Second Order Wedgelets in Image Coding

Agnieszka Lisowska*
*University of Silesia, Institute of Informatics, ul. Bedzinska 39, 41-200 Sosnowiec, Poland,
e-mail: *alisow@ux2.math.us.edu.pl*

***Abstract*— In these days efficient image coding plays a very important role. There are well known and recognized theories concerning this topic such as wavelets or new and recently developed ones as geometrical wavelets. The latter one, thanks to the possibility of catching discontinuities in different locations, scales and orientations better reflects the Human Visual System than classical wavelets. In the paper we proposed the improvement of the algorithm used in image coding based on wedgelets — a kind of geometrical wavelets. The proposed algorithm is based on second order wedgelets which are based not only on straight edges but also on fragments of second degree curves in order to ensure more sparse approximation of an image in rate distortion sense. The performed experiments confirmed that the use of second order wedgelets ensures better compression properties in image coding than the use of wedgelets.**

***Keywords*— Wedgelets, second order wedgelets, geometrical image coding.**

I. Introduction

The problem of efficient image approximation plays an important role in image processing. For years the wavelets theory has been seen as the most promising one. However taking into account all characteristics of a digital still image (such as changings in location, scale and orientation) one can conclude that wavelets are too poor to reflect all the features related to images and the way in which human eye and brain interpret information about its content. Indeed, wavelets are good tool only in the case of catching point discontinuities of signal. They fail in the case of line singularities, so often present in images as edges. This is due to the fact that standard wavelet transforms applied to images are separable. Furthermore the Human Visual System is sensitive, in first order, just to edges present in images and only in next order to textures [1].

To better reflect all the characteristics of an image the theory of geometrical wavelets has been developed. It has evaluated in many directions giving the definitions of wedgelets [2] beamlets [3], curvelets [4], ridgelets [5] and others "X–lets" as well as their appropriate fast transforms. The main advantage of these new wavelets lies in the fact that they possess all the advantages of classical wavelets, that is space localization and scalability (catching global as well as local characteristics of a signal). But additionally the geometrical wavelet transforms have strong directional character. They allow to catch changes of a signal in different directions.

Geometrical wavelets were improved and applied with success to many areas of digital image processing. In particular the theory of wedgelets was used, for example, in multiresolution compression of images [6], [7]. As shown in [7] the compressor based on wedgelets gives better performance in image compression even than the recognized standard $JPEG2000$. Wedgelets have also been used successfully in image segmentation and noise removal [8]. Also other applications are still found in these days [9].

At the beginning, in the literature related to adaptive geometrical wavelets only the wavelets, especially wedgelets or beamlets, based on straight edges were taken into considerations. Such an approach has been recognized as sufficient because it has been proven that it is the optimal one in asymptotic sense. But taking into account that discontinuities present in images are modeled as functions called "horizons" which are elements of C^2 class, from the mathematical point of view it is clear that such functions may be approximated by any order polynomials (or even any other kind of functions) than the linear ones used so far in the case of mentioned wavelets. On the other hand, from the practical point of view, it must be remembered that the higher the order of the approximating polynomial the larger the number of coefficients needed to represent it, which causes the larger size of the dictionary of such generalized wavelets. So, because too large dictionaries are not good, due to too complex computations, some compromise should be found. Such an approach should assure good approximation of an image on the one hand and quite compact and small dictionary on the other hand.

Such a compromise approach (that is wedgelets based on second degree curves) has been first proposed and published by the author in 2003 [10] and later [11], [12], [9]. But in 2004 also the similar idea of generalization to any degree polynomial appeared independently, under the name of surflets [13]. Though the latter authors proposed in some sense a similar solution of generalization of wedgelets, they built the dictionary of surflets in different way and also proposed different use of such dictionary in image coding.

In this paper the modification of the one of the most commonly used algorithm for wedgelet image coding has been proposed. The new algorithm is based on generalized wedgelets presented in [9], [10], [12].

II. Classical Wedgelets

A. Preliminaries

The majority of definitions presented in this section follows from the work of Donoho [2] and are the basic foundations on which the other adaptive geometrical wavelets have been defined.

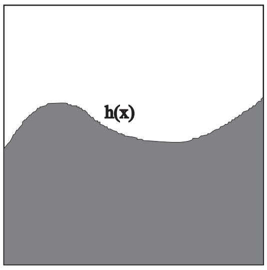

Fig. 1. Graphical example of "horizon" and "horizon function".

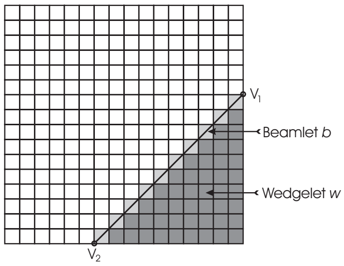

Fig. 2. Graphical representation of an edge (called beamlet after 2000) and wedgelet.

B. The Class of Horizon Functions

Let us define an image domain $S = [0,1]\times[0,1]$. Next, let us denote function $h(x)$ defined on S as the "horizon", that is any continuous and smooth function defined on the interval $[0,1]$. Such function, for our further purposes, must fulfill the appropriate Hölder regularity conditions [2]. However, in practical applications it is sufficient to assume that the function h is of C^2 class.

Further, consider the characteristic function

$$H(x_1,x_2) = \mathbf{1}\{x_2 \geq h(x_1)\}\ , \qquad 0 \leq x_1, x_2 \leq 1\ . \quad (1)$$

Then the function H is called the "horizon function" if h is "horizon". The function H models a black and white image with a horizon where the image is white above the horizon and black below. The graphical example of such "horizon" and "horizon function" is presented in Fig. 1.

C. Dictionary of Wedgelets

Having an image domain as $S = [0,1]\times[0,1]$ one can, in some sense, discretize it on different levels of multiresolution. Consider the dyadic square $S(j_1,j_2,i)$ as the two dimensional interval

$$S(j_1,j_2,i) = [j_1/2^i, j_1+1/2^i]\times[j_2/2^i, j_2+1/2^i]\ , \quad (2)$$

where $0 \leq j_1, j_2 < 2^i$, $i \geq 0$ and $j_1, j_2, i \in \mathbb{N}$. Note that $S(0,0,0)$ denotes the whole image domain S, that is the square $[0,1]\times[0,1]$. On the other hand $S(j_1,j_2,I)$ for $0 \leq j_1, j_2 < N$ denote appropriate pixels from $N \times N$ grid, where N is dyadic (it means that $N = 2^I$). From this moment let us consider a domain of image as such $N \times N$ grid of pixels.

Having assumed that an image domain is the square $[0,1]\times[0,1]$ and that it consists of $N \times N$ pixels (or, more precisely, quantum squares of size $1/N$) one can note that on each border of any square $S(j_1,j_2,i)$, $0 \leq j_1, j_2 < 2^i$, $0 \leq i \leq \log_2 N$, $j_1, j_2, i \in \mathbb{N}$, we may denote the vertices with distance equal to $1/N$. Let us next enumerate them starting from the upper right corner of the square in clockwise direction. Every two such vertices in any fixed square may be connected to form a straight line — edge (also called *beamlet* after the work of Donoho and Huo [3]).

In such a way, within a domain S, one can define edges with different location, scale and orientation. It has been proven [2] that such a set of edges is sufficient to represent edges present in an image. The set of such edges forms the dyadic dictionary. Of course, as one can easily see, that dictionary does not contain all possible edges, which may occur in an image (it is easy to see that the number of all such edges is far larger than that of the dictionary). But this dictionary, as it has been shown [2], allows to represent any smooth (that is of C^2 class) image with high accuracy. It follows from the fact that the dictionary contains edges in many locations, scales and orientations.

Assume now that the considered edges are not degenerated, that is, they do not lie at the border of the square. Then each such edge b splits any square S (we skip the subscripts denoting location and scale for a moment for better clarity) into two pieces. Let us consider one of the two pieces which is bounded by lines connecting in turn in clockwise direction, from the upper right corner, the first of the two edge vertices and then the second one. Let us define then the indicator function of that piece

$$w(x_1,x_2) = \mathbf{1}\{x_2 \leq b(x_1)\}\ , \qquad (x_1,x_2) \in S\ . \quad (3)$$

Such a function we call *wedgelet* defined by the edge b. The graphical representation of the wedgelet on S defined by the beamlet b is presented in Fig. 2.

It is obvious that on an arbitrary square S one can define many different wedgelets. Moreover, also the function which is the indicator function of the whole square S is taken as the wedgelet. So, more formally, one can define the set of wedgelets on any S as [2]

$$W(S) = \{\mathbf{1}(S)\} \cup \{\text{all possible } w \text{ defined on } S\}\ . \quad (4)$$

Additionally, within the whole image domain $S = [0,1]\times[0,1]$, the wedgelets are defined in different scales and locations (as stated in the case of beamlets). So, finally, one can define the wedgelet's dictionary W as the sum of all sets $W(S(j_1,j_2,i))$ of all dyadic squares $S(j_1,j_2,i)$, $0 \leq j_1, j_2 < 2^i$, $0 \leq i \leq \log_2 N$, $j_1, j_2, i \in \mathbb{N}$.

Let us assume from now on that the pair of subscripts j_1, j_2 such that $0 \leq j_1, j_2 < 2^i$ is replaced by the only subscript j such that $0 \leq j < 4^i$. Such enumerations are equivalent, but the last one is more flexible. And let us denote $S(j_1,j_2,i)$ as $S_{i,j}$. Additionally, for the parameterization of direction (denoted so far by coordinates v_1, v_2) let us denote by m. Because in practical applications different parameterizations of direction are used such a general model seems to be more flexible. Indeed, the above assumptions allow to parameterize the wedgelets' dictionary using one parameter for scale i, one for location j and one for orientation m. More formally, we have the following definition.

Definition 1. The *Wedgelets' Dictionary* is defined as the following set:

$$W = \{w_{i,j,m} : i = 0, \ldots, \log_2 N, j = 0, \ldots, 4^i - 1, \\ m = 0, \ldots, M_W(S_{i,j}) - 1\}, \quad (5)$$

where $M_W(S_{i,j})$ denotes the number of wedgelets on $S_{i,j}$.

Note that such a dictionary of wedgelets contains quite a large set of constant functions with discontinuities (seen as edges in images) along different locations, scales and orientations, which may be used in image representation. It is astonishing that such a set of functions can better approximate a wide class of images than the classical wavelet basis, so far seen as the best in the field of image representation.

D. Wedgelet Transform

In adaptive methods of representation, the correlation of an image with all atoms of the dictionary must be assigned. In such a case the least squares projection is computed. So, the same occurs in the case of wedgelets' dictionary. Having defined such dictionary and an image $F : S \to \mathbb{N}$, the wedgelet transform can be defined.

Definition 2. The *Wedgelet Transform* is defined by the following formula:

$$\alpha_{i,j,m} = \frac{1}{T} \iint_S F(x_1, x_2) w_{i,j,m}(x_1, x_2) dx_1 dx_2 , \quad (6)$$

where

$$T = \iint_S w_{i,j,m}(x_1, x_2) dx_1 dx_2 \quad (7)$$

is the normalization factor and $S = [0, 1] \times [0, 1]$, $\alpha_{i,j,m} \in \mathbb{R}$, $w_{i,j,m} \in W$, $0 \le i \le \log_2 N$, $0 \le j < 4^i$, $0 \le m < M(S_{i,j})$ and $i, j, m \in \mathbb{N}$.

From the practical point of view it means that the mean of all pixels (the values of which are denoted as $F(x_1, x_2)$) belonging under an appropriate wedgelet is computed. In the case of grayscale images coefficients are additionally quantaized to $\alpha_{i,j,m} \in \{0, \ldots, 255\}$. Such coefficients denote mean grayscale intensities of the regions covered by appropriate wedgelets. In the case of binary images we have quantization such that $\alpha_{i,j,m} \in \{0, 1\}$.

The wedgelet image representation is defined by the following formula:

$$F(x_1, x_2) = \sum_{i,j,m} \alpha_{i,j,m} w_{i,j,m}(x_1, x_2) . \quad (8)$$

But because W is a dictionary, not a basis, not all the coefficients $\alpha_{i,j,m}$ from the transform are used in the above representation (or in other words, some of them are equal to 0). The way in which they are chosen to represent F is described in the next section.

Because we look for the best (in mean square error sense) approximations with the use of the smallest number of atoms from a given dictionary of wedgelets, in the next section the commonly used method of image approximation in the best rate distortion sense [2] is described.

E. Wedgelet Analysis of Image

Nearly all multiresolution methods of image approximations use quadtrees as the main data structures. So is also in the case of wedgelets approximations. There are few different ways in which wedgelets may be stored with the help of quadtrees. It means that the coefficients may be stored differently from method to method, depending on applications. The simplest and commonly used one assumes that in each node of quadtree the set of coefficients determining appropriate wedgelet is stored. More sophisticated methods, exactly as in the case of classical wavelets assume the storage in nodes of tree differences of coefficients between two different consecutive levels of tree. Such an approach is based on the so–called Laplacian pyramid [14] commonly used in the case of classical as well as geometrical wavelet coding.

The base algorithm of image decomposition is performed in two steps. The first one is the full wedgelet decomposition of an image with the help of wedgelet transform. It means that for each square $S_{i,j}$, $0 \le j < 4^i$, $0 \le i \le \log_2 N$ the best approximation in mean square error sense by wedgelet is found. After the full decomposition (on all levels) all wedgelet coefficients are stored in the nodes of quadtree. Then, in the second step, some kind of optimization algorithm, the bottom-up tree pruning algorithm, is applied to get a possibly minimum number of atoms in approximation, ensuring the best image quality. Indeed, the following weighted sum is often minimized [2]:

$$R_\lambda = min_P\{||F - F_W||_2^2 + \lambda^2 K\} , \quad (9)$$

where P is homogenous partition of an image (elements of which are stored in the quadtree from the first step), F denotes the original image, F_W its wedgelet approximation, K is the number of bits needed to code the approximation and λ is the distortion rate parameter (also called penalization factor). In the case of exact image approximation the quality is determined and the reconstructed image is exactly like the original one. The two steps are described in more detail in [2], [9].

III. Second Order Wedgelets

A. Background

To model a smooth image with smooth discontinuity as in the previous section, consider a "horizon" h as has been defined and "horizon function" H related to it. In reality, the modeling of such two dimensional "horizon function" is equivalent to modeling of one dimensional "horizon". It follows from the fact that wedgelets approximating H are determined by appropriate beamlets which approximate h.

The theory of wedgelets assumes that any "horizon", as a one dimensional function from C^2 class may be

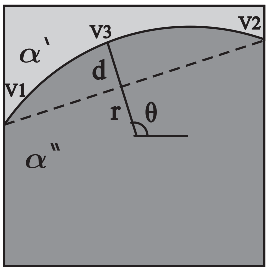

Fig. 3. Parametrization of second order beamlet and wedgelet.

piecewise approximated by the linear one, called beamlet. But the theory of function approximations is far wider than only approximating by linear functions. Indeed, there is a great number of other functions which may be used in approximations. The most commonly used are, for example, polynomials. So why not use them in place of linear beamlets? Unfortunately, the answer is not so simple and optimistic. It mainly follows from the fact that the higher the order of the approximating function the larger the number of parameters to represent it. Additionally, the time of computations also is longer. So we would like to simultaneously satisfy two opposite criteria, which is impossible. So, in some sense a compromise solution should be found. One looks for sparse approximations but having in mind also a small number of coefficients to code them. Taking into account applications in computer graphics such solutions exist, indeed the most commonly used functions in approximation or modeling are of degree two or three [15].

From the wedgelets or beamlets approximations' point of view, it seems that similar solution should give satisfactory results in an approximation. Indeed, as proposed in [9], [10], [11], adding one more parameter to the wedgelet or beamlet representation may cause better approximations, preserving comparable number of bits used in representation. Whereas adding second parameter does not cause such spectacular improvement in approximation and causes that too large number of bits is used in representation.

It is a well–known fact that any curve of degree two may be defined with the help of the general formula [15]:

$$ax_1^2 + 2bx_1x_2 + cx_2^2 + 2dx_1 + 2ex_2 + f = 0\,, \qquad (10)$$

where coefficients a, b, c are not simultaneously equal to zero and it is assumed that $acf + 2bde - cd^2 - ae^2 - b^2 f \neq 0$, $a, b, c, d, e, f \in \mathbb{R}$. Depending on the coefficients, three kinds of the so–called conic curves may be defined: parabolas, ellipses and hiperbolas. Let us define then a *second order beamlet* as a fragment of one of the cone curves. The example of parameterization of the new defined second order beamlet together with based on it second order wedgelet based on parabola is shown in Fig. 3

Note that in all the kinds of generalization the new beamlet may be concave or convex. To differentiate such two possibilities let us assume that for convex beamlets the parameter d is negative. Additionally, in all the cases of generalized beamlet representation, the additional parameter d in practical applications must be quantized. This may be done in two ways. The first one assumes that a constant number of bits (not depending on scale) is used to code d related to any beamlet. And the second one assumes that it is dependent on the scale and the larger the scale of beamlet the larger the number of bits to code d. In this paper the first approach has been used and the parameter d is coded with the help of four bits.

Note that such an approach causes that many similar methods of generalizations are proposed. But because all of them give comparable practical results, it is pointless to present the results of all of them in this paper. Additionally, the kinds of parameterizations related to any of these methods may be different. But the proposed ones seem to be the ones which are most natural and simple in implementation. In practical applications, only the ellipsoidal beamlets and wedgelets will be taken into account, due to the fact that other approaches give comparable results.

Finally, note that the class of second order curves does not include linear functions. But thanks to the presented method of parameterization, especially the parameter d, it follows that the class of generalized beamlets includes the class of straight beamlets (when $d = 0$), although in practical applications it is often better to treat these two classes (when $d = 0$ and $d \neq 0$) separately.

B. Second Order Wedgelets' Dictionary

Consider the notation used in the previous section where S is any dyadic square. Having the second order beamlet proposed above, one can define the *second order wedgelet* as

$$\widehat{w}(x_1, x_2) = \mathbf{1}\{x_2 \leq \widehat{b}(x_1)\}\,, \qquad (x_1, x_2) \in S\,, \quad (11)$$

where $\widehat{b}$ denotes the second order beamlet, which is defined as one of any curve of degree two described above.

Taking into account all the second order wedgelets of all locations, scales, orientations and curvature one can define the following dictionary.

Definition 3. The *Second Order Wedgelets' Dictionary* is defined as the following set:

$$\begin{aligned}\widehat{W} = \{\widehat{w}_{i,j,m,d} : i = 0, \ldots, \log_2 N,\\ j = 0, \ldots, 4^i - 1,\\ m = 0, \ldots, M_W(S_{i,j}) - 1,\\ d = -2^{D-1}, \ldots, 2^{D-1} - 1\}\,,\end{aligned} \qquad (12)$$

where $M_W(S_{i,j})$ denotes the number of straight wedgelets on $S_{i,j}$ and $D \in \mathbb{N}$ is the number of bits needed to code d.

Note that when $d = 0$ then $W = \widehat{W}$, in any case $W \subseteq \widehat{W}$. Such generalized wedgelets are characterized by four properties: scale, location, orientation and curvature. Note also that every second order wedgelet is related to an appropriate straight wedgelet. But the inverse relation is not so obvious. Note also that for some straight wedgelets (as well as beamlets related to them) appropriate second order wedgelets (or beamlets) may not exist. Indeed, the beamlets which determine wedgelets may fall outside the domain square. It is possible when a straight beamlet is too close to the border of the square. It is assumed in the paper that parameter d is coded with four bits, what gives sixteen possible values of it. So, theoretically, one wedgelet should determine sixteen generalized wedgelets, but in practice the number may be smaller. Moreover, in the case of small wedgelets (for example for squares of 2×2 pixels), the second order wedgelets are not defined.

C. *Second Order Wedgelet Transform*

Having defined second order wedgelets' dictionary, the following transform may be defined.

Definition 4. The *Second Order Wedgelet Transform* is defined by the following formula:

$$\widehat{\alpha}_{i,j,m,d} = \frac{1}{T} \iint_S F(x_1, x_2) \widehat{w}_{i,j,m,d}(x_1, x_2) dx_1 dx_2 \,, \tag{13}$$

where

$$T = \iint_S \widehat{w}_{i,j,m,d}(x_1, x_2) dx_1 dx_2 \tag{14}$$

is the normalization factor and $S = [0,1] \times [0,1]$, $\widehat{\alpha}_{i,j,m,d} \in \mathbb{R}$, $\widehat{w}_{i,j,m,d} \in \widehat{W}$, $0 \le i \le \log_2 N$, $0 \le j < 4^i$, $0 \le m < M_W(S_{i,j})$, $-2^{D-1} \le d < 2^{D-1}$ and $i, j, m, d \in \mathbb{N}$.

From the practical point of view, the second order wedgelet decomposition is computed in the same way as the wedgelet decomposition. But to speed up computations the best second order wedgelet matching is done in the following way. In the first order the best straight wedgelet matching of the appropriate image square is performed. Then, basing on such wedgelet, the best second order wedgelet matching is found only in the neighborhood of the straight wedgelet. So not all the possible second order wedgelets are taken into account during computations which saves time giving quite satisfactory results.

D. *Wedgelets Versus Second Order Wedgelets*

It may be proved that proposed second order wedgelets give better approximations in rate distortion sense then wedgelets [9]. However in order to testify also the practical differences in image coding with the use of both methods — the one based on wedgelets versus the second one based on second order wedgelets some experiments of standard benchmark images [16] coding have been performed. The results are presented in Fig. 4. It presents the graphical dependency between the number of bytes needed for the wedgelet and second order wedgelet representations, respectively, versus the Mean Square Error of such representations. As one can see the plot related to generalized wedgelets is situated slightly below the plot related to wedgelets. It means that, independently of the penalization factor λ, the minimum cost is smaller

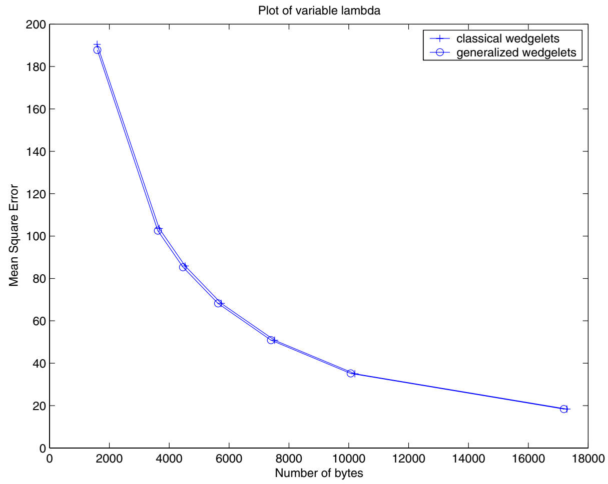

Fig. 4. The plot of dependency between the number of bytes and Mean Square Error.

in the case of second order wedgelet decomposition. Both of these two plots were obtained as the results of practical experiments performed on a number of standard benchmark still images.

IV. Improvement of the Coding Algorithm

The algorithm of image decomposition, along with the tree pruning mentioned above, constitute the basis used in nearly all applications of image processing and coding based on adaptive geometrical wavelets approach. In the case of processing it is necessary to have such a decomposition and there may be no need to code such an image. But in the case of image coding the decomposition obtained in such a way must be used somehow to obtain a bitstream. Such a bitstream may then be further compressed with the use of any known compression algorithm. Because there is a great number of different methods of coding and even more methods of its use in compression, it is difficult to give one standard description of image coding. This is because the theory of geometrical wavelets is quite new and new solutions are still being found these days. It may be said that nearly every group of researchers has its own ideas of image coding. Nevertheless, one method of coding seems to be simple and quite often used, though in many variations.

To better visualize the concept of image coding, the example of general scheme will be presented below, which additionally shows the progressiveness of the coding model.

A. *The Base Coding Algorithm*

Suppose that one has the abstract (but easy in the explanation of the idea) image as in Fig. 5[a]. The quadtree related to it is presented in Fig. 5[b]. Depending on the kind of wedgelet, the appropriate nodes of the tree are marked as D – decorated, U – undecorated or I – internal. Except for such marks, the nodes of the quadtree store three kinds of information:

- no information in the case of **internal** node;

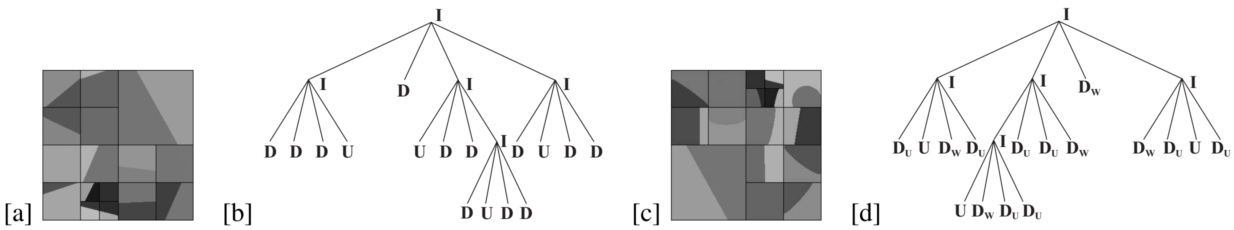

Fig. 5. Example of image coding: [a] sample wedgelet partition and [b] example of its coding; [c] sample second order wedgelet partition and [d] example of its coding.

- $\alpha_{i,j,m}$ coefficient in the case of **undecorated** node;
- $(m, \alpha'_{i,j,m}, \alpha''_{i,j,m})$ coefficients determining the wedgelet in the case of **decorated** node.

Having such a quadtree with parameters, one can simply write the data to the bitstream, according to the sequence starting from the root and writing all the coefficients together with the marks of nodes from every node one level below the root and so on. In reality, such symbols are coded as binary numbers. Note that the knowledge of the code allows to reconstruct the quadtree and next the image. Basing on that approach, the coding scheme like in the EZW [17] algorithm may be built where the most important coefficients are coded first.

B. The Improved Coding Algorithm

Suppose that one has the optimal decomposition of an image based on second order wedgelets. Such a decomposition may be stored in the quadtree in a similar way as in the base algorithm. More precisely, depending on the kind of wedgelet, the appropriate nodes of the tree are marked as I – internal, U – undecorated, D_W – decorated by straight wedgelet (that is generalized wedgelet for $d = 0$), D_U – decorated by second order wedgelet ($d \neq 0$). Except such marks the nodes of quadtree store four types of information:

- no information in the case of **internal** node;
- $\alpha_{i,j,m}$ coefficient in the case of **undecorated** node;
- $(m, \alpha'_{i,j,m}, \alpha''_{i,j,m})$ coefficients determining the wedgelet in the case of node **decorated by wedgelet**;
- $(m, d, \widehat{\alpha}'_{i,j,m,d}, \widehat{\alpha}''_{i,j,m,d})$ coefficients determining the wedgelet in the case of node **decorated by second order wedgelet**.

The example of image with second degree wedgelets together with the quadtree related to it are presented in Fig. 5[c] and Fig. 5[d], respectively.

Let us note that such a modification does not lengthen the code of the whole quadtree. Indeed, in the case of node marks in the base algorithm two bits per one mark are needed (because there are three different marks), so is also in the case of improved algorithm (where there are four marks). Additionally, from performed experiments follows that nearly 7% of all wedgelets used to code an arbitrary still image constitute second order wedgelets. From the same experiments follows also that often in the process of image coding four nodes from the base quadtree, decorated by wedgelets and having the same parent are replaced by the only one node decorated by second order wedgelet from the improved quadtree. So the additional four bits needed to code the parameter d in second order wedgelet are often outnumbered by the profit of the number of nodes reduction from four to only one. In fact the overall code length from improved algorithm is shorter than that in the case of base algorithm.

V. Experimental Results

The experiments presented in the paper have been performed on the database consisting of about 30 well known grayscale images taken from [16], [18]. Such images are widely known in the researchers' community as the benchmarks used in nearly all experiments. Because color images may be treated in a similar way, they were omitted. Due to the fact that in the quadtree–based methods of image coding or processing, the images of dyadic size are taken into account all of the tested images, when necessary, have been resized to 256×256 pixels.

In order to check the presented theory in practice a number of experiments of image coding have been performed. For comparison purposes all tested images were coded with the help of two methods: the base one and the proposed one, both described in the previous section. According to the known fact that artificial and still images are characterized by different characteristics, both of these groups have been treated independently.

A. Artificial Images

Wedgelets seem to be a great tool of artificial image coding, due to the fact that they often are composed of constant regions divided by smooth edges. Such a model is ideal for wedgelets. Additionally, from performed experiments it follows that the use of generalized wedgelets in image coding improves the results.

Tab. I gathers the numerical results of coding performed on a number of artificial images using the standard method and the improved method of coding presented above. In the last column of the table, the relative improvement concerning the number of bytes is presented. As one can see, in the case of coding images with many arc edges ("blobs" and "circles") the byte saving may amount even to 25%, preserving the original quality of reconstructed images. On the other hand, coding of image with only straight edges ("squares") leads to the same byte budget, giving no improvement. Indeed, the results are identical. In the rest of cases ("france", "skier" and "slope") the results are more scattered and in the case of

TABLE I
EXPERIMENTAL RESULTS OF CODING PERFORMED ON ARTIFICIAL IMAGES.

Coding	classical		generalized		$1-\frac{b_g}{b_c}$
of image	bytes	PSNR	bytes	PSNR	(%)
blobs	949	∞	740	∞	22.02
circles	1638	∞	1224	∞	25.27
france	16126	33.47	15935	33.45	1.18
skier	6365	36.85	6155	36.92	3.30
slope	2832	37.46	2772	37.44	2.12
squares	70	∞	70	∞	0.00

TABLE II
EXPERIMENTAL RESULTS OF CODING PERFORMED ON STILL IMAGES.

Coding	classical		generalized		$1-\frac{b_g}{b_c}$
of image	bytes	PSNR	bytes	PSNR	(%)
balloons	14419	32.09	14386	32.14	0.23
barbara	14226	31.39	13949	31.37	1.95
bird	4784	36.08	4691	36.06	1.94
blocks	6273	35.83	6234	35.83	0.62
boat	12449	32.90	12266	32.90	1.47
bridge	21758	30.48	21292	30.45	2.14
cameraman	10234	32.82	10083	32.81	1.48
collie	9288	32.46	9159	32.49	1.39
frog	12902	30.94	12756	30.97	1.13
goldhill	13717	31.36	13434	31.37	2.06
house	6498	34.00	6442	34.01	0.86
lab	10898	33.25	10783	33.23	1.06
lena	11169	31.97	11075	31.99	0.84
mandril	24101	30.06	23733	30.03	1.53
monarch	13172	32.48	12880	32.44	2.22
mountain	24498	29.98	24132	29.96	1.49
pentag	21718	30.62	21197	30.59	2.40
peppers	9892	33.49	9723	33.48	1.71
san256	14568	33.10	14342	33.08	1.55
zelda	8350	33.59	8294	33.64	0.67
AVERAGE	13246	32.44	13043	32.44	1.44

comparable quality by the mean of PSNR values the byte saving is nearly 2.2%.

In the case of coding of artificial images it is difficult to formulate any constructive conclusions about the order of improvement. From the performed experiments it follows that the improvements of image coding may be quite varied, from 0% to 25%. Additionally, it is obvious that such a new method of coding is pointless in the case of images with only straight edges. It does not give better results and only lengthens time of computations. In all other cases (to mention also the coding of isobars by second order beamlets described by the author in [11]) it works satisfactorily.

B. Still Images

Recently, wedgelets have been used also in still image coding. Unlike artificial images, this kind of images is characterized by more details and smooth, but not constant, regions. However, also in such a case the use of wedgelets makes sense. Moreover it constitutes a competitive method in comparison with classical wavelets.

Tab. II gathers the numerical results of coding performed on a number of the still benchmark images well known to research community. Also in that table in the last column the relative improvement concerning the number of bytes is presented. The tested images have been coded to give a comparable visual quality by the mean of PSNR value in both methods of coding. As one can see, the PSNR values of images coded by both methods are comparable only, not exactly the same. For such data the byte saving is from 0.23% to 2.40%. But on the average, the PSNR values for both methods of coding are identical, equal to $32.44\,dB$. And for such a quality of images the byte saving equals 1.44% on the average.

From the numerical results presented in the table it follows that in the case of coding of still images the improvement is not as spectacular as in the case of artificial ones. But, on the other hand, the results are more stable and predictable.

Additionally, in Fig. 6 an arbitrary chosen example of image coding has been presented. As one can see second order wedgelets representations give better quality of the image in the mean of PSNR values, simultaneously using a smaller number of bytes to code it.

The experimental results presented in this section are related only to efficient image coding. The compression step has been omitted in order to show the improvement lying only in the method of sparse image representation, independently on the compressor used. It means that different compressors may give slightly different results. Indeed, they influence the results. Additionally, the proposed method of representation may be used not only in image compression but also in other areas of image processing, as presented in [9].

VI. CONCLUSION

In recent years the theory of geometrical wavelets is one of the most promising and researched theories in these days because, in comparison with the classical wavelets, allows for more sparse representation of images due to the possibility of catching geometrical properties of images. This paper is concentrated on wedgelets because they may be used in artificial and still image coding in quite efficient way. Also the second order wedgelets were presented in the paper mainly due to the fact that in the earlier works [9], [10], [12] they were proposed without any special name, just as only generalized wedgelets. Additionally, the second order wedgelets have been presented in the paper in a systematized way.

The proposed improvement of the coding algorithm described in the paper consists of two steps. In the first step, in which the full quadtree must be found, the best wedgelet within a fixed node is found from the enlarged set of second degree wedgelets instead of the set of

Fig. 6. Example of image ”barbara” coding: [a] original image, [b] ”barbara” coded with 5821 wedgelets, 14211 bytes, PSNR 31.39 dB, [c] ”barbara” coded with 5695 second order wedgelets, 14185 bytes, PSNR 31.45 dB.

wedgelets. In the second step the improved quadtree is used and the standard bottom-up tree pruning algorithm is performed to find optimal decomposition of an image. Of course, as one can see, such improvement is time consuming. It means that the computation time is about four times longer, what follows from the experiments, however the estimated computation time remains still the same. It is normal and well known situation in image processing tasks that improvement of image quality or compression ratio always causes enlarging of memory or time consuming. But finally, one can conclude that the use of second order wedgelets together with the improved coding algorithm tend to better compression properties.

REFERENCES

[1] P.K. Kaiser: *The Joy of Visual Perception*, `http://www.yorku.ca/eye/thejoy.htm`, 2004.

[2] D.L. Donoho, “Wedgelets: Nearly–Minimax Estimation of Edges”, *Annals of Statistics*, vol. 27, pp. 859–897, 1999.

[3] D.L. Donoho, and X. Huo, “Beamlet Pyramids: A New Form of Multiresolution Analysis, Suited for Extracting Lines, Curves and Objects from Very Noisy Image Data”, *Proceedings of SPIE*, vol. 4119, 2000.

[4] E. Candès, and D.L. Donoho, “Curvelets — A Surprisingly Effective Nonadaptive Representation For Objects with Edges”, *Curves and Surfaces Fitting*, A. Cohen, C. Rabut, and L.L. Schumaker, Eds. Saint-Malo: Vanderbilt University Press, 1999.

[5] E. Candès, *Ridgelets: Theory and Applications*, Ph.D. Thesis, Departament of Statistics, Stanford University, 1998.

[6] J. Romberg, M.B. Wakin, and R. Baraniuk, “Approximation and Compression of Piecewise Smooth Images Using a Wavelet/Wedgelet Geometric Model”, *IEEE International Conference on Image Processing*, USA, September, 2003.

[7] M.B. Wakin, J. Romberg, H. Choi, and R. Baraniuk, “Geometric Tools for Image Compression”, *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, USA, November, 2002.

[8] L. Demaret, F. Friedrich, H. Führ, T. Szygowski, “Multiscale wedgelet denoising algorithms”, *Proceedings of SPIE*, San Diego, Wavelets XI, Vol. 5914, X1-12, 2005.

[9] A. Lisowska, *Geometrical Wavelets and their Generalizations in Digital Image Coding and Processing*, PhD Thesis, University of Silesia, Sosnowiec, Poland, 2005.

[10] A. Lisowska, “Effective coding of images with the use of geometrical wavelets”, *Proceedings of Decision Support Systems Conference*, Zakopane, Poland, 2003 (in Polish).

[11] A. Lisowska, “Bent Beamlets — Efficient Tool in Image Coding”, *Annales UMCS Informatica AI*, vol. 2, pp. 217–225, 2004.

[12] A. Lisowska, “Extended Wedgelets — Geometrical Wavelets in Efficient Image Coding”, *Machine Graphics & Vision*, vol. 13, no. 3, pp. 261–273, 2004.

[13] V. Chandrasekaran, M.B. Wakin, D. Baron, and R. Baraniuk, “Surflets: A Sparse Representation for Multidimensional Functions Containing Smooth Discontinuities”, *IEEE International Symposium on Information Theory*, Chicago, USA, 2004.

[14] P.J. Burt, and E.H. Adelson, “The Laplacian Pyramid as a Compact Image Code”, *IEEE Transactions on Communications*, vol. 9, no. 4, pp. 532–540, 1983.

[15] D. Marsh, *Applied Geometry for Computer Graphics and CAD*, Springer, 2000.

[16] *Waterloo BragZone*, `http://links.uwaterloo.ca/bragzone.base.html`, 2004.

[17] J.M. Shapiro, “Embedded Image Coding Using Zerotrees of Wavelet Coefficients”, *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, 1993.

[18] Y. Fisher, *Fractal Image Compression — Theory and Application*, Springer–Verlag, New York, 1994.