Image Approximation by Adaptive Tetrolet Transform

Jens Krommweh

Department of Mathematics, University of Duisburg-Essen, Campus Duisburg, 47048 Duisburg, Germany.
jens.krommweh@uni-due.de

Abstract:

In order to get an efficient image representation we introduce a new adaptive Haar wavelet transform, called **Tetrolet Transform**. Tetrolets are Haar-type wavelets whose supports are tetrominoes which are shapes made by connecting four equal-sized squares. The corresponding filter bank algorithm is simple but enormously effective. Numerical results show the strong efficiency of the tetrolet transform for image compression.

1. Introduction

The main task in every kind of image processing is finding an efficient image representation that characterizes the significant image features in a compact form. In the last years a lot of methods have been proposed to improve the treatment with orientated geometric image structures. Curvelets [1], contourlets [2], shearlets [5], and directionlets [10] are wavelet systems with more directional sensitivity than classical tensor product wavelets.

Instead of choosing a priori a basis or a frame one may adapt the function system depending on the local image structures. Wedgelets [3] and bandelets [7] stand for this second class of image representation schemes which is a wide field of further research. Very recent approaches are the grouplets [8] or the EPWT [9] which are based on an averaging in adaptive neighborhoods of data points.

In [6] we have introduced a new adaptive algorithm whose underlying idea is similar to the idea of digital wedgelets where Haar functions on wedge partitions are considered. We divide the image into 4×4 blocks, then we determine in each block a tetromino partition which is adapted to the image geometry in this block. Tetrominoes are shapes made by connecting four equal-sized squares, each joined together with at least one other square along an edge. On these geometric shapes we define Haar-type wavelets, called *tetrolets*, which form a local orthonormal basis. The main advantage of Haar-type wavelets is the lack of pseudo-Gibbs artifacts. The corresponding filter bank algorithm decomposes an image into a compact representation.

The tetrolet transform is also very efficient for compression of real data arrays.

2. The Adaptive Tetrolet Transform

2.1 Definitions and Notations

Let be $I = \{(i,j) : i,j = 0,\dots,N-1\} \subset \mathbb{Z}^2$ the *index set* of a digital image $\mathbf{a} = (a[i,j])_{(i,j)\in I}$ with $N = 2^J, J \in \mathbb{N}$. We determine a *4-neighborhood* of an index $(i,j) \in I$ by $N_4(i,j) := \{(i-1,j),(i+1,j),(i,j-1),(i,j+1)\}$. An index that lies at the boundary has three neighbors, an index at the vertex of the image has two neighbors.
A set $E = \{I_0,\dots,I_r\}, r \in \mathbb{N}$, of subsets $I_\nu \subset I$ is a *disjoint partition* of I if $I_\nu \cap I_\mu = \emptyset$ for $\nu \neq \mu$ and $\bigcup_{\nu=0}^{r} I_\nu = I$.
In this paper we consider disjoint partitions of the index set I that satisfy two conditions for all I_ν:

1. each subset I_ν contains four indices, i.e. $\#I_\nu = 4$,
2. every index of I_ν has a neighbor in I_ν, i.e. $\forall (i,j) \in I_\nu \; \exists (i',j') \in I_\nu : (i',j') \in N_4(i,j)$.

We call such subsets I_ν *tetromino*, since the tiling problem of the square $[0,N)^2$ by shapes called tetrominoes is a well-known problem being closely related to our partitions of the index set $I = \{0,1,\dots,N-1\}^2$. We shortly introduce this tetromino tiling problem in the next subsection.

2.2 Tilings by Tetrominoes

Tetrominoes were introduced by Golomb in [4]. They are shapes formed from a union of four unit squares, each connected by edges, not merely at their corners. The tiling problem with tetrominoes became popular through the famous computer game classic 'Tetris'. Disregarding rotations and reflections there are five different shapes, the so called *free tetrominoes*, see Figure 1.
It is clear that every square $[0,N)^2$ can be covered by tetrominoes if and only if N is even. But the number of different coverings explodes with increasing N. There are 117 solutions for disjoint covering of a 4×4 board with four tetrominoes. As represented in Figure 2, we have 22

Figure 1: The five free tetrominoes.

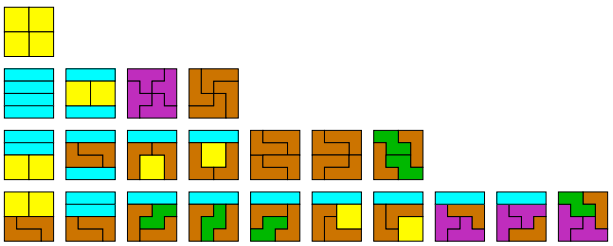

Figure 2: The 22 fundamental forms tiling a 4×4 board. Regarding additionally rotations and reflections there are 117 solutions.

fundamental configurations (disregarding rotations and reflections). One solution (first line) is unaltered by rotations and reflections, four solutions (second line) give a second version applying the isometries. Seven forms can occur in four orientations (third line), and ten asymmetric cases in eight directions (last line).

2.3 The Idea of Tetrolets

In the two-dimensional classical Haar case, the low-pass filter and the high-pass filters are just given by the averaging sum and the averaging differences of each four pixel values which are arranged in a 2×2 square, i.e., with $I_{i,j} = \{(2i, 2j), (2i+1, 2j), (2i, 2j+1), (2i+1, 2j+1)\}$ for $i, j = 0, 1, \ldots, \frac{N}{2} - 1$, we have a dyadic partition $E = \{I_{0,0}, \ldots, I_{\frac{N}{2}-1,\frac{N}{2}-1}\}$ of the image index set I. Let L be a bijective mapping which maps the four pixel pairs (i, j) to the scalar set $\{0, 1, 2, 3\}$, that means it brings the pixels into a unique order. Then we can determine the low-pass part $\mathbf{a}^1 = (a^1[i, j])_{i,j=0}^{\frac{N}{2}-1}$ as well as the three high-pass parts $\mathbf{w}_l^1 = (w_l^1[i, j])_{i,j=0}^{\frac{N}{2}-1}$ for $l = 1, 2, 3$ with

$$a^1[i, j] = \sum_{(i',j') \in I_{i,j}} \epsilon[0, L(i', j')]\, a[i', j'] \quad (1)$$

$$w_l^1[i, j] = \sum_{(i',j') \in I_{i,j}} \epsilon[l, L(i', j')]\, a[i', j'], \quad (2)$$

where the coefficients $\epsilon[l, m], l, m = 0, \ldots, 3$, are entries from the Haar wavelet transform matrix

$$W := (\epsilon[l, m])_{l,m=0}^{3} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}. \quad (3)$$

Obviously, the fixed blocking by the dyadic squares $I_{i,j}$ is very inefficient because the local structures of an image are disregarded. Our idea is, to allow more general partitions such that the local image geometry is taken into account. Namely, we use tetromino partitions. As described in the previous subsection we shall restrict us to 4×4 blocks. This leads to a third condition for the desired disjoint partition E of the index set I introduced in Section 2.1:

3. Each 4×4 square $Q_{i,j} := \{4i, \ldots, 4i + 3\} \times \{4j, \ldots, 4j + 3\}, i, j = 0, 1, \ldots, \frac{N}{4} - 1$, is covered by four subsets (tetrominoes) $I_0, \ldots, I_3$.

In other words, we first divide the index set I of an image $\mathbf{a}$ into $\frac{N^2}{16}$ squares $Q_{i,j}$ and then we consider the admissible tetromino partitions there. Among the 117 solutions we compute an optimal partition in each image block such that the wavelet coefficients defined on the tetrominoes have minimal l^1-norm.

3. Detailed Description of the Algorithm

The rough structure of the tetrolet filter bank algorithm is described in Table 1.

Adaptive Tetrolet Decomposition Algorithm
Input: Image $\mathbf{a} = (a[i, j])_{i,j=0}^{N-1}$ with $N = 2^J, J \in \mathbb{N}$.
1. Divide the image into 4×4 blocks.
2. Find in each block the sparsest tetrolet representation.
3. Rearrange the low- and high-pass coefficients of each block into a 2×2 block.
4. Store the tetrolet coefficients (high-pass part).
5. Apply step 1 to 4 to the low-pass image.
Output: Decomposed image $\tilde{\mathbf{a}}$.

Table 1: Adaptive tetrolet decomposition algorithm.

Going into detail our main attention shall be turned to step 2 of the algorithm where the adaptivity comes into play. We start with the input image $\mathbf{a}^0 = (a[i, j])_{i,j=0}^{N-1}$ with $N = 2^J, J \in \mathbb{N}$. In the rth-level, $r = 1, \ldots, J - 1$, we apply the following computations.

1. Divide the low-pass image $\mathbf{a}^{r-1}$ into blocks $Q_{i,j}$ of size 4×4, $i, j = 0, \ldots, \frac{N}{4^r} - 1$.

2. In each block $Q_{i,j}$ we compute analogously to (1) and (2) the pixel averages for every admissible tetromino covering $c = 1, \ldots, 117$ by

$$a^{r,(c)}[s] = \sum_{(m,n) \in I_s^{(c)}} \epsilon[0, L(m, n)]\, a^{r-1}[m, n],$$

as well as the three high-pass parts for $l = 1, 2, 3$

$$w_l^{r,(c)}[s] = \sum_{(m,n) \in I_s^{(c)}} \epsilon[l, L(m, n)]\, a^{r-1}[m, n],$$

$s = 0, \ldots, 3$, where the coefficients are given in (3) and L is the mapping mentioned above. Then we choose the covering c^* such that the l^1-norm of the tetrolet coefficients becomes minimal

$$c^* = \arg\min_c \sum_{l=1}^{3} \sum_{s=0}^{3} |w_l^{r,(c)}[s]|. \quad (4)$$

Hence, for every block $Q_{i,j}$ we get an optimal tetrolet decomposition $[\mathbf{a}^{r,(c^*)}, \mathbf{w}_1^{r,(c^*)}, \mathbf{w}_2^{r,(c^*)}, \mathbf{w}_3^{r,(c^*)}]$. By doing this, the local structure of the image block is adapted. The best configuration c^* is a covering whose tetrominoes do not intersect an important structure like an edge in the image $\mathbf{a}^{r-1}$. Because the tetrolet coefficients become as minimal as possible a sparse image representation will be obtained. We have to store for each block $Q_{i,j}$ which covering c^* has been chosen, since this information is necessary for reconstruction.

3. In order to be able to apply further levels of the tetrolet decomposition algorithm, we rearrange the entries of the vectors $\mathbf{a}^{r,(c^*)}$ and $\mathbf{w}_l^{r,(c^*)}$ into 2×2 matrices,

$$\mathbf{a}^r_{|_{Q_{i,j}}} = \begin{pmatrix} a^{r,(c^*)}[0] & a^{r,(c^*)}[2] \\ a^{r,(c^*)}[1] & a^{r,(c^*)}[3] \end{pmatrix},$$

and in the same way $\mathbf{w}^r_{l|_{Q_{i,j}}}$ for $l = 1, 2, 3$.

4. After finding a sparse representation in every block $Q_{i,j}$ for $i, j = 0, \ldots, \frac{N}{4^r} - 1$, we store (as usually done) the low-pass matrix $\mathbf{a}^r$ and the high-pass matrices $\mathbf{w}_l^r, l = 1, 2, 3$, replacing the low-pass image $\mathbf{a}^{r-1}$ by the matrix

$$\begin{pmatrix} \mathbf{a}^r & \mathbf{w}_2^r \\ \mathbf{w}_1^r & \mathbf{w}_3^r \end{pmatrix}.$$

After a suitable number of decomposition steps, one can apply a shrinkage to the tetrolet coefficients in order to get a sparse image representation.

4. An Orthonormal Basis of Tetrolets

We describe the discrete basis functions which correspond to the above algorithm. Remember that the digital image $\mathbf{a} = (a[i,j])_{(i,j)\in I}$ is a subset of $l_2(\mathbb{Z}^2)$. For any tetromino I_ν of I we define the discrete functions

$$\phi_{I_\nu}[m,n] := \begin{cases} 1/2, & (m,n) \in I_\nu, \\ 0, & \text{else}, \end{cases}$$

$$\psi^l_{I_\nu}[m,n] := \begin{cases} \epsilon[l, L(m,n)], & (m,n) \in I_\nu, \\ 0, & \text{else}. \end{cases}$$

Due to the underlying tetromino support, we call ϕ_{I_ν} and $\psi^l_{I_\nu}$ *tetrolets*. As a straightforward consequence of the orthogonality of the standard 2D Haar basis functions and the disjoint partition of the discrete space by the tetromino supports, we have the following essential statement.

Theorem 1 *For every admissible covering $\{I_0, I_1, I_2, I_3\}$ of a 4×4 square $Q \subset \mathbb{Z}^2$ the tetrolet system*

$$\{\phi_{I_\nu} : \nu = 0, 1, 2, 3\} \cup \{\psi^l_{I_\nu} : \nu = 0, 1, 2, 3; l = 1, 2, 3\}$$

is an orthonormal basis of $l^2(Q)$.

5. Cost of Adaptivity: Modified Tetrolet Transform

We will address the costs of storing additional adaptivity information. Our observations will lead to some relaxed versions of the tetrolet transform in order to reduce these costs.

It is well known that a vector of length N and with entropy E can be stored with $N \cdot E$ bits. Hence, the entropy describes the required bits per pixel (bpp) and is an appropriate measure for the quality of compression.

In the following, we propose three methods of entropy reduction in order to reduce the adaptivity costs. An application of these modified transforms as well as of combinations of them is given in the last section.

The simplest approach of entropy reduction is reduction of the symbol alphabet. The tetrolet transform uses the alphabet $\{1, \ldots, 117\}$ for the chosen covering in each image block. If we restrict ourselves to 16 essential configurations that feature different directions we considerably reduce the entropy as well as the computation time.

A second approach to reduce the entropy is to change the distribution of the symbols. Relaxing the tetrolet transform we could ensure that only very few tilings are preferred. Hence, we allow the choice of an *almost* optimal covering c^* in (4) in order to get a tiling which is already frequently chosen. More precisely, we replace (4) by the two steps:

1. Find the set of almost optimal configurations that satisfy

$$\sum_{l=1}^{3}\sum_{s=0}^{3} |w_l^{r,(c)}[s]| \leqslant \min_c \sum_{l=1}^{3}\sum_{s=0}^{3} |w_l^{r,(c)}[s]| + \theta$$

with a predetermined tolerance parameter θ.

2. Among these tilings choose the covering c which is chosen most frequently in the previous image blocks.

Using an appropriate relaxing parameter θ, we achieve a satisfactory balance between low entropy (low adaptivity costs) and minimal tetrolet coefficients.

The third method also reduces the entropy by optimization of the tiling distribution. After an application of an edge detector we use the classical Haar wavelet transform inside flat image regions. In the image blocks that contain edges we make use of the strong adaptivity of the proposed tetrolet transform.

More details of the modified versions can be found in [6].

6. Numerical Experiments

We apply a complete wavelet decomposition of an image and use a shrinkage with global hard-thresholding.

The detail 'monarch' image in Figure 3 shows the enormous efficiency in handling with several directional edges due to the high adaptivity. It can be well noticed that the tetrolet transformation gives excellent results for piecewise constant images. Though the tetrolets are not continuous the approximation of the 'cameraman' image in Figure 4 illustrates that even for natural images the tetrolet filter bank outperforms the tensor product wavelets with the biorthogonal 9-7 filter bank, since no pseudo-Gibbs phenomena occur. This confirms the fact already noticed with wedgelets [3] and bandelets [7]: While nonadaptive methods need smooth wavelets for excellent results, well constructed adaptive methods need not. See [6] for more numerical examples.

Considering the adaptivity costs we compare the standard tetrolet transform with its modified versions. Of course, reduction of adaptivity cost produces a loss of approximation quality. Hence, a satisfactory balance is necessary.

For a rough estimation of the complete storage costs of the compressed image with N^2 pixels we apply a simplified scheme

$$cost_{full} = cost_W + cost_P + cost_A,$$

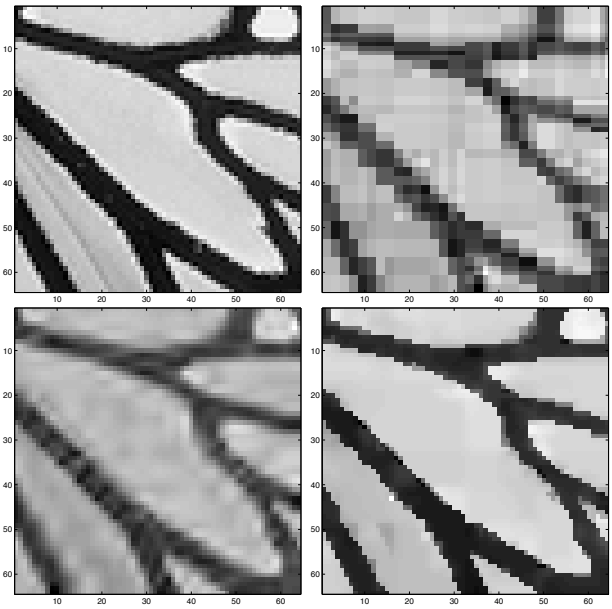

Figure 3: Approximation with 256 coefficients. (a) Input, (b) classical Haar, PSNR 18.98, (c) Biorthogonal 9-7, PSNR 21.78, (d) Tetrolets, PSNR 24.43.

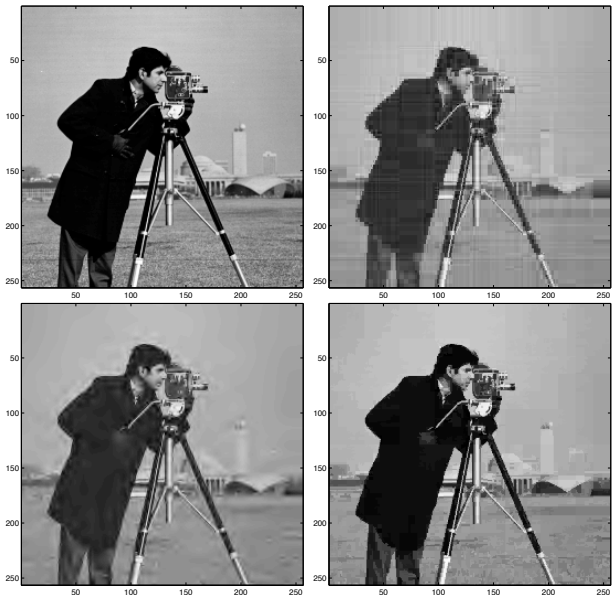

Figure 4: Approximation with 2048 coefficients. (a) Input, (b) classical Haar, PSNR 25.47, (c) Biorthogonal 9-7, PSNR 27.26, (d) Tetrolets, PSNR 29.17.

where $cost_W = 16 \cdot M/N^2$ are the costs in bpp of storing M non-zero wavelet coefficients with 16 bits. The term $cost_P$ gives the cost for coding the position of these M coefficients by $-\frac{M}{N^2}\log_2(\frac{M}{N^2}) - \frac{N^2-M}{N^2}\log_2(\frac{N^2-M}{N^2})$. The third component appearing only with the tetrolet transform contains the cost of adaptivity, $cost_A = E \cdot R/N^2$, for R adaptivity values and the entropy E previously discussed.

Table 2 presents some results for the monarch detail image (Fig. 3) where different versions of the tetrolet transform are compared with the tensor product wavelet transformation regarding to quality and storage costs. We have tried to balance the modified tetrolet transform such that the full costs are in the same scale as with the 9-7 filter. For the relaxed versions we have used the parameter $\theta = 25$.

	coeff	PSNR	entropy	$cost_{full}$
Tensor Haar	300	19.58	-	1.55
Tensor 9-7 filter	300	22.62	-	1.55
Tetrolet	256	24.43	0.53	1.86
Tetro 16	256	23.56	0.30	1.64
Tetro rel	256	24.51	0.32	1.66
Tetro edge	256	24.24	0.43	1.77
Tetro 16 edge rel	256	23.48	0.21	1.55

Table 2: Comparison between tensor wavelet transforms and the different versions of the tetrolet transform regarding quality (PSNR) and storage cost ($cost_{full}$ in bpp).

7. Acknowledgments

The research is funded by the project PL 170/11-1 of the Deutsche Forschungsgemeinschaft (DFG). This is gratefully acknowledged.

References:

[1] E.J. Candes and D.L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Communications on Pure and Applied Mathematics*, 57(2):219–266, 2004.

[2] M.N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12):2091–2106, 2005.

[3] D.L. Donoho. Wedgelets: Nearly-minimax estimation of edges. *Annals of Statistics*, 27(3):859–897, 1999.

[4] S.W. Golomb. *Polyominoes*. Princeton University Press, 1994.

[5] K. Guo and D. Labate. Optimally sparse multidimensional representation using shearlets. *SIAM Journal on Mathematical Analysis*, 39(1):298–318, 2007.

[6] Jens Krommweh. Tetrolet transform: A new adaptive Haar wavelet algorithm for sparse image representation. 2009.

[7] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *IEEE Transactions on Image Processing*, 14(4):423–438, 2005.

[8] S. Mallat. Geometrical grouplets. *Applied and Compu-tational Harmonic Analysis*, 26(2):161–180, 2009.

[9] G. Plonka. Easy path wavelet transform: A new adaptive wavelet transform for sparse representation of two-dimensional data. *Multiscale Modeling and Simulation*, 7(3):1474–1496, 2009.

[10] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P.L. Dragotti. Directionlets: Anisotropic multidirectional representation with separable filtering. *IEEE Transactions on Image Processing*, 15(7):1916–1933, 2006.