Tetrolet Transform: A New Adaptive Haar Wavelet Algorithm for Sparse Image Representation

Jens Krommweh[a]

[a]*Department of Mathematics, University of Duisburg-Essen, Campus Duisburg, 47048 Duisburg, Germany*

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 fast filter bank algorithm is simple but very effective. In every level of the filter bank algorithm we divide the low-pass image into 4×4 blocks. Then in each block we determine a local tetrolet basis which is adapted to the image geometry in this block. An analysis of the adaptivity costs leads to modified versions of our method. Numerical results show the strong efficiency of the tetrolet transform for image approximation.

Key words: adpative wavelet transform, directional wavelets, Haar-type wavelets, locally orthonormal wavelet basis, tetromino tiling, image approximation, data compression, sparse representation
2000 MSC: 65T60, 42C40, 68U10, 94A08

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. Here, the 2D discrete wavelet transform (DWT) is one of the most important tools. Conventionally, the 2D DWT is a separable construction, based on the 1D wavelet transformation which is independently applied to the rows and columns of an image. Therefore, the horizontal and vertical directions are preferred, and the DWT fails to achieve optimal results with images that contain geometric structures in other directions.

In the last years a lot of methods have been proposed to improve the treatment of orientated geometric image structures. Curvelets [2], contourlets [6], shearlets [10], and directionlets [17] are wavelet systems with more directional sensitivity.

Beside these non-adaptive function systems one may also consider adaptive image representation schemes: Instead of choosing a priori a basis or a frame one may try to adapt the function system depending on the local image structures. Wedgelets [5] and bandelets [15] are examples of such adaptive methods which offer a wide field of further research. Very recent approaches are the grouplets [14] or the easy path wavelet transform (EPWT) [16] which are based on an averaging in adaptive neighborhoods of data points. While in the grouplet context neighborhood is defined

Email address: jens.krommweh@uni-due.de (Jens Krommweh)

by an association field, the EPWT uses neighborhoods on a path through all data points. Another kind of promising adaptive approach is the usage of directional lifting-based wavelets [3], [4].

In this paper, we introduce a new adaptive algorithm whose underlying idea is simple but very fast and effective. The proposed method is especially designed for sparse image approximation due to the non-redundance of the basis functions. The construction is similar to the idea of digital wedgelets [8], 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. Originally, tetrominoes were introduced by Golomb [9], and they became popular through the famous computer game classic 'Tetris' [1]. On these geometric shapes we define Haar-type wavelets, called *tetrolets*, which form a local orthonormal basis. The non-redundance leads to a critically sampled filter bank which decomposes an image into a sparse representation. In order to obtain an image approximation, one can apply a suitable shrinkage procedure to the tetrolet coefficients and reconstruct the image.

It should be mentioned that the tetrolet transform is not confined to image processing. Quite the contrary, the method is very efficient for compression of real data arrays.

The paper is organized as follows. In Section 2 we present the rough idea of the tetrolet transformation, in Section 3 a detailed description of the corresponding fast and adaptive algorithm is given. Then in the next section, we show the important fact that the tetrolets form a local orthonormal basis. Section 5 is devoted to the analysis of the adaptivity cost which results in modified versions of the tetrolet transform. Finally, in the last section we present some numerical results applying the tetrolet transform to image approximation.

2. The Tetrolet Transform: A New Locally Adaptive Algorithm

2.1. Definitions and Notations

In order to explain the idea of the tetrolet transform we first need some definitions and notations. For simplicity we restrict our considerations to two-dimensional square data sets. Let $I = \{(i,j) : i,j = 0, \ldots, N-1\} \subset \mathbb{Z}^2$ be 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. For our analysis we use a one-dimensional index set $J(I)$ by taking the bijective mapping $J : I \to \{0, 1, \ldots, N^2 - 1\}$ with $J((i,j)) := jN + i$.

A set $E = \{I_0, \ldots, I_r\}, r \in \mathbb{N}$, 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 E of the index set I that satisfy two conditions:

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

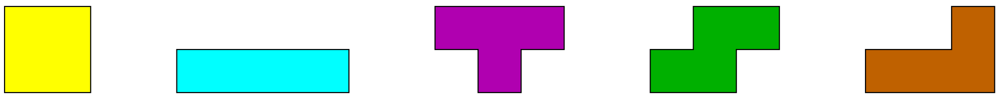

Figure 1: The five free tetrominoes. O-I-T-S-L-tetromino.

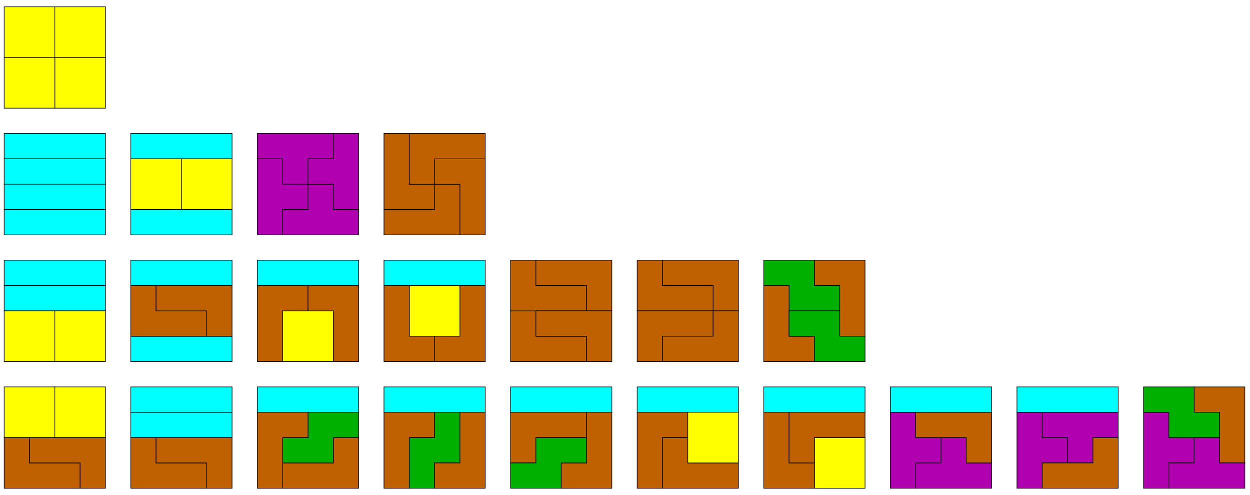

Figure 2: The 22 fundamental forms tiling a 4×4 board. Regarding additionally rotations and reflections there are 117 solutions.

We call such subsets I_ν *tetromino*, since the tiling problem of the square $[0, N)^2$ by shapes called tetrominoes [9] is a well-known problem being closely related to our partitions of the index set $I = \{0, 1, \dots, N-1\}^2$. We shortly recall this tetromino tiling problem in the next subsection.

For a simple one-dimensional indexing of the four elements in one tetromino subset I_ν, we apply the bijective mapping J as follows. For $I_\nu = \{(i_1, j_1), (i_2, j_2), (i_3, j_3), (i_4, j_4)\}$ let $L : I_\nu \to \{0, 1, 2, 3\}$ be given by the rule that we order the values $J(i_1, j_1), \dots, J(i_4, j_4)$ by size and map them to $\{0, 1, 2, 3\}$ such that the smallest index is identified with 0 and the largest with 3.

2.2. Tilings by Tetrominoes

Tetrominoes were introduced by Golomb in [9]. 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' [1]. Disregarding rotations and reflections there are five different shapes, the so called *free tetrominoes*, see Figure 1.

Taking the isometries into account, it is clear that every square $[0, N)^2$ can be covered by tetrominoes if and only if N is even. In 1937, Larsson showed that there are 117 solutions for disjoint covering of a 4×4 board with four tetrominoes [13]. For an 8×8 board we compute $117^4 > 10^8$ as a rough lower bound of possible tilings. Thus, in order to handle the number of solutions, it will be reasonable to restrict ourselves to an image partition into 4×4 squares.

As represented in Figure 2, we have 22 fundamental solutions in the 4×4 board (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). Further studies with tetrominoes can be found in [11].

2.3. The Idea of Tetrolets

The two-dimensional classical Haar wavelet decomposition leads to a special tetromino partition. Introducing the discrete tetrolet transformation, we recall the conventional Haar case in a notation consistent to the following tetrolet idea.

In the Haar filter bank, 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. More precisely, 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 the bijective mapping mentioned above which maps the four pixel pairs of $I_{i,j}$ to the set $\{0, 1, 2, 3\}$, i.e., it brings the pixels into a unique order.

Then we can determine the low-pass part

$$\mathbf{a}^1 = (a^1[i,j])_{i,j=0}^{\frac{N}{2}-1} \quad \text{with} \quad a^1[i,j] = \sum_{(i',j') \in I_{i,j}} \epsilon[0, L(i',j')]\; a[i',j'] \tag{2.1}$$

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

$$\mathbf{w}_l^1 = (w_l^1[i,j])_{i,j=0}^{\frac{N}{2}-1} \quad \text{with} \quad w_l^1[i,j] = \sum_{(i',j') \in I_{i,j}} \epsilon[l, L(i',j')]\; a[i',j'], \tag{2.2}$$

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

$$W := (\epsilon[l,m])_{l,m=0}^{3} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}. \tag{2.3}$$

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 divide the image index set into 4×4 blocks. Then instead of using the classical Haar wavelet transform, which corresponds to a partition of the 4×4 block into squares (as in the first line of Figure 2), we compute the 'optimal' partition of the block into four tetrominoes according to the geometry of the image. In the following detailed description of the algorithm, we will explain more precisely what 'optimal' means, namely that the wavelet coefficients defined on the tetrominoes have minimal l^1-norm.

3. Detailed Description of the Tetrolet Filter Bank Algorithm

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

Let us now go into detail considering separately each step of the tetrolet decomposition algorithm. Of course, our main attention shall be turned to step 2 of the algorithm. This is the step where the adaptivity comes into play.

We start with the input image $\mathbf{a}^0 = (a[i,j])_{i,j=0}^{N-1}$ with $N = 2^J, J \in \mathbb{N}$. Then we will be able to apply $J-1$ levels of the tetrolet transform. In the rth-level, $r = 1, \ldots, J-1$, we do the following computations.

```
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 $4 \times 4$ blocks.
    2. Find the sparsest tetrolet representation in each block.
    3. Rearrange the low- and high-pass coefficients of each block into a $2 \times 2$ 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.

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 consider the 117 admissible tetromino coverings $c = 1, \ldots, 117$. For each tiling c we apply a Haar wavelet transform to the four tetromino subsets $I_s^{(c)}$, $s = 0, 1, 2, 3$. In this way we obtain for each tiling c four low-pass coefficients and 12 tetrolet coefficients. More precisely, in $Q_{i,j}$ we compute analogously to (2.1) and (2.2) the pixel averages for every admissible tetromino configuration $c = 1, \ldots, 117$ by

$$\mathbf{a}^{r,(c)} = (a^{r,(c)}[s])_{s=0}^{3} \quad \text{with} \quad a^{r,(c)}[s] = \sum_{(m,n) \in I_s^{(c)}} \epsilon[0, L(m,n)]\, a^{r-1}[m,n], \tag{3.1}$$

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

$$\mathbf{w}_l^{r,(c)} = (w_l^{r,(c)}[s])_{s=0}^{3} \quad \text{with} \quad w_l^{r,(c)}[s] = \sum_{(m,n) \in I_s^{(c)}} \epsilon[l, L(m,n)]\, a^{r-1}[m,n], \tag{3.2}$$

where the coefficients $\epsilon[l, L(m,n)]$ are given in (2.3) and where L is the bijective mapping mentioned in subsection 2.1 relating the four index pairs (m,n) of $I_s^{(c)}$ with the values 0, 1, 2, and 3 in descending order. That means, by the one-dimensional indexing $J(m,n)$ the smallest index is identified with the value 0, while the largest with 3.
Then we choose the covering c^* such that the l^1-norm of the 12 tetrolet coefficients becomes minimal

$$c^* = \arg\min_c \sum_{l=1}^{3} \|\mathbf{w}_l^{r,(c)}\|_1 = \arg\min_c \sum_{l=1}^{3} \sum_{s=0}^{3} |w_l^{r,(c)}[s]|. \tag{3.3}$$

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 covering 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. For each block $Q_{i,j}$ we have to store the covering c^* that has been chosen, since this information is necessary for reconstruction. If the optimal covering is not unique, then we take the tiling c^* that has already been chosen most frequently in the previous blocks. Thus, the coding of the used coverings becomes cheaper.
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 using a reshape function

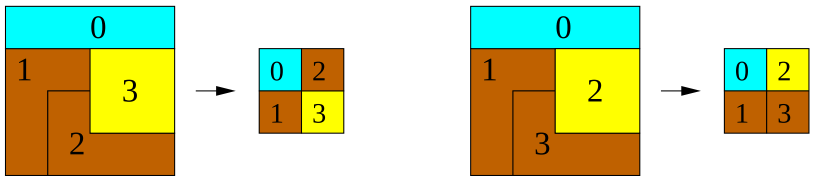

Figure 3: Example of labeling tetrominoes with corresponding low-pass image block. (a) Bad order (ten deviations from square case), (b) best order (eight deviations).

R,

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

and in the same way $\mathbf{w}^r_{l|Q_{i,j}} = R(\mathbf{w}_l^{r,(c^*)})$ for $l = 1, 2, 3$, see Figure 3. For an efficient representation in the next level, a suitable arrangement of the low-pass values is essential. That means, the order of labeling the tetrominoes of c^* in each block $Q_{i,j}$ by $s = 0, 1, 2,$ and 3 is very important. The labeling should be done in a way, such that the geometry of the tiling is suitably mapped to $\left(\begin{smallmatrix} 0 & 2 \\ 1 & 3 \end{smallmatrix}\right)$. Therefore we label the four shapes of the chosen partition c^* by comparing with the square case. Among the 24 possibilities to label the four tetrominoes, the numbering with the highest correlation with the Haar partition is preferred. See an illustration in Figure 3: For comparison of different labeling of the four tetrominoes, we have computed the number of deviations from the Haar wavelet tiling, i.e. we count the number of small squares in a block $Q_{i,j}$, where the label differs from the label of these squares in the Haar wavelet tiling. We apply the order with minimal deviations. This optimal order needs not to be unique. The labeling in Figure 3(a) would lead to a distorted low-pass image, while 3(b) shows a reasonable order.

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 $\mathbf{a}^r = \left(\mathbf{a}^r_{|Q_{i,j}}\right)_{i,j=0}^{\frac{N}{4^r}-1}$ and the high-pass matrices $\mathbf{w}^r_l = \left(\mathbf{w}^r_{l|Q_{i,j}}\right)_{i,j=0}^{\frac{N}{4^r}-1}$, $l = 1, 2, 3$, replacing the low-pass image $\mathbf{a}^{r-1}$ by the matrix

$$\begin{pmatrix} \mathbf{a}^r & \mathbf{w}^r_2 \\ \mathbf{w}^r_1 & \mathbf{w}^r_3 \end{pmatrix}.$$

After a suitable number of decomposition steps, we apply a *shrinkage procedure* to the tetrolet coefficients in order to get a sparse image representation. In our experiments in Section 6, we shall use the hard threshold function

$$S_\lambda(x) = \begin{cases} x, & |x| \geq \lambda, \\ 0, & |x| < \lambda. \end{cases}$$

For the reconstruction of the image, we need the low-pass coefficients from the coarsest level and the tetrolet coefficients as usual. Additionally, the information about the respective covering in each level and block is necessary.

Remark 3.1. Note that we can understand the adaptive tetrolet filter bank as a Haar wavelet filter bank with locally permuted pixel values. This interpretation leads to an efficient implementation. More precisely, we put the 16 pixel values of each 4×4 image block $Q_{i,j}$ into a vector $\mathbf{q} = (q_1, \ldots, q_{16})^T$ by stacking the columns of the image block into $\mathbf{q}$. After applying a permutation matrix $P_c \in \mathbb{R}^{16\times 16}$, which rearranges the image values according to an optimal tetromino covering $c \in \{1, \ldots, 117\}$, we compute the low- and high-pass parts from (3.1) and (3.2) by

$$\tilde{\mathbf{q}} = (I_4 \otimes W)\, P_c\, \mathbf{q}, \tag{3.4}$$

where W is the Haar transformation matrix from (2.3), I_4 is the unit matrix of size 4×4, and $\otimes$ is the Kronecker product, i.e. $(I_4 \otimes W) = \mathrm{blockdiag}(W, W, W, W)$. The resulting vector $\tilde{\mathbf{q}} \in \mathbb{R}^{16}$ has the form

$$\tilde{\mathbf{q}} = (a^{r,(c)}[0], w_1^{r,(c)}[0], w_2^{r,(c)}[0], w_3^{r,(c)}[0], \ldots, a^{r,(c)}[3], w_1^{r,(c)}[3], w_2^{r,(c)}[3], w_3^{r,(c)}[3])^T.$$

Example. Consider the 4×4 block of a synthetic image given by

$$\begin{bmatrix} 20 & 20 & 20 & 20 \\ 20 & 160 & 160 & 20 \\ 20 & 160 & 160 & 20 \\ 20 & 20 & 20 & 20 \end{bmatrix}, \tag{3.5}$$

see Figure 4(a). The conventional Haar wavelet transform of the image function has maximal number of non zero wavelet coefficients. Using (2.1) and (2.2) we obtain after one decomposition step

$$\begin{bmatrix} \mathbf{a}^1 & \mathbf{w}_2^1 \\ \mathbf{w}_1^1 & \mathbf{w}_3^1 \end{bmatrix} = \begin{bmatrix} 110 & 110 & -70 & -70 \\ 110 & 110 & 70 & 70 \\ -70 & 70 & 70 & -70 \\ -70 & 70 & -70 & 70 \end{bmatrix}.$$

All 12 wavelet coefficients do not vanish. By contrast, if we take the tetrolet transform, all pixel differences will vanish according to the adaptive covering by tetrominoes in Figure 4(c). Note, that regarding rotations there are four optimal partitions by tetrominoes.

We stack the columns of (3.5) into a vector

$$\mathbf{q} = (20, 20, 20, 20, 20, 160, 160, 20, 20, 160, 160, 20, 20, 20, 20, 20)^T$$

and apply a permutation matrix P_c corresponding to the tetromino covering in Figure 4(c). In our case, it means, that P_c is given by

$$P_c\,(1, \ldots, 16)^T = (1, 5, 9, 13, 2, 3, 4, 8, 6, 7, 10, 11, 12, 14, 15, 16)^T.$$

So, we get indeed a sparse representation $\tilde{\mathbf{q}} = (40, 0, 0, 0, 40, 0, 0, 0, 320, 0, 0, 0, 40, 0, 0, 0)^T$ by (3.4) because all tetrolet coefficients are zero.

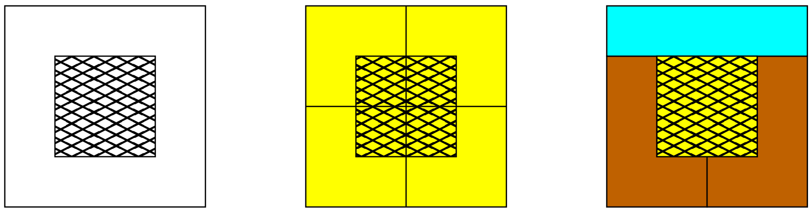

Figure 4: Example of block covering by adaptive tetrominoes. (a) image function, (b) square supports of classical Haar wavelets, (c) supports of adaptive tetrolets.

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

$$\phi_{I_\nu}[m,n] := \begin{cases} 1/2, & (m,n) \in I_\nu, \\ 0, & \text{otherwise}, \end{cases} \qquad \text{and} \qquad \psi^l_{I_\nu}[m,n] := \begin{cases} \epsilon[l, L(m,n)], & (m,n) \in I_\nu, \\ 0, & \text{otherwise}. \end{cases}$$

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 partition of the discrete space by the tetromino supports, we have the following essential statement.

Theorem 4.1. *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*

$$\{\phi_{I_\nu} : \nu = 0,1,2,3\} \cup \{\psi^l_{I_\nu} : \nu = 0,1,2,3; l = 1,2,3\} \tag{4.1}$$

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

5. Cost of Adaptivity: Modified Tetrolet Transform

The tetrolet transform proposed in the previous sections reduces the number of wavelet coefficients compared with the classical tensor product wavelet transform. This improvement has to be payed with the storage of additional information which is not negligible. In this section we shall address this issue in detail. It will lead to some relaxed versions of the tetrolet transform in order to reduce the costs of adaptivity.

In the rth decomposition level of the tetrolet transform we need to store $\frac{N^2}{4^{r+1}}$ covering values c, and therefore after J levels one has $\frac{N^2}{12}(1 - \frac{1}{4^J})$ values. For a complete decomposition, i.e. for $J = \log_2(N) - 1$ we have to store $(N^2 - 4)/12$ values additionally.

It is well-known that a vector of length N and with entropy E can be stored with $N \cdot E$ bits, where the entropy

$$E = -\sum_{i=1}^{n} p(x_i) \log_2(p(x_i)) \tag{5.1}$$

describes the required bits per pixel (bpp) and is an appropriate measure for the quality of compression. The entropy (5.1) can be interpreted as the expected length of a binary code over all the symbols from the given alphabet $A = \{x_1, \dots, x_n\}$. Here, $p(x_i)$ describes the relative occurrence of the symbol x_i in the data set.

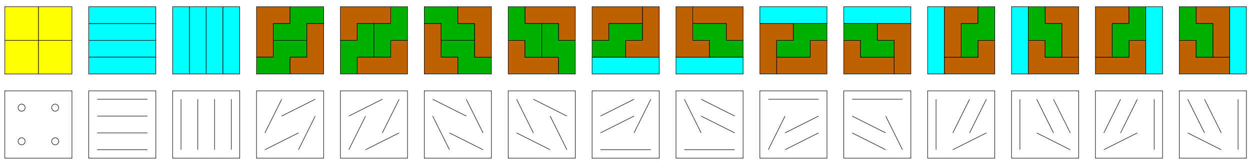
Figure 5: The a priori selected 16 tilings with different directions.

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.

(a) The simplest approach of entropy reduction is a reduction of the alphabet A. The standard tetrolet transform selects the optimal covering in each image block from the alphabet $\{1,\ldots,117\}$. Due to the similarity of some tetromino configurations (see Figure 2) we can disregard certain tilings. We restrict ourselves to 16 suitable configurations. Of course, the optimal choice of these configurations depends on the image. But we can choose a collection of tilings a priori by selecting configurations that feature different directions, see Figure 5. Furthermore, restricting to 16 configurations implicates an essential gain in computation time.

(b) A second approach to reduce the entropy is to manipulate the propabilities $p(x_i)$ in (5.1) by changing their distribution. Relaxing the tetrolet transform we can demand that only very few tilings are preferred. Hence, we allow the choice of an *almost* optimal covering c^* in (3.3) in order to get a tiling which is already frequently choosen. More precisely, we replace (3.3) by the two steps:

   1. Find the set $A' \subset A$ of almost optimal configurations c that satisfy

   $$\sum_{l=1}^{3} \|\mathbf{w}_l^{r,(c)}\|_1 \;\le\; \min_{c\in A} \sum_{l=1}^{3} \|\mathbf{w}_l^{r,(c)}\|_1 + \theta$$

   with a predetermined tolerance parameter θ.

   2. Among these tilings take the covering $c^* \in A'$ 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.

(c) 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 image blocks that contain edges we make use of the strong adaptivity of the proposed tetrolet transform. This method leads to a huge amount of the square tiling corresponding to the Haar wavelet case.

6. Numerical experiments

As already mentioned in the beginning, our method is also very efficient for compression of real data arrays, but in the following we consider digital images.

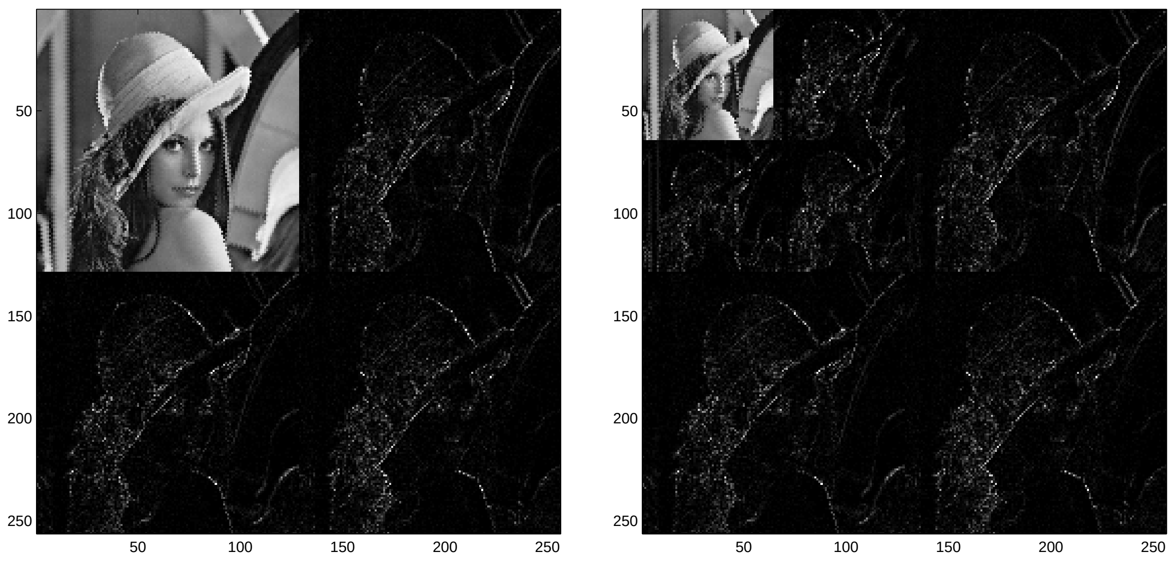

Figure 6: Transform coefficients after the first two levels of the tetrolet decomposition filter bank.

While wavelet frames are useful for denoising (because redundancy information gives more hope to reconstruct the origin data), for image compression a basis is desirable. The tetrolets form a basis of a subspace of $l^2(\mathbb{Z}^2)$ and lead therefore to a sparse image representation using the efficient filter bank algorithm described above. An example of the transform coefficients of the one low-pass part and the three high-pass bands is displayed by the classical tree structure in Figure 6. High-pass coefficients of large amplitude are shown in white.

6.1. Standard tetrolet transform

We apply a complete wavelet decomposition of an image and use a wavelet shrinkage with global hard-thresholding choosing the threshold λ such that a certain number of largest wavelet coefficients is retained.

The 256×256 synthetic image in Figure 7 shows that the tetrolet transformation gives excellent results for piecewise constant images. With only 512 coefficients after thresholding, the reconstructed image achieves a remarkable PSNR of 38.47 dB because the orientated edges are well adapted. Though the Haar-like tetrolets are not continuous the Figures 8–11 illustrate that even for natural images the tetrolet filter bank outperforms the tensor product wavelets with the biorthogonal 9-7 filter bank. This confirms the fact already noticed with wedgelets [5]: While nonadaptive methods need smooth wavelets for excellent results, well constructed adaptive methods need not. For visual purposes the images are slightly smoothed with a bilateral filter in a post-processing step.

We compare our method with the traditional tensor product Haar wavelet transform and the 9-7 biorthogonal filter. Furthermore, we consider two directional wavelet transforms, the contourlet transform [6] and the discrete curvelet transform [2]. Both are based on directional wavelet frames and their redundancy reduces the compact image representation. For the computation of the contourlet and the curvelet transform we have used the MATLAB toolboxes from `www.ifp.uiuc.edu/~minhdo/software/` and `www.curvelet.org`. The detailed commented MATLAB codes of the tetrolet transform are available at our homepage `www.uni-due.de/mathematik/krommweh/`.

The approximation results are summarized in Table 2. The 256×256 images 'cameraman' and 'pepper' were approximated with 2048 coefficients, the 128×128 'barbara' detail image with 512 coefficients, and the 64×64 'monarch' detail image with 256 coefficients. In particular, in the monarch image, the enormous efficiency in handling with several directional edges due to the high adaptivity can be well noticed. Even for the textures in Figure 11 the tetrolet transform offers satisfactory results. Of course, this kind of texture is excellently approximated by contourlets.

	synthetic (Fig. 7)	cameraman (Fig. 8)	pepper (Fig. 9)	monarch (Fig. 10)	barbara (Fig. 11)
Coefficients after shrinkage	512	2048	2048	256	512
Tensor Haar	28.13	25.47	26.11	18.98	19.69
Tensor biorthogonal 9-7	30.23	27.26	28.96	21.78	20.49
Tetrolets	38.47	29.17	30.00	24.43	21.05
Contourlets	30.21	26.07	27.70	21.00	23.16
Curvelets	26.02	21.91	22.77	12.85	18.37

Table 2: PSNR values (in dB) with approximation.

6.2. Modified tetrolet transform

Considering the adaptivity costs we compare the standard tetrolet transform and the modified versions. A complete tetrolet decomposition of the 256×256 cameraman image yields 5461 adaptivity values $c \in \{1, \ldots, 117\}$. The distribution of these values with the standard tetrolet transformation is shown in Figure 12(a); the entropy (5.1) of the distribution vector is 0.56 bpp. According to the second modification mentioned above we relax the tetrolet transform with $\theta = 25$, the resulting histogram is given in Figure 12(b), the corresponding entropy is reduced to 0.25 bpp. This is a remarkable improvement under the small loss of quality (instead of 29.17 dB PSNR we have only 28.91 dB PSNR). Of course, reduction of adaptivity cost produces a loss of approximation quality. Hence, a satisfactory balance is necessary. Table 3 presents some results applying the modified versions of the tetrolet transform proposed in the previous section to the monarch detail and the cameraman image. The three modifications (called 'Tetrolet 16', 'Tetrolet rel', 'Tetrolet edge') and a combination of them ('Tetrolet 16 rel edge') 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. Then, one can observe that the tetrolets lead to slightly higher PSNR values than the 9-7 filter. For the relaxed versions we have used the global relaxation parameter $\theta = 25$.

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

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

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.

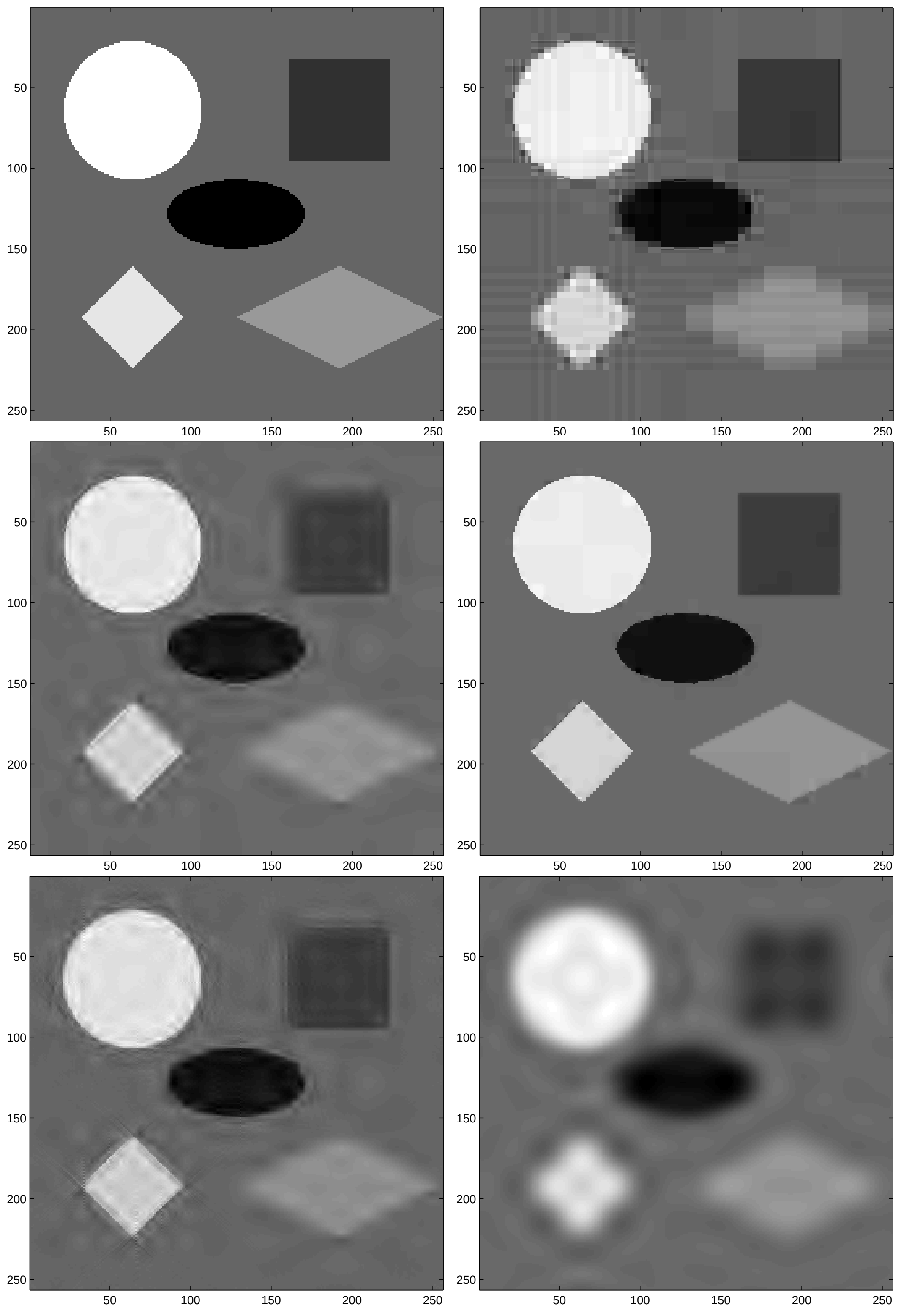

Figure 7: Approximation of a synthetic image with 512 coefficients. See PSNR values in Table 2. (a) Input, (b) classical Haar, (c) Biorthogonal 9-7, (d) Tetrolets, (e) Contourlets, (f) Curvelets.

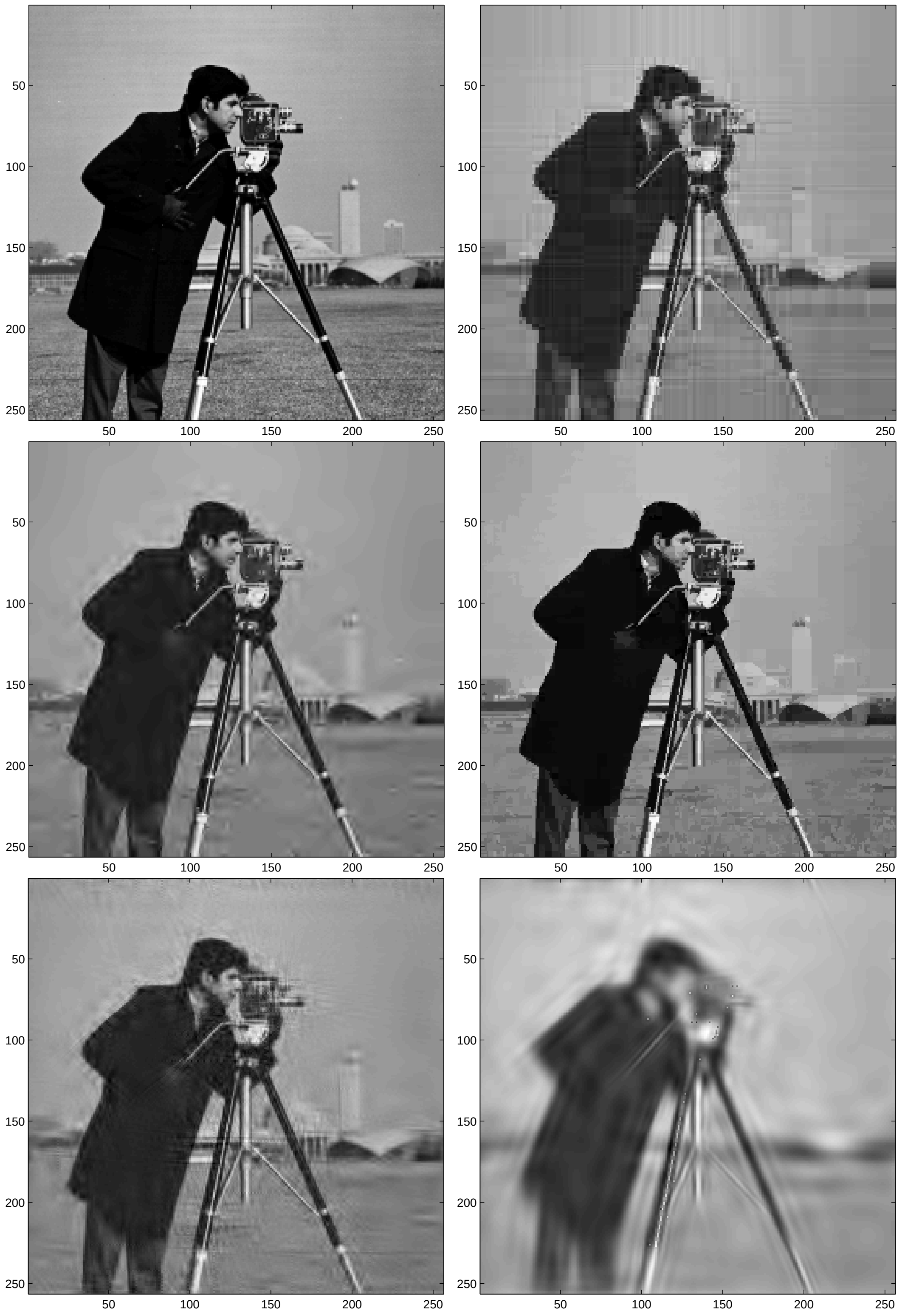

Figure 8: Approximation of the 'cameraman' image with 2048 coefficients. See PSNR values in Table 2. (a) Input, (b) classical Haar, (c) Biorthogonal 9-7, (d) Tetrolets, (e) Contourlets, (f) Curvelets.

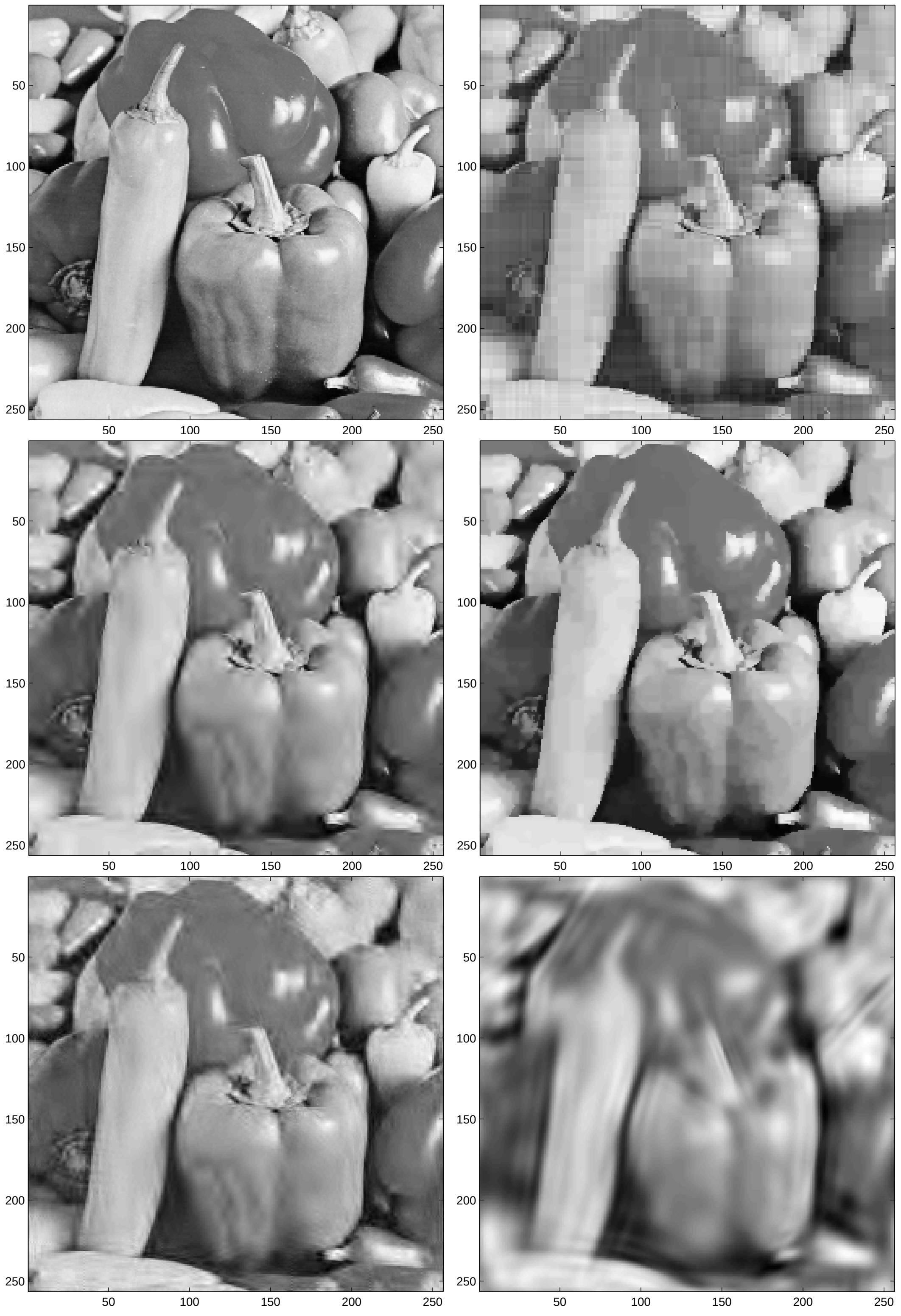

Figure 9: Approximation of the 'pepper' image with 2048 coefficients. See PSNR values in Table 2. (a) Input, (b) classical Haar, (c) Biorthogonal 9-7, (d) Tetrolets, (e) Contourlets, (f) Curvelets.

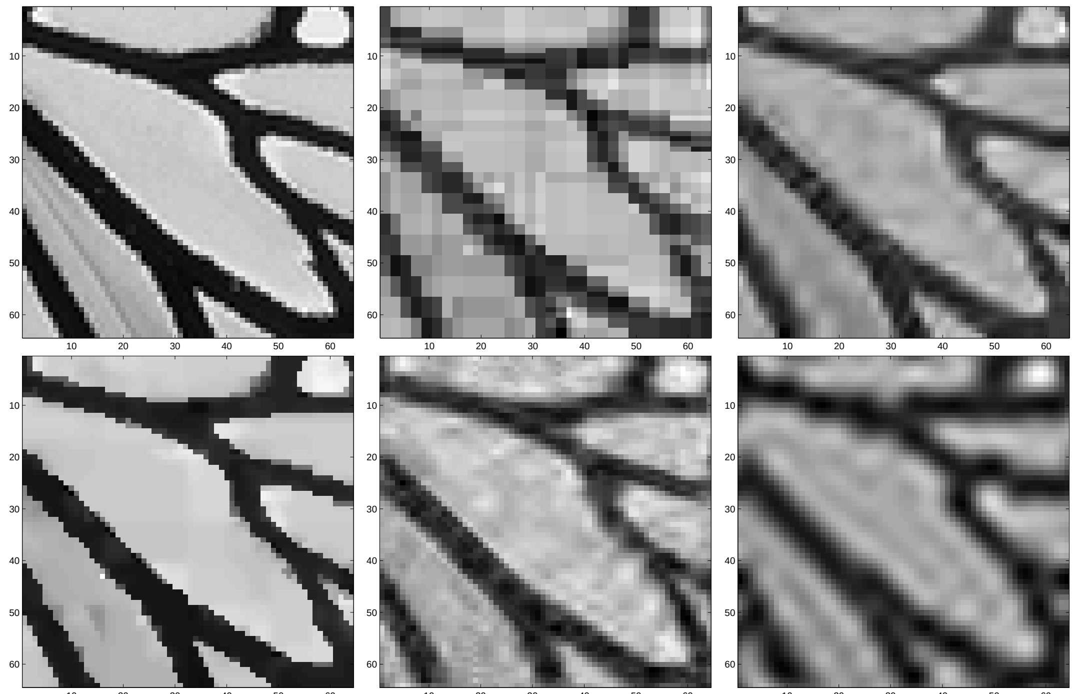
Figure 10: Approximation of the 64 × 64 'monarch' detail image with 256 coefficients. See PSNR values in Table 2. (a) Input, (b) classical Haar, (c) Biorthogonal 9-7, (d) Tetrolets, (e) Contourlets, (f) Curvelets.

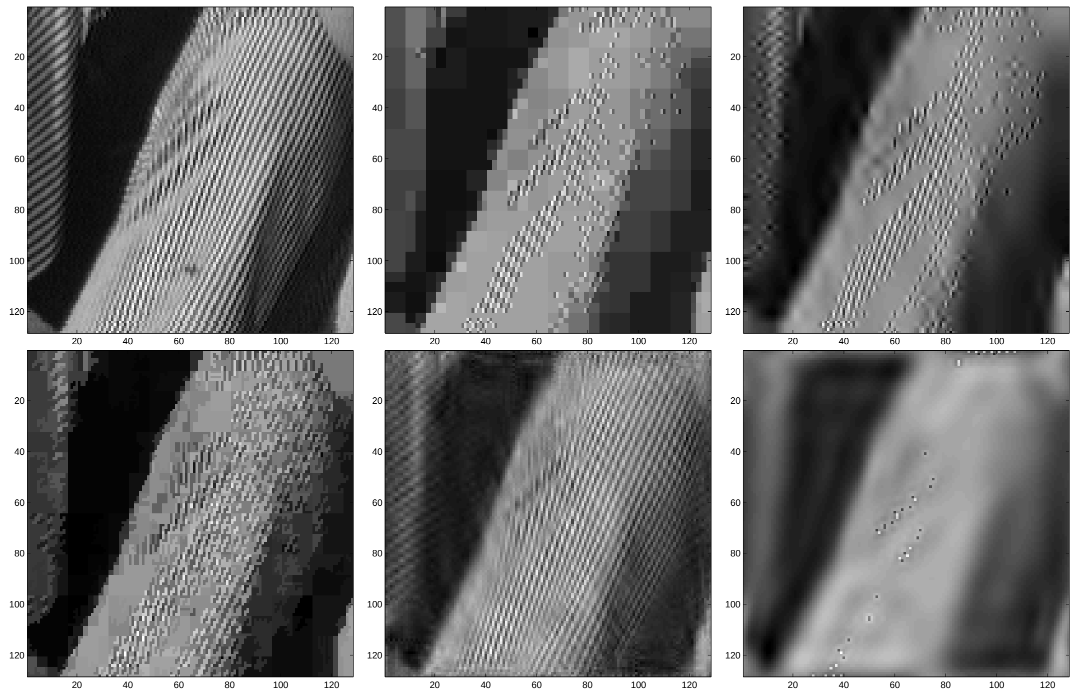
Figure 11: Approximation of the 128 × 128 'barbara' detail image with 512 coefficients. See PSNR values in Table 2. See PSNR values in Table 2. (a) Input, (b) classical Haar, (c) Biorthogonal 9-7, (d) Tetrolets, (e) Contourlets, (f) Curvelets.

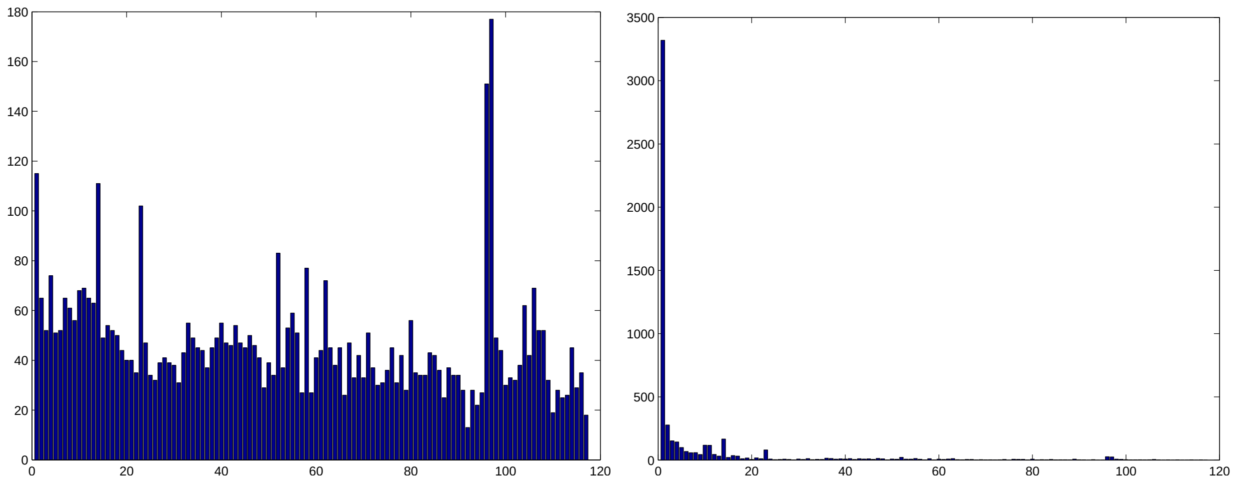

Figure 12: Distribution of the 5461 adaptivity values with the cameraman image. (a) Standard tetrolet transform, entropy is 0.56 bpp, (b) relaxed tetrolet transform with $\theta = 25$, entropy is 0.25 bpp.

An exemplary image comparison is given in Figure 13 for the cameraman image. The several tetrolet approximations with 2048 coefficients are compared regarding PSNR value and the entropy of the adaptivity values. This example illustrates that the three tetrolet modifications as well as the combination of them essentially reduce the entropy under small loss of approximation quality.

	monarch				cameraman			
	coeff	PSNR	entropy	$cost_{full}$	coeff	PSNR	entropy	$cost_{full}$
Tensor Haar	300	19.58	-	1.55	2500	26.29	-	0.84
Tensor 9-7	300	22.62	-	1.55	2500	28.14	-	0.84
Tetrolet	256	24.43	0.53	1.86	2048	29.17	0.56	1.26
Tetrolet 16	256	23.56	0.30	1.64	2048	28.44	0.32	1.02
Tetrolet rel	256	24.51	0.32	1.66	2048	28.91	0.25	0.95
Tetrolet edge	256	24.24	0.43	1.77	2048	28.94	0.32	1.02
Tetrolet 16 rel edge	256	23.48	0.21	1.55	2048	28.24	0.14	0.84

Table 3: Comparison of tensor wavelet transforms and modified tetrolet transforms regarding quality (PSNR in dB) and storage cost ($cost_{full}$ in bpp). For an image comparison of the 'cameraman' see Figure 13.

6.3. Computational efficiency

The rapidness of the filter bank algorithm is an important feature of the tetrolet transform which makes it relevant for practical tasks. Our test was based on using a MacBook with a 2 GHz Intel Core 2 Duo processor and 4 GB of RAM. The routines were tested in MATLAB (without C routines), we have computed five decomposition levels for the 'pepper' image as reference. We have compared the computational efficiency of the decomposition and reconstruction of several versions of the tetrolet transform with the contourlet and curvelet transform, see Table 4. Of course, the directional contourlet and curvelet transforms achieve shorter computational time, but note that these transforms are non-adaptive. Considering the computation times in Table 4, we should also keep in mind that the fast computation of contourlet and curvelet transform is based

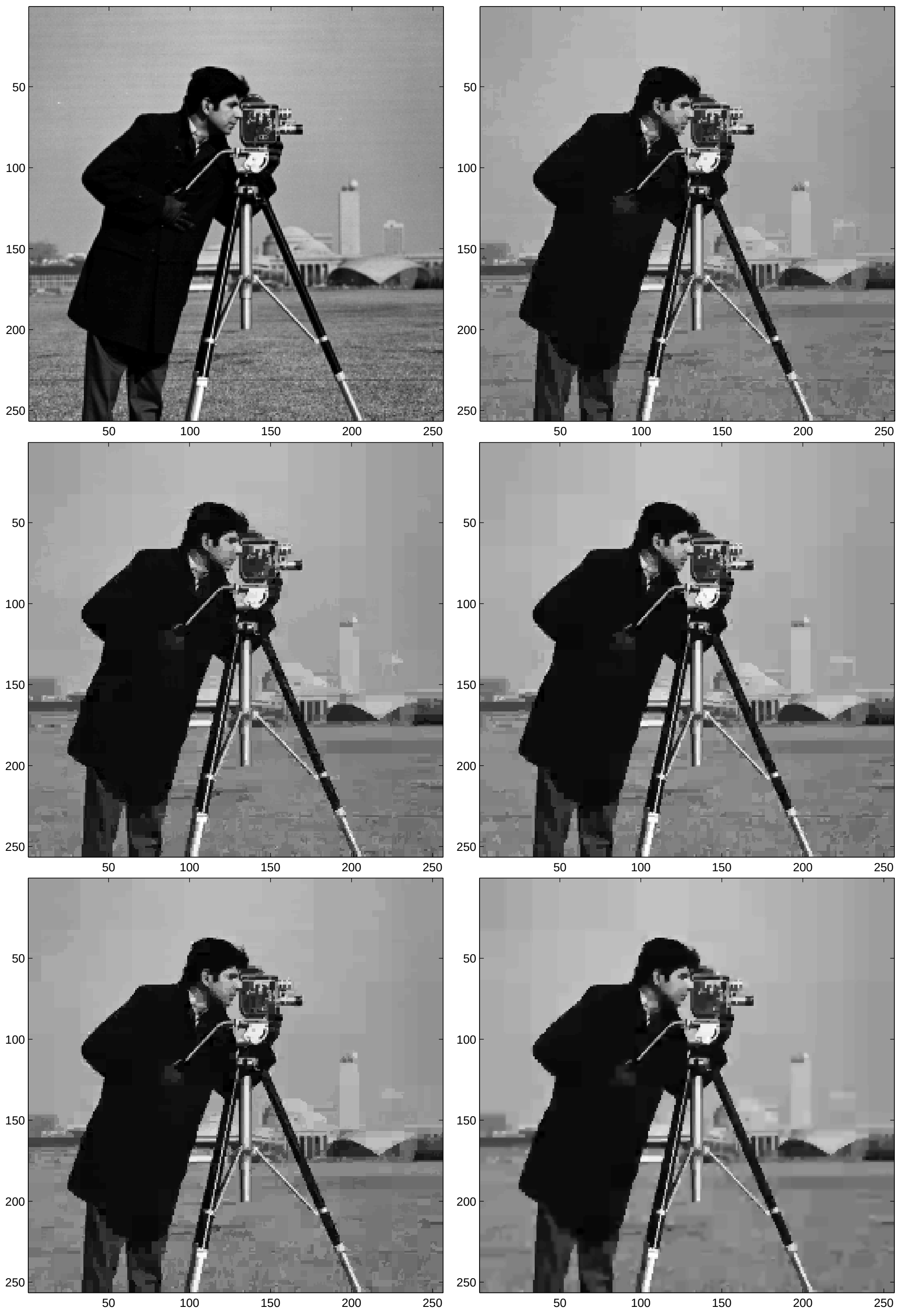

Figure 13: Comparison of modified tetrolet transforms. (a) Input, (b) Tetrolet, PSNR 29.17 dB, $E = 0.56$, (c) Tetrolet 16, PSNR 28.44 dB, $E = 0.32$, (d) Tetrolet rel, PSNR 28.91 dB, $E = 0.25$, (e) Tetrolet edge, PSNR 28.94 dB, $E = 0.32$, (f) Tetrolet 16 rel edge, PSNR 28.24 dB, $E = 0.14$.

on the usage of C++ programms while our tetrolet transform only uses MATLAB features. For adaptive methods, the measured CPU time of the tetrolet transform is exceeding fast. For an image of 256×256 the decomposition needs only a single second. The rapidness of our filter bank is based on Remark 3.1, where the tetrolet transform is interpreted as a Haar wavelet transform with locally permuted pixels. Other adaptive schemes like wedgelets or bandelets are much more expensive, see [7].

	Decomposition		Reconstruction	
image size	256×256	512×512	256×256	512×512
Tetrolet	1.0212	4.0259	0.4451	2.0324
Tetrolet 16	0.6592	2.6735	0.4125	1.7821
Tetrolet rel	1.1419	4.5773	0.4302	1.8668
Tetrolet edge	1.8096	5.5622	0.4150	1.8373
Tetrolet 16 rel edge	0.6548	2.8035	0.4188	1.8457
Contourlets	0.1536	0.4689	0.1539	0.4782
Curvelets	0.3490	1.9876	0.3741	1.5079

Table 4: Comparison of directional wavelet transforms and modified tetrolet transforms regarding computation time (in sec).

The inverse tetrolet transform does not depend on the different modifications, thus the computation time of the reconstruction filter bank algorithm is similar for all tetrolet versions. In constrast, the decomposition time varies, because there the adaptivity comes into play. Notice that the reduction of the number of admissible tetromino tiling to only 16 configurations ('Tetrolet 16') almost halves the computation time.

7. Conclusion

In this paper, we have proposed a geometric adaptive transform which is especially designed for sparse image representation. The basis functions have tetromino support and are able to adapt different directions in images. The Haar-type tetrolets produce a fast filter bank algorithm which offers good approximation results even for natural images.

In a further paper [12], we have shown that we can improve the tetrolet approximation quality by a suitable post-processing step which increases the regularity of the piecewise constant approximation.

A natural question is the application of the tetrolet transform to image denoising. As shown in Section 4, our tetrolet system forms a basis and not a frame, and hence, it is non-redundant. For sparse image representation non-redundancy is desirable, while for image denoising redundant information is helpful. Therefore the tetrolet transform is not very effective in denoising application. A second reason is the small support of the tetrolets, which leads to a small filtermask of length four. But in image denoising an averaging over more than four image pixels would be advantageous.

However, in order to apply the tetrolet transform in image denoising, one may combine the tetrolet transform with a pre-processing scheme which is suitable for denoising. Such a hybrid method seems to be promising.

Acknowledgment

The author would like to thank the referees for their suggestions which substantially improved this paper. The research in this paper is funded by the project PL 170/11-2 of the Deutsche Forschungsgemeinschaft (DFG). This is gratefully acknowledged. Furthermore, the author would like to thank his advisor Gerlind Plonka for her helpful instructions and permanent support as well as Stefanie Tenorth and Michael Wozniczka for valuable comments.

References

[1] R. Breukelaar, E. Demaine, S. Hohenberger, H. Hoogeboom, W. Kosters, and D. Liben-Nowell, Tetris is hard, even to approximate, *Internat. J. Comput. Geom. Appl.* **14**(1-2) (2004), 41–68.

[2] E.J. Candès and D.L. Donoho, New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities, *Comm. Pure Appl. Math.* **57**(2) (2004), 219–266.

[3] C.-L. Chang and B. Girod, Direction-adaptive discrete wavelet transform for image compression, *IEEE Trans. Image Process.* **16**(5) (2007), 1289–1302.

[4] W. Ding, F. Wu, X. Wu, S. Li, and H. Li, Adaptive directional lifting-based wavelet transform for image coding, *IEEE Trans. Image Process.* **16**(2) (2007), 416–427.

[5] D.L. Donoho, Wedgelets: nearly minimax estimation of edges, *Ann. Statist.* **27**(3), (1999), 859–897.

[6] M.N. Do and M. Vetterli, The contourlet transform: an efficient directional multiresolution image representation, *IEEE Trans. Image Process.* **14**(12) (2005), 2091-2106.

[7] F. Friedrich, L. Demaret, H. Führ, K. Wicker, Efficient moment computation over polygonal domains with an application to rapid wedgelet approximation, *SIAM J. Scientific Computing* **29**(2) (2007), 842–863.

[8] H. Führ, L. Demaret, and F. Friedrich, Beyond wavelets: New image representation paradigms. Book chapter, in: M. Barni and F. Bartolini, Document and Image Compression, CRC Press, 2006.

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

[10] K. Guo and D. Labate, Optimally sparse multidimensional representation using shearlets, *SIAM J. Math. Anal.* **39**(1) (2007), 298–318.

[11] M. Korn, Geometric and algebraic properties of polyomino tilings, PhD thesis, Massachusetts Institute of Technology (2004).

[12] J. Krommweh and J. Ma, Tetrolet shrinkage with anisotropic total variation minimization for image approximation, to appear in *Signal Process.*, 2009.

[13] B. Larsson, Problem 2623, in: *Fairy Chess Review* **3**(5) (1937), 51.

[14] S. Mallat, Geometrical grouplets, *Appl. Comput. Harmon. Anal.* **26**(2) (2009), 161–180.

[15] E. Le Pennec and S. Mallat, Sparse geometric image representations with bandelets, *IEEE Trans. Image Process.* **14**(4) (2005), 423–438.

[16] G. Plonka, Easy path wavelet transform: a new adaptive wavelet transform for sparse representation of two-dimensional data, *Multiscale Model. Simul.* **7**(3) (2009), 1474–1496.

[17] V. Velisavljević, B. Beferull-Lozano, M. Vetterli, and P.L. Dragotti, Directionlets: anisotropic multi-directional representation with separable filtering, *IEEE Trans. Image Process.* **17**(7) (2006), 1916–1933.