
\frac{1}{2}\begin{pmatrix} 7 & 3 & 17 & 21 \\ 5 & 13 & 3 & 3 \\ -5 & -1 & -7 & -7 \\ -3 & -7 & -1 & -1 \end{pmatrix} &
\frac{1}{\sqrt{2}}\begin{pmatrix} 5 & 19 & 2 & -2 \\ 9 & 3 & -4 & 0 \\ -3 & -7 & -2 & 0 \\ -5 & -1 & 2 & 0 \end{pmatrix} &
\begin{matrix} 12 & -7 & 0 & 2 \\ 6 & 3 & -2 & -2 \\ -5 & 2 & -1 & -1 \\ 3 & -2 & 1 & 1 \end{matrix} \\
\hline
\end{array}
$$

$M_{1,0}$

1	1	1	1
1	1	1	1
0	1	0	1
0	1	0	0
1	1	1	1
0	1	0	0
1	1	1	0
0	1	0	0
1	1	0	1
0	1	0	0
0	1	1	0
1	1	0	0

$M_{2,0}$

1
1
0
0
1
0
1
0
1
0
0
0

$M_{0,1}$

2	2	3	3	2	3	3	3	2	3	3	2
2	2	2	3	2	3	2	3	3	3	2	2
2	3	3	3	2	3	2	3	2	2	3	2
2	3	3	3	2	3	2	3	2	2	2	2

$M_{1,1}$

1	3	1	1
0	1	0	0
2	1	0	1
2	1	2	0

$M_{2,1}$

1
0
0
2

$M_{0,2}$

2	3	3	3	2	3	2	3	2	2	2	2

$M_{1,2}$

2	3	0	0

$M_{2,2}$

0

After $L = 2$ stages in both x and y direction we have only one element in the cost matrix and only one mark in the top layer of the tiling information array. Although this is not the way the actual algorithm is programmed, let us indicate how the rest of the tiling information can be collected from previous top layers. The top mark of $M_{2,2}$ is 0, which means that we can find the information about the previous tiling in $M_{1,2}$. This gives the marks 0 : 2 3. Marks 2 and 3 correspond to tilings in the y-direction, which means that the information of the generation before will be found in $M_{1,1}$. This gives 0 : 2 3 : 1 0 1 1. The marks 0 and 1 point to tilings in the x- direction, we go to $M_{0,1}$ and have the resulting tiling information vector

$$0 : 2\ 3 : 1\ 0\ 1\ 1 : 2\ 3\ 2\ 2\ 3\ 3\ 3\ 3.$$

Unfolding this information on the level of coefficients leads to the vector

$$(3/\sqrt{2}, 11/\sqrt{2}, -1, 0, 0, 1, -1, -4, 19/\sqrt{2}, 3/\sqrt{2}, -2/\sqrt{2}, 0, -7/\sqrt{2}, -1/\sqrt{2}, 0, 0)$$

which represents X in the chosen orthonormal basis.

4. COMPRESSION OF THE TILING INFORMATION VECTOR

The location information of the chosen coefficients can be given as (row, column)-indices in the $(L+1)N \times (L+1)N$ matrix P or as the tiling information vector of length $N^2 - 1$. The cost for saving the location information is not negligible. In practice we save only the largest coefficients and their locations. For real pictures the coefficients are spread over the whole $(L+1)N \times (L+1)N$ matrix, see Figure 1. We can see that we really use the freedom of choosing among all levels in the packet tree.

The information in the tiling mark vector is intended for all of the chosen best basis coefficients. Usually we save only the non-zero coefficients and their locations. The result of the best basis search, the tiling information vector, can be compressed to the information corresponding to only the kept coefficients.

The first mark in the tiling mark vector tells us how to tile at the first level of tilings, the following two marks tell us how to tile at the second level, the following four marks at third level and so on. At each node we split into two halves. This splitting can be made in four different ways. We can draw the tiling information vector as a dyadic

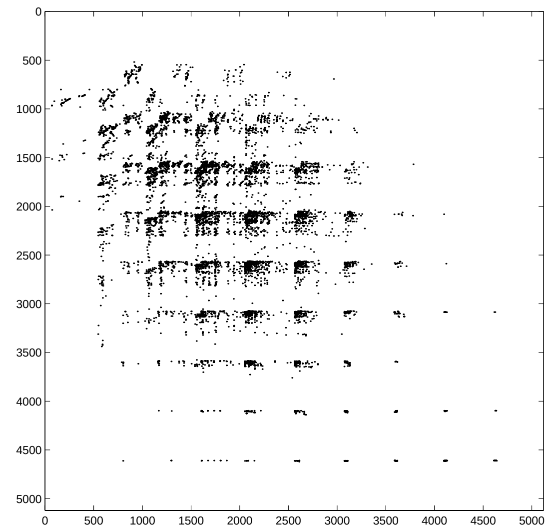


Figure 1. Location of the 8192 largest Walsh wavelet packet coefficients in the best basis for a 512×512 image, (Lena). The total address matrix of size 5120×5120 is composed of 100 submatrices each of size 512×512. All basis functions with addresses inside a submatrix share the same rectangular support shape. The original basis is the top left submatrix and the global Walsh basis is the bottom right submatrix. Wavelet and isotropic wavelet packet bases would be concentrated along the diagonal. We see that both global and very local functions are used and that very thin rectangular supports come into play.

tree structure, in which we read the tiling information level for level from the root down to the leaves. If we instead read the information in the tree for each leaf from the root down to the leaf we get individual tiling information for each coefficient. We can compute this individual tiling information for each of best basis coefficients from the tiling information vector. After computing the individual tiling information we save it only for the kept coefficients. For a $N \times N$ picture we have N^2 coefficients. We produce a $N^2 \times 2L$ matrix in which the rows consist of the individual tiling information for each of the best basis coefficients. The first individual mark for all coefficients is the same as the first element in the tiling information vector because all coefficients have been computed through this first tiling. Thus the first column in the individual tiling information matrix consists of N^2 marks of the same type as the first mark in the tiling information vector. In the second column there are $N^2/2$ marks of the same type as the second mark and $N^2/2$ marks of the same type as the third mark in the tiling information vector. After this we write the fourth mark in the first $N^2/4$ rows, the fifth mark in the following $N^2/4$ rows, the sixth mark in the next $N^2/4$ rows and the seventh mark in the last $N^2/4$ rows in the third column of the individual tiling information matrix. We continue until all marks in the tiling information vector are used. This gives us N^2 rows with $2L$ elements in each.

Usually we keep only a certain percentage of the largest coefficients. We need a significance map, a vector which tells us which coefficients are kept. In the individual tiling information matrix we then keep only the rows which belong to kept coefficients. This matrix still contains duplicate information. When we reconstruct the image from the kept coefficients, we read the tiling information for the coefficients. If two coefficients are put together at a certain node the information for them is the same from that node to the root of the tree and thus duplicated in the individual tiling information vectors. We can remove the duplicated information using the significance map, from which we can compute location in the tree and which of the two halves, upper or lower in space or high or low in frequency, a coefficient belongs to. The location of a non-zero coefficient is marked by 1 in the significance map. We can make pairs of the marks in the significance map. If the pair contains two ones it means that the corresponding coefficients will be put together already at the first level of the reconstruction, which means that the individual tiling information is the same for these coefficients and one of the duplicate rows in the tiling information matrix can be removed. For all pairs with two ones we remove one of the corresponding rows in the individual tiling information matrix. All pairs with one zero have only one of the coefficients among the kept coefficients.

The information in the individual tiling information matrix can be compressed to a vector which only contains the information about how to handle the kept coefficients when the image is reconstructed. There is no information duplicated in this new address vector. From the significance map we can read which half the coefficient belongs to and whether it should be put together with an other coefficient or with a zero. The new address vector tells if this is made in space or in frequency and in the x or the y direction. We can build the address vector from the individual tiling information matrix using following algorithm: Make pairs of the elements in the significance map. If the pairs contain two ones, remove one of the corresponding duplicate rows in the individual tiling information matrix. We start from the leaves, i.e. from the kept coefficients when we reconstruct the picture therefore we compress the individual tiling vectors for the kept coefficients starting from the last columns which corresponds to the leaves in the tree. We move the information in the last column of the individual tiling information matrix to be the first elements in the new address vector, which is a row vector. Remove then the last column in the tiling information matrix and put 0 in the significance map instead of the pairs which have two zeros and 1 instead of the other pairs. Then start from the beginning again: make pairs of the elements now in the significance map, for each pair consisting of two ones remove one of the corresponding duplicate rows and after that move the last column in the tiling matrix to the end of the address vector. Then remove the last column in the tiling matrix and change all pairs (0,0) in the significance map to 0 and the other pairs to 1. Continue until all elements in the individual tiling information matrix are removed.

4.1. Compressing the tiling information for the 4×4 example

Let us keep only the four largest coefficients in the example of Section 3.1. This gives the significance map $R =$ 010000011000100. We compute the individual tiling information for the coefficients and save it only for the kept coefficients.

Individual tiling vectors	Coefficient
0 2 1 2	$11/\sqrt{2}$
0 2 0 2	-4
0 3 1 3	$19/\sqrt{2}$
0 3 1 3	$-7/\sqrt{2}$

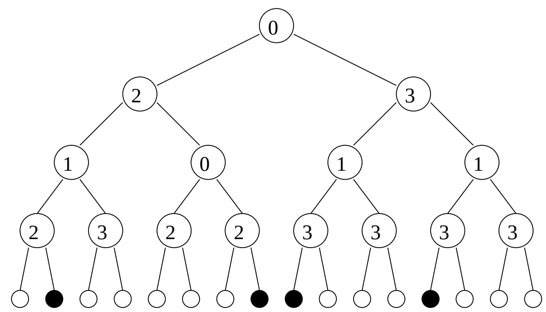


Figure 2. Tiling information vector for the example in this paper drawn as a dyadic tree structure. Each node of the tree shows how the tiling is done. The tree corresponds to the tiling vector 0 : 2 3 : 1 0 1 1 : 2 3 2 2 3 3 3 3. We get individual tiling information for each coefficient when we read the code from root to leaves for each leaf. We keep only the four largest coefficients and their individual tiling information. This gives the significance map 0100000110001000.

We can see in Figure 2 that there is a lot of information duplicated in the individual tiling vectors for the kept coefficients. The new address vector, i.e. the compressed tiling information vector for the kept coefficients is $C_Z =$ 2 2 3 3 1 0 1 1 2 3 0. It contains the tiling information at the nodes of the reduced dyadic tree which is obtained by removing nodes that only treats coefficients put to zero, see Figure 3. Clearly, we can write the information in C_Z in two vectors which both consist of ones and zeros and have the same length as C_Z. This gives C_Z in form of two other vectors which both can be run-length encoded.

5. RECONSTRUCTION OF THE IMAGE

Start the procedure for reconstruction by using the compressed tiling information vector C_Z and forming pairs of the entries in the significance map R. Write the pairs containing ones in a vector H, which tells whether both halves in the pair are included or not and which half the coefficient belongs to. After this, replace the pairs consisting of two zeros in R with 0 and the other pairs with 1. Start from the beginning by making pairs of the marks in the new significance map and continue to build H until R is empty.

When we reconstruct the signal we have the kept coefficients in the leaves of the tree. The first numbers in C_Z tell us what to do with these leaves. Compute the number of rows in R consisting of two ones, call it c_2, and

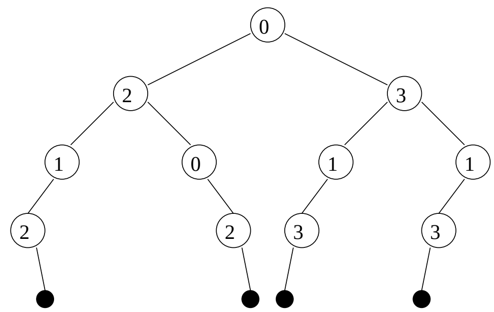


Figure 3. The result of compressing the tiling information vector for the 4×4 example of Section 3.1. The reduced tiling information is $C_Z = 2\ 2\ 3\ 3\ 1\ 0\ 1\ 1\ 2\ 3\ 0$, the significance map is $R = 0100000110001000$, and the 4 kept coefficients are $(11/\sqrt{2}, -4, 19/\sqrt{2}, -7/\sqrt{2})$. Only nodes which have to do with the kept coefficients need to be stored.

the number of pairs in R consisting of 01 or 10, call this number c_1. Read the first $c_2 + c_1$ marks in C_Z, i.e., the information about what to do with the coefficients in this first step. Join the coefficients in space or in frequency depending on the marks in C_Z. Join them with each other or with zeros depending on the marks in H. Continue this procedure and join the matrices from the first step with each other or with zero matrixes of the same size according to marks in C_Z and H. Continue until all marks in C_Z are used. For each mark in the compressed tiling information vector C_Z we read two marks in the vector H. Reconstruction simply corresponds to joining the coefficients using the information in C_Z and H.

5.1. Reconstruction of the image for the 4×4 example.

We have

$$\begin{array}{ll} \text{Tiling information (=how to join):} & C_Z = 2\ 2\ 3\ 3\ 1\ 0\ 1\ 1\ 2\ 3\ 0 \\ \text{Join together or with zeros:} & H = 01\ 01\ 10\ 10\ 10\ 01\ 10\ 10\ 11\ 11\ 11 \end{array}$$

$$\begin{array}{ccccccccc}
11/\sqrt{2} & \to & \begin{pmatrix} 0 \\ 11/\sqrt{2} \end{pmatrix} & \to & \begin{pmatrix} 0 & 0 \\ 11/2 & 11/2 \end{pmatrix} & \searrow & & & \\
 & & & & & & \begin{pmatrix} 0 & 0 \\ 11/2 & 11/2 \\ 0 & 0 \\ 0 & -4 \end{pmatrix} & \searrow & \\
-4 & \to & \begin{pmatrix} 0 \\ -4 \end{pmatrix} & \to & \begin{pmatrix} 0 & 0 \\ 0 & -4 \end{pmatrix} & \nearrow & & & \begin{pmatrix} 0 & 0 & 3 & 3 \\ 11/2 & 11/2 & 13/2 & 13/2 \\ 0 & 0 & 3 & 3 \\ 0 & -4 & 13/2 & 13/2 \end{pmatrix} \\
19/\sqrt{2} & \to & \begin{pmatrix} 19/2 \\ 19/2 \end{pmatrix} & \to & \frac{1}{2\sqrt{2}}\begin{pmatrix} 19 & 19 \\ 19 & 19 \end{pmatrix} & \searrow & & & \\
 & & & & & & \begin{pmatrix} 3 & 3 \\ 13/2 & 13/2 \\ 3 & 3 \\ 13/2 & 13/2 \end{pmatrix} & \nearrow & \\
-7/\sqrt{2} & \to & \begin{pmatrix} -7/2 \\ -7/2 \end{pmatrix} & \to & \frac{1}{2\sqrt{2}}\begin{pmatrix} -7 & -7 \\ -7 & -7 \end{pmatrix} & \nearrow & & &
\end{array}$$

Figure 4. Reconstruction of the image in the example from the four largest coefficients using the tiling information and the significance map.

6. EXPERIMENTS

Several pictures have been compared using the methods Haar wavelets, adaptive anisotropic wavelet packets, anisotropic local Walsh bases, and adaptive joint space-frequency tilings. In all these experiments the cost function has been the ℓ^1-cost of the coefficients, and in all cases only Haar-Walsh filters have been used. This filters have poor frequency resolution but they are easy to use and we are free from the problems at borders which occur for longer filters. Joint space-frequency method is possible for these filters because of the disjoint supports of the corresponding wavelet functions. The pictures used are gray scale pictures with pixel values between 0 and 255. We thus need 8 bits per pixel in the representation of the original images. The pictures used are Lena, Baboon, Peppers and Boat.

We refer to Lindberg[3] for results on Peppers and Boat. They have all been transformed to best basis wavelet or wavelet packet coefficients using the methods mentioned above. The quality of the reconstructions is measured by peak signal-to-noise ratio (PSNR). The measure PSNR is a good indicators of the quality of the image but if the bit rate is low the visual quality of the image does not always agree with these measures.

The real compression ratio depends on the need of memory for the kept coefficients and for the location of them. The need of memory for the kept coefficients was computed using Shannon's theorem which gives the average bit rate for the coefficients. The need of memory for the location of the kept coefficients was computed from the significance map of size N^2 for wavelets, from the significance map of size $(L+1)^2N^2$ for anisotropic wavelet packets and local Walsh bases and from the compressed tiling information vector and the significance map of size N^2 for the joint space-frequency method. All these vectors, giving the location for the kept coefficients consist of ones and zeros and were run-length encoded before computation of the need of memory. We found that the total need of memory for the significance map and for the tiling information for the joint space-frequency method was about twice the need of memory for the significance map for wavelets. Joint space-frequency tilings gives the same visual quality in the reconstructed images with half the number of coefficients needed for the other methods. When the total need of memory was computed we found that the joint space-frequency method gives the best both visual and measured results for all real images tested in this paper. The reconstructed images of Lena and Baboon are given in Figures 5 and 6, and they show that we have the same visual quality for about half the number of coefficients for joint space-frequency method compared with the other methods. This means that we have to compare the results for this method with the other methods when twice the number of coefficients is used.

Table 1. Compression results for Lena image

Fraction of kept coeffs.	Method	coeff [bits/coeff]	Significance map small [bits/coeff]	Significance map large [bits/coeff]	Tiling info [bits/coeff]	Total bits per coeff	Total bits/pixel
1/32	Haar wavelets	8.32	4.06			12.38	0.39
	Anisotr. WP	8.23		7.95		16.19	0.51
	Local Walsh	8.27		5.79		14.07	0.44
	Joint space/freq.	8.20	4.92		4.66	17.78	0.56
1/64	Joint space/freq.	8.70	5.31		6.94	20.60	0.32

The experiments gave that the need of memory for tiling information for real images can be compressed to 4-7 bits per kept coefficient. When the location of the kept coefficients for Lena from 16384 coefficients (1/16 of all pixels) was given as the significance map in the wavelet packet tree matrix of size 5120×5120 the need of memory was 13.62 bits per kept coefficient. This shall be compared with the need of memory of 8.57 bits per kept coefficient for the method introduced in Lindberg[3] and described here in Section 4. This shows the efficiency of the proposed method for the location information of the chosen bases.

REFERENCES

1. N. N. Bennett, "Fast algorithm for best anisotropic Walsh bases and relatives", *Appl. Comput. Harmon. Anal.* **8**, 86–103, 2000.
2. C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint space- frequency segmentation using balanced wavelet packet trees for least cost image representation", *IEEE Transactions on Image Processing*, **6**, no. 9, 1213–1230, Sept. 1997.
3. M. Lindberg, *Two-dimensional adaptive Haar-Walsh tilings*, Licentiate thesis, Åbo Akademi University, 1999.
4. F. G. Meyer, A. Z. Averbuch, and J. O. Strömberg, "Fast adaptive wavelet packet image compression", *IEEE Transactions on Image Processing*, **9**, 792–800, May 2000.
5. F. G. Meyer and R. R. Coifman, "Brushlets: a tool for directional image analysis and image compression", *Appl. Comput. Harmon. Anal.* **4**, 147 –187. 1997
6. C. M. Thiele and L. F. Villemoes, "A fast algorithm for adapted time-frequency tilings", *Appl. Comput. Harmon. Anal.* **3**, 91–99, 1996.
7. J. L. Walsh, "A closed set of normal orthogonal functions", *Amer. J. Math.* **45**, 5–24, 1923.
8. M. V. Wickerhauser, *Adapted wavelet analysis from theory to software*, A K Peters, Ltd., Wellesley, MA, 1994.

Original 512×512

Haar wavelets 32:1, PSNR 30.0 dB

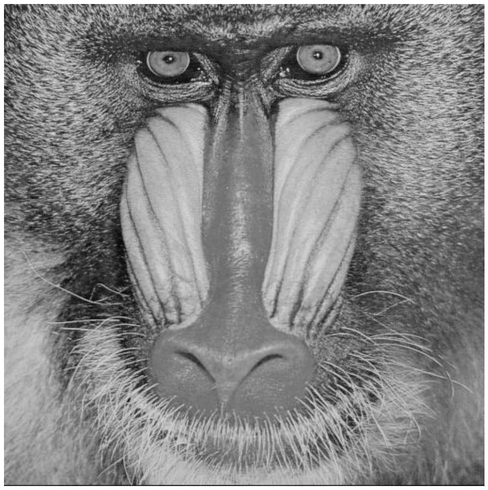
Anisotropic wavelet packets 32:1, PSNR 30.4 dB

Local Walsh bases 32:1, PSNR 29.0 dB

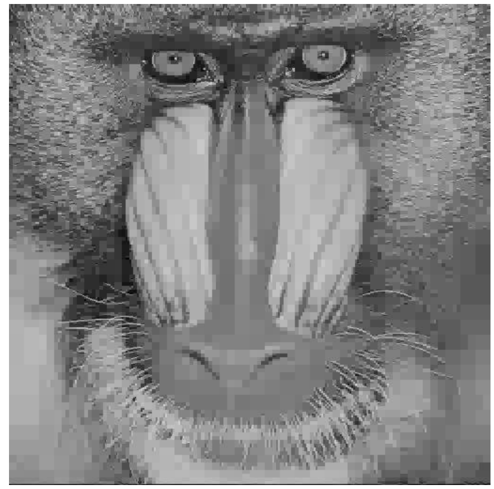
Space-frequency tiling 32:1, PSNR 32.7 dB

Space-frequency tiling 64:1, PSNR 29.8 dB

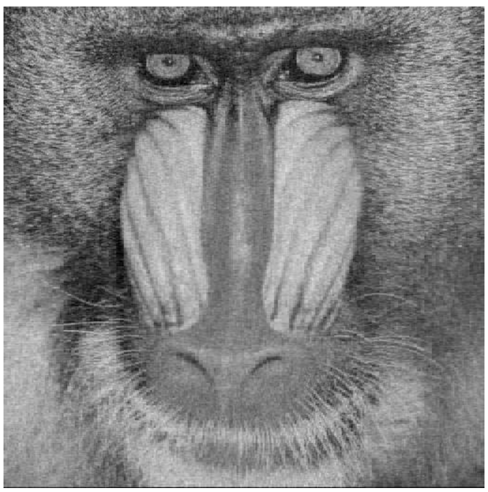
Figure 5. Nonlinear approximation of Lena image, using Haar filters. For bit-rates, see Table 1.



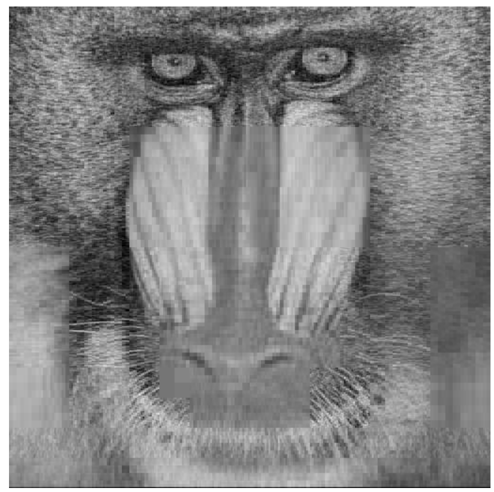
Original 512 × 512



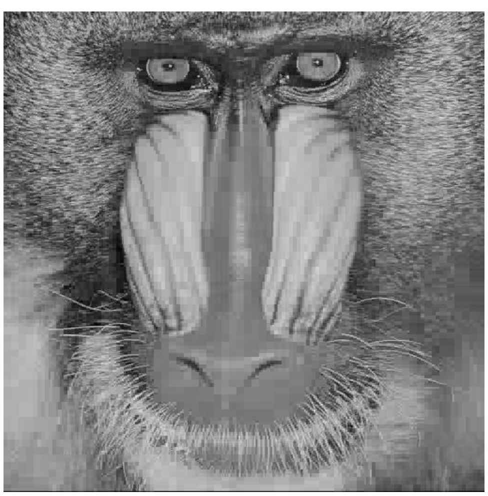
Haar wavelets 32:1, PSNR 22.0 dB



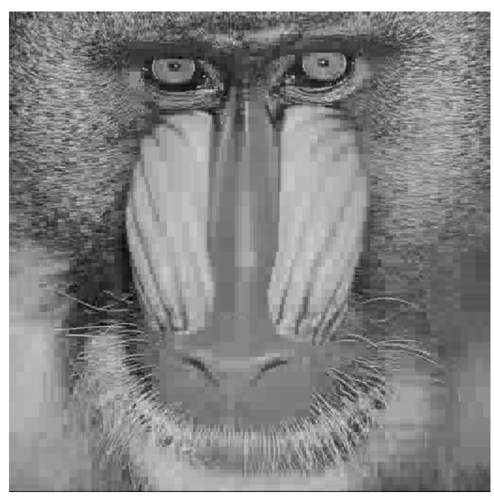
Anisotropic wavelet packets 32:1, PSNR 22.5 dB



Local Walsh bases 32:1, PSNR 21.9 dB



Space-frequency tiling 32:1, PSNR 23.7 dB



Space-frequency tiling 64:1, PSNR 22.1 dB

Figure 6. Nonlinear approximation of Baboon image, using Haar filters.