TERNARY WAVELETS AND THEIR APPLICATIONS TO SIGNAL COMPRESSION

GHULAM MUSTAFA*, FALAI CHEN*
ZHANGJIN HUANG*

* Department of Mathematics
University of Science and Technology of China
Hefei, Anhui 230026, P. R. China
e-mail: mustafa_rakib@yahoo.com, {chenfl, zjhuang}@ustc.edu.cn

We introduce ternary wavelets, based on an interpolating 4-point C^2 ternary stationary subdivision scheme, for compressing fractal-like signals. These wavelets are tightly squeezed and therefore they are more suitable for compressing fractal-like signals. The error in compressing fractal-like signals by ternary wavelets is at most half of that given by four-point wavelets (Wei and Chen, 2002). However, for compressing regular signals we further classify ternary wavelets into 'odd ternary' and 'even ternary' wavelets. Our odd ternary wavelets are better in part for compressing both regular and fractal-like signals than four-point wavelets. These ternary wavelets are locally supported, symmetric and stable. The analysis and synthesis algorithms have linear time complexity.

Keywords: subdivision, wavelets, multiresolution, signal compression

1. Introduction

Multiresolution analysis and wavelets have received considerable attention in recent years. Besides a broad range of applications in approximation theory (Daubechies, 1988), signal processing (Mallat, 1989) and physics, wavelets have also recently been applied to many problems in computer graphics. These graphics applications include image compression (DeVore *et al.*, 1992), fast methods for solving simulation problems in 3D modelling, and animation (Liu *et al.*, 1994), etc. Multiresolution analysis decomposes a complicated function into a low resolution part, together with a collection of perturbations, called wavelets coefficients, necessary to recover the original function. Wavelets provide a powerful and remarkably flexible set of tools for handling fundamental problems in science and engineering. There are many constructions of wavelets for functions parametrized over an interval (Andersson *et al.*, 1993). These have found use in signal processing, signal compression (Wei and Chen, 2002), and many other applications involving functions parametrized in one dimension. Classically, wavelets are functions generated from one basic function by dilations and translations. They admit a hierarchical decomposition. The functions that can be hierarchically decomposed can be generated through a simple process known as recursive subdivision. The strong connection between subdivision and wavelets allows us to create hierarchical bases for multiresolution analysis by subdivision.

In this paper, we develop a new class of wavelets, called ternary wavelets, based on an interpolating 4-point C^2 ternary stationary subdivision scheme (Hassan *et al.*, 2002), and describe how to use them in signal compression. The ternary wavelets are tightly squeezed and therefore are more appropriate for compressing fractal-like signals. Their degree of smoothness depends on various weight parameters. When the values of parameters are large, the corresponding scaling functions and wavelets are fractal-like. The article is organized as follows: First, we present some basic notions required to understand the theoretical frame of subdivision and multiresolution analysis. Secondly, we detail the construction of ternary wavelets based on a modified interpolating 4-point C^2 ternary stationary subdivision scheme. Finally, we give applications of our wavelets in signal compression where we show that our wavelets compare favorably with other similar wavelets, and we summarize the results for future research directions.

2. Subdivision and Multiresolution Analysis

As the present work depends on a previous one, for the clarity of foundations it will be necessary to review the subdivision and multiresolution analysis machinery previously developed (Hassan *et al.*, 2002; Lounsbery *et al.*, 1997).

2.1. An Interpolating 4-Point C^2 Ternary Stationary Subdivision Scheme

Stationary subdivision schemes, which are classified into binary and ternary subdivision schemes, have been studied in the fields of approximation and computer aided geometric design. Their mathematical description over arbitrary control polygons is as follows. Suppose that the initial control points of a control polygon are denoted by $p_i^0,\ i \in Z$. Then, for binary schemes the refined control points $\{p_i^{j+1}\}$ are obtained recursively from $\{p_i^j\}$ by the following formula:

$$p_i^{j+1} = \sum_{k\in Z} \gamma_{(2k-i)} p_k^j,$$

and similarly for ternary schemes:

$$p_i^{j+1} = \sum_{k\in Z} \gamma_{(3k-i)} p_k^j,$$

where $\gamma = (\gamma_i)$ is the mask of the scheme. There are many binary and ternary univariate subdivision schemes such as a 4-point subdivision scheme (Dyn *et al.*, 1987), a 6-point subdivision scheme (Weissman, 1990), ternary and 3-point univariate subdivision schemes (Hassan and Dodgson, 2001) and an interpolating 4-point C^2 ternary stationary subdivision scheme with a tension parameter (the 4-point ternary scheme, for brevity). We can achieve greater smoothness with the same number of control points by using a ternary rather than a binary subdivision scheme. Also, for the same smoothness, the 4-point ternary scheme has a much smaller support and a slightly lower computational cost than the equivalent binary scheme (Dubuc, 1986). These properties of ternary subdivision schemes motivate us to design wavelets based on a ternary, rather than a binary subdivision scheme. Here we give a brief introduction to the 4-point ternary scheme.

In this scheme, a polygon $P^j = (p_i^j)$ is mapped to a refined polygon $P^{j+1} = (p_i^{j+1})$ by applying the following three-subdivision rules:

$$\begin{cases} p_{3i}^{j+1} = p_i^j, \\ p_{3i+1}^{j+1} = a_0 p_{i-1}^j + a_1 p_i^j + a_2 p_{i+1}^j + a_3 p_{i+2}^j, \\ p_{3i+2}^{j+1} = a_3 p_{i-1}^j + a_2 p_i^j + a_1 p_{i+1}^j + a_0 p_{i+2}^j, \end{cases} \tag{1}$$

where the weights $\{a_j\}$ are given by

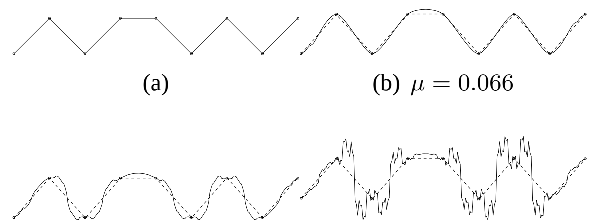
$$\begin{cases} a_0 = -\dfrac{1}{18} - \dfrac{1}{6}\mu, \\ a_1 = \dfrac{13}{18} + \dfrac{1}{2}\mu, \\ a_2 = \dfrac{7x}{18} - \dfrac{1}{2}\mu, \\ a_3 = -\dfrac{1}{18} + \dfrac{1}{6}\mu, \end{cases}$$

and $a_0 + a_1 + a_2 + a_3 = 1$. However, the above scheme does not interpolate endpoints. We modify it to interpolate the endpoints as follows:

$$\begin{cases} p_1^{j+1} = \dfrac{3}{4} p_0^j + \dfrac{1}{4} p_1^j, \\ p_2^{j+1} = \dfrac{1}{4} p_0^j + \dfrac{3}{4} p_1^j, \\ p_{3i}^{j+1} = p_i^j, \quad i = 0, 1, \ldots, n+1, \\ p_{3i+1}^{j+1} = a_0 p_{i-1}^j + a_1 p_i^j + a_2 p_{i+1}^j + a_3 p_{i+2}^j, \\ \qquad i = 1, 2, \ldots, n-1, \\ p_{3i+2}^{j+1} = a_3 p_{i-1}^j + a_2 p_i^j + a_1 p_{i+1}^j + a_0 p_{i+2}^j, \\ \qquad i = 1, 2, \ldots, n-1, \\ p_{3n+1}^{j+1} = \dfrac{3}{4} p_n^j + \dfrac{1}{4} p_{n+1}^j, \\ p_{3n+2}^{j+1} = \dfrac{1}{4} p_n^j + \dfrac{3}{4} p_{n+1}^j, \end{cases} \tag{2}$$

where the j-th control points are $\{p_i^j\},\ i = 0 \to n+1$, and the $(j+1)$-th control points are $\{p_i^{j+1}\},\ i = 0 \to 3n+3$.

The smoothness of the limit function f generated by (1) and (2) depends on the tension parameter μ. Generally, f is C^2 for $1/15 < \mu < 1/9$, and it is fractal-like for $\mu > 1/9$. Examples of curves generated by the modified 4-point ternary scheme are shown in Fig. 1.



(a) (b) $\mu = 0.066$

(c) $\mu = 0.4$ (d) $\mu = 0.8$

Fig. 1. The modified 4-point ternary scheme for curves: (a) The initial control polygon. (b) The limit curve with $\mu = 0.066$. (c) The limit curve with $\mu = 0.4$. (d) The limit curve with $\mu = 0.8$.

2.2. Multiresolution Analysis

Here we give a brief introduction to the multiresolution analysis construction process. For details, the reader is referred to (Lounsbery *et al.*, 1997; Stollnitz *et al.*, 1996).

The starting point for multiresolution analysis is a nested set of linear function spaces $V^0 \subset V^1 \subset \ldots$, with the resolution of functions in V^j increasing with j. These nested spaces can be constructed by considering

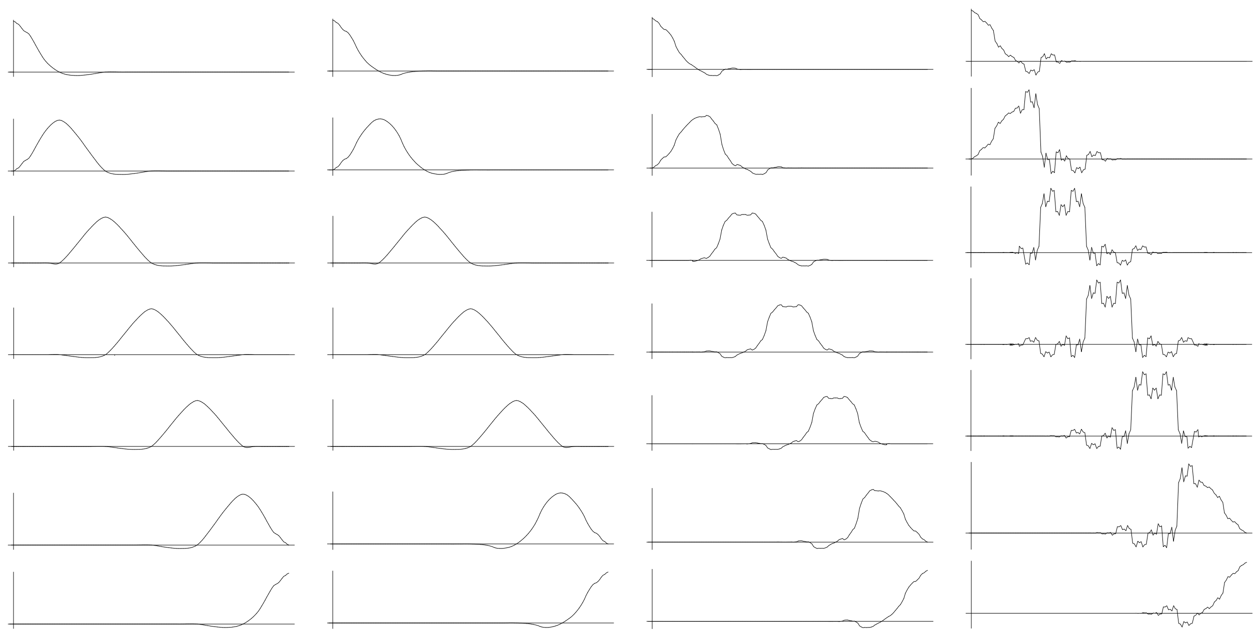


Fig. 2. 4-point ternary scaling functions $\phi_0^j, \phi_1^j, \ldots, \phi_i^j, \ldots, \phi_n^j$ from top to bottom with different values of the tension parameter.

all linear combinations of translated and scaled functions. Let

$$\Phi^j(x) = \begin{pmatrix} \phi_0^j(x) & \phi_1^j(x) & \ldots \end{pmatrix}, \qquad j = 0, 1, \ldots,$$

be the collection of scaling functions and

$$V^j = \operatorname{span}\left\{\phi_0^j(x), \phi_1^j(x), \ldots\right\}.$$

Then nesting these spaces is equivalent to the fact that the scaling functions are refinable, i.e.

$$\Phi^j(x) = \Phi^{j+1}(x) S^j. \tag{3}$$

The next step in multiresolution analysis is to define wavelet spaces also called orthogonal complement spaces, denoted by W^j. The inner product is used to define W^j as

$$W^j = \left\{ f \in V^{j+1} \mid \quad \langle f, g \rangle = 0, \quad \forall\, g \in V^j \right\},$$

where the inner product is

$$\langle f, g \rangle = \int f(x) g(x)\,\mathrm{d}x.$$

The set of functions that span wavelet spaces are called wavelets. The end point for multiresolution analysis is that the analysis and synthesis filters associated with wavelets are constructed and applied in linear time. Mallat (1989) provides a convenient framework to develop the analysis and synthesis filters.

3. Ternary Wavelets

In the previous section, we have introduced the general framework of multiresolution analysis. In the following, our first step is to define the scaling functions for a nested set of function spaces. Then we will construct ternary wavelets.

There is a straightforward recipe for computing scaling functions of ternary wavelets: simply run the modified 4-point ternary scheme starting with a sequence of values $c_k^j = \delta_{i,k},\ k = 0, 1, \ldots, n$ on level j. The limit functions ϕ_i^j are the scaling functions for the ternary wavelets. Figure 2 shows the scaling functions for different tension parameter values. We see that the smoothness of the scaling functions depends on the tension parameter μ when it exceeds 0.11 and then the scaling functions are fractal-like signals. As we know, during the subdivision process by the modified 4-point ternary scheme, at each stage we keep all the old vertices and insert two new vertices “in between” the old ones. The new vertices can be classified into odd and even vertices because these vertices are inserted by using two different subdivision rules. Therefore it is easy to classify the vertices into three categories (old, odd new and even new). If S^j is a subdivision matrix for the modified 4-point ternary scheme and S_O^j, $S_{N_1}^j$ and $S_{N_2}^j$ represent the portions of the subdivision matrix which weight the ‘old’, ‘odd new’ and ‘even new’ vertices, respectively, then S^j can be written down in the block form as

$$S^j = \begin{pmatrix} S_O^j \\ S_{N_1}^j \\ S_{N_2}^j \end{pmatrix},$$

where S_O^j is the identity matrix, $S_{N_1}^j$ and $S_{N_2}^j$ having the following form:

$$\begin{pmatrix} 3/4 & 1/4 & & & & & \\ a_0 & a_1 & a_2 & a_3 & & & \\ & a_0 & a_1 & a_2 & a_3 & & \\ & & & \ddots & & & \\ & & a_0 & a_1 & a_2 & a_3 & \\ & & & a_0 & a_1 & a_2 & a_3 \\ & & & & & 3/4 & 1/4 \end{pmatrix},$$

$$\begin{pmatrix} 1/4 & 3/4 & & & & & \\ a_3 & a_2 & a_1 & a_0 & & & \\ & a_3 & a_2 & a_1 & a_0 & & \\ & & & \ddots & & & \\ & & a_3 & a_2 & a_1 & a_0 & \\ & & & a_3 & a_2 & a_1 & a_0 \\ & & & & & 1/4 & 3/4 \end{pmatrix},$$

respectively. The columns of $S_{N_1}^j$ and $S_{N_2}^j$ are sparse. The first and last three columns of $S_{N_1}^j$ and $S_{N_2}^j$ are relatively different, but the remaining interior columns are shifted versions of the column 4 of their respective matrices $S_{N_1}^j$ and $S_{N_2}^j$. Blanks entries are taken to be zero, and the dots indicate that the previous column is repeated, shifted down by one row each time. This phenomena reflect the fact that ternary basis functions are locally supported.

We can further split the subdivision matrix S^j into two submatrices

$$S_s^j = \begin{pmatrix} S_O^j \\ S_{N_s}^j \end{pmatrix}, \quad s = 1, 2.$$

Similarly, we can write the collection of scaling functions in the block form as

$$\Phi^{j+1}(x) = \left(O^{j+1}(x) \quad N_1^{j+1}(x) \quad N_2^{j+1}(x)\right),$$

where $O^{j+1}(x)$ consists of all scaling functions $\phi_i^{j+1}(x)$ associated with the old vertices of the coarse polygon, $N_1^{j+1}(x)$ and $N_2^{j+1}(x)$ consist of the remaining scaling functions associated with the 'odd new' and 'even new' vertices, respectively, added when obtaining a refined polygon from a coarse polygon. Further, we can split $\Phi^{j+1}(x)$ into the block form

$$\Phi_s^{j+1}(x) = \left(O^{j+1}(x) \quad N_s^{j+1}(x)\right), \quad s = 1, 2.$$

Equation (3) can be decomposed into the block matrix form

$$\Phi_s^j(x) = \left(O^{j+1}(x) \quad N_s^{j+1}(x)\right) \begin{pmatrix} S_O^j \\ S_{N_s}^j \end{pmatrix}, \quad s = 1, 2.$$

Now, we are in a position to define the basis of two wavelet spaces, called the 'odd ternary' and 'even ternary' wavelets spaces, denoted by W_1^j and W_2^j respectively, which are orthogonal complements of V^j in V^{j+1}. The projection of $N_s^{j+1}(x)$ onto W_s^j will give us an orthogonal basis $\Psi_s^j(x) = \{\psi_{s_i}^j(x)\}$ for W_s^j. This basis can be expressed in the matrix form

$$\Psi_s^j(x) = N_s^{j+1}(x) - \Phi_s^j(x)\alpha_s^j, \quad s = 1, 2. \tag{4}$$

The coefficients α_s^j are the solution to the linear system formed by taking the inner products of each side of (4) with $\Phi_s^j(x)$ and using the fact that $\langle \Phi_s^j(x), \Psi_s^j(x) \rangle = 0$:

$$\langle \Phi_s^j(x), \Phi_s^j(x) \rangle \alpha_s^j = (S_s^j)^T \langle \Phi_s^{j+1}(x), N_s^{j+1}(x) \rangle, \quad s = 1, 2. \tag{5}$$

The synthesis filters S_s^j and Q_s^j of odd ternary and even ternary wavelets are defined by the matrix relation

$$\left(S_s^j \quad Q_s^j\right) = \begin{pmatrix} I & -\alpha_s^j \\ S_{N_s}^j & I - S_{N_s}^j \alpha_s^j \end{pmatrix}$$

and the analysis filters A_s^j and B_s^j are

$$\begin{pmatrix} A_s^j \\ B_s^j \end{pmatrix} = \begin{pmatrix} I - \alpha_s^j S_{N_s}^j & \alpha_s^j \\ -S_{N_s}^j & I \end{pmatrix}.$$

So far, we have presented two sequences of orthogonal wavelets named the odd ternary and even ternary wavelets. Unfortunately, like other existing wavelets, the synthesis and analysis filters of odd ternary and even ternary wavelets are not sparse enough, either, which increases the computational time in practical problems. As orthogonality is not the only desirable property in a wavelets spaces, compact support, smoothness and symmetry are sometimes more important in practice. Thus, if we desire efficient, smooth, and symmetric wavelets with compact support, we will have to sacrifice orthogonality. That is, we no longer require the wavelets $\Psi_s^j(x)$ to be orthogonal to V^j, but to preserve good approximation properties, we require the wavelets to be as orthogonal as possible subject to the linear time requirement. Such a kind of wavelets are called biorthogonal wavelets. The work (Lounsbery *et al.*, 1997) gives a method called *priori* to construct biorthogonal wavelets. The idea is that, for each $\psi_{s_i}^j(x) \in \Psi_s^j(x)$, those members of $\Phi_s^j(x)$ whose supports are sufficiently distant from the support of $N^{j+1}_{s_i} \in N^{j+1}_s$ have their corresponding coefficients in the i-th column of α^j_s set to zero. The remaining non-zero coefficients can be determined by solving a smaller, local variant of (5). For the modified 4-point ternary scheme, the support of $\phi^j_{s_i}(x)$ belongs to $[i-3, i+3]$. We then take the wavelets $\psi^j_{s_i}(x)$ to be

$$\psi^j_{s_i} = N^{j+1}_{s_i} - \left(\phi^j_{s_{i-d}}, \dots, \phi^j_{s_{i+d}}\right) \left(\alpha_{s_{-d}}, \dots, \alpha_{s_d}\right)^T,$$

where $\left(\alpha_{s_{-d}}, \dots, \alpha_{s_d}\right)$ are the non-zero entries of the i-th column of α^j_s and d is the d-disc around a vertex v, defined to be the set of all vertices reachable from v by following d.

For ternary wavelets we simply take the following wavelets:

$$\psi^j_i = N^{j+1}_i - \left(\phi^j_{i-d}, \dots, \phi^j_{i+d}\right) (\alpha_{-d}, \dots, \alpha_d)^T,$$

where $(\alpha_{-d}, \dots, \alpha_d)$ are the non-zero entries of the i-th column of α^j and N^{j+1}_i are the scaling functions corresponding to new vertices. The synthesis and analysis filters are defined by

$$\begin{pmatrix} S^j & Q^j \end{pmatrix} = \begin{pmatrix} I & -\alpha^j \\ S^j_N & I - S^j_N \alpha^j \end{pmatrix},$$

$$\begin{pmatrix} A^j \\ B^j \end{pmatrix} = \begin{pmatrix} I - \alpha^j S^j_N & \alpha^j \\ -S^j_N & I \end{pmatrix},$$

respectively, where S^j_N represents the portion of the subdivision matrix which weighs the new vertices. For $\mu = 0.066$ and $d = 4$, the matrices α^j and $\alpha^j_s, s = 1, 2$ are as follows:

$$\alpha^0 = \beta \begin{pmatrix} 4874 & 685 & -889 & -256 & 279 & 57 & -121 & -4 \\ 585 & 4212 & 2907 & 749 & -893 & -203 & 398 & 12 \\ 11 & -977 & 888 & 3244 & 3244 & 888 & -977 & 11 \\ 12 & 398 & -203 & -893 & 749 & 2907 & 4212 & 585 \\ -4 & -121 & 57 & 279 & -256 & -889 & 685 & 4874 \end{pmatrix},$$

$$\alpha^0_1 = \beta \begin{pmatrix} 4874 & -889 & 279 & -121 \\ 585 & 2907 & -893 & 398 \\ 11 & 888 & 3244 & -977 \\ 12 & -203 & 749 & 4212 \\ -4 & 57 & -256 & 685 \end{pmatrix},$$

$$\alpha^0_2 = \beta \begin{pmatrix} 685 & -256 & 57 & -4 \\ 4212 & 749 & -203 & 12 \\ -977 & 3244 & 888 & 11 \\ 398 & -893 & 2907 & 585 \\ -121 & 279 & -889 & 4874 \end{pmatrix},$$

$$\alpha^1 = \beta \left(\begin{array}{c} \begin{matrix} 4874 & 686 & -890 & -258 & 275 & 55 & -102 & -22 & & \\ 585 & 4211 & 2908 & 754 & -881 & -193 & 331 & 70 & -110 & -24 \\ 11 & -967 & 881 & 3216 & 3222 & 909 & -887 & -189 & 334 & 71 \\ 13 & 404 & -199 & -890 & 897 & 3193 & 3196 & 902 & -878 & -187 \\ -3 & -129 & 73 & 338 & -185 & -877 & 901 & 3195 & 3195 & 902 \\ & & -25 & -111 & 70 & 334 & -187 & -878 & 902 & 3195 \\ & & & & -24 & -110 & 71 & 334 & -187 & -878 \\ & & & & & & -24 & -110 & 71 & 334 \\ & & & & & & & & -24 & -110 \end{matrix} \quad \ddots \\ \ddots \quad \begin{matrix} -110 & -24 & & & & & & & & \\ 334 & 71 & -110 & -24 & & & & & & \\ -878 & -187 & 334 & 71 & -110 & -24 & & & & \\ 3195 & 902 & -878 & -187 & 334 & 70 & -111 & -25 & & \\ 902 & 3195 & 3195 & 901 & -877 & -185 & 338 & 73 & -129 & -3 \\ -187 & -878 & 902 & 3196 & 3193 & 897 & -890 & -199 & 404 & 13 \\ 71 & 334 & -189 & -887 & 909 & 3222 & 3216 & 881 & -967 & 11 \\ -24 & -110 & 70 & 331 & -193 & -881 & 754 & 2908 & 4211 & 585 \\ & & -22 & -102 & 55 & 275 & -258 & -890 & 686 & 4874 \end{matrix} \end{array} \right),$$

where the blank entries are taken to be zero, and the dots indicate that the previous two consecutive columns are repeated and shifted down by two rows each time. The order of matrix α^j is $(n3^j + 1) \times (2n3^j)$, where n is the number of the vertices of the initial polygon,

$$\alpha^1_1 = \beta \begin{pmatrix} 4874 & -890 & 275 & -102 & & & & & & & \\ 585 & 2908 & -881 & 331 & -110 & & & & & & \\ 11 & 881 & 3222 & -887 & 334 & & & & & & \\ 13 & -199 & 897 & 3196 & -878 & & & & & & \\ -3 & 73 & -185 & 901 & 3195 & & -110 & & & & \\ & -25 & 70 & -187 & 902 & & 334 & -110 & & & \\ & & -24 & 71 & -187 & \ddots & -878 & 334 & -110 & & \\ & & & -24 & 71 & & 3195 & -878 & 334 & -111 & \\ & & & & -24 & & 902 & 3195 & -877 & 338 & -129 \\ & & & & & & -187 & 902 & 3193 & -890 & 404 \\ & & & & & & 71 & -189 & 909 & 3216 & -967 \\ & & & & & & -24 & 70 & -193 & 754 & 4211 \\ & & & & & & & -22 & 55 & -258 & 686 \end{pmatrix},$$

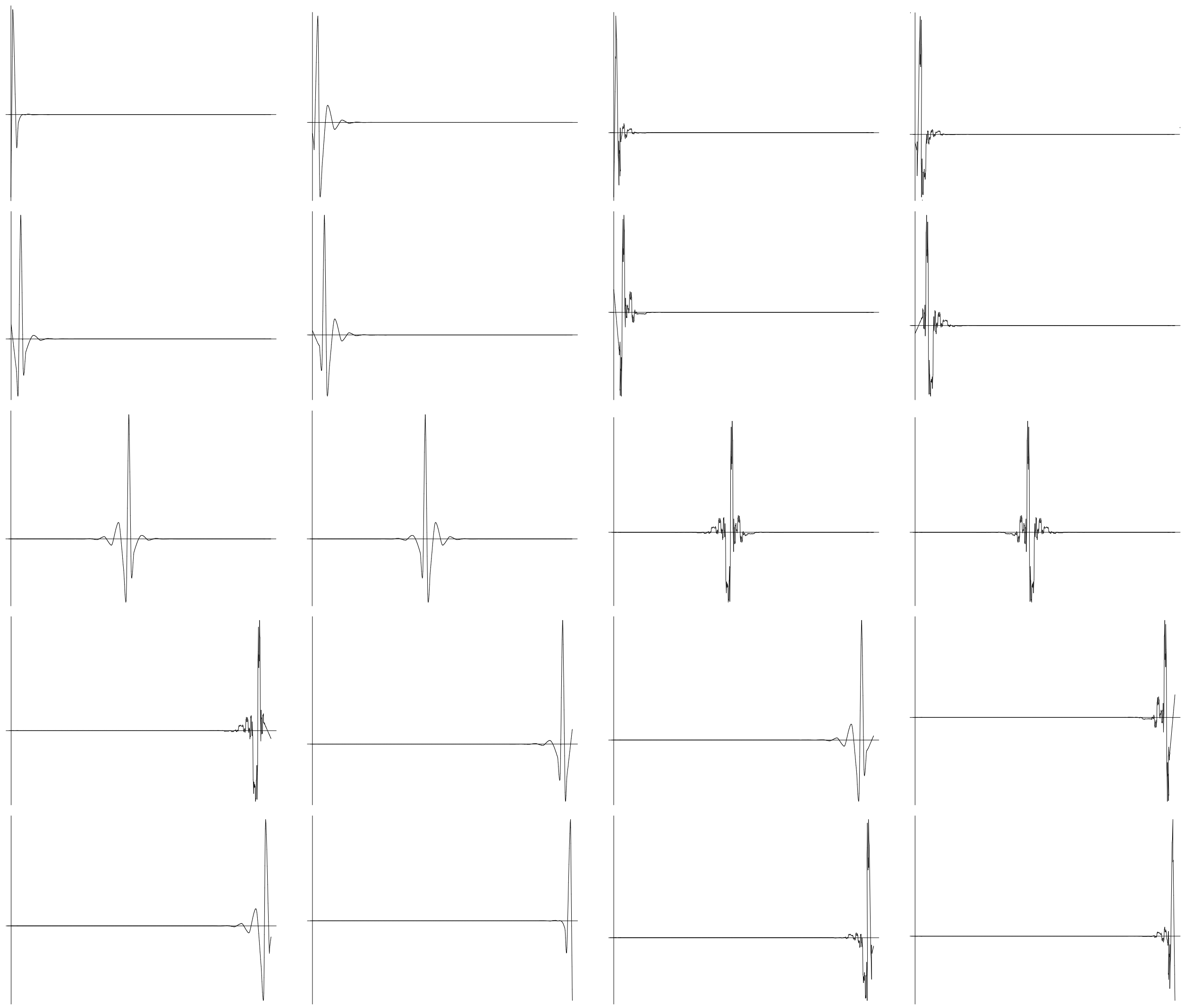
$$\alpha^1_2 = \beta \begin{pmatrix} 686 & -258 & 55 & -22 & & & & & & & \\ 4211 & 754 & -193 & 70 & -24 & & & & & & \\ -967 & 3216 & 909 & -189 & 71 & & & & & & \\ 404 & -890 & 3193 & 902 & -187 & & & & & & \\ -129 & 338 & -877 & 3195 & 902 & & -24 & & & & \\ & -111 & 334 & -878 & 3195 & & 71 & -24 & & & \\ & & -110 & 334 & -878 & \ddots & -187 & 71 & -24 & & \\ & & & -110 & 334 & & 902 & -187 & 70 & -25 & \\ & & & & -110 & & 3195 & 901 & -185 & 73 & -3 \\ & & & & & & -878 & 3196 & 897 & -199 & 13 \\ & & & & & & 334 & -887 & 3222 & 881 & 11 \\ & & & & & & -110 & 331 & -881 & 2908 & 585 \\ & & & & & & & -102 & 275 & -890 & 4874 \end{pmatrix},$$

where $\beta = 10^{-4}$, the blank entries are taken to be zero, and the dots indicate that the previous column is repeated and shifted down by one row each time. The orders of matrices α^j_1 and α^j_2 are $(n3^j + 1) \times (n3^j)$.

Figure 3 shows the ternary wavelets for different values of the tension parameter. At this point, we have completed the steps in designing the ternary wavelet basis and its filter bank. The next section is devoted to its applications in signal compression.

4. Applications of Ternary Wavelets

The subject of this section is to apply the ternary wavelets and further classified odd ternary and even ternary wavelets in signal compression. The goal of com-



(a) $\mu = 0.066$ (b) $\mu = 0.8$

Fig. 3. Ternary wavelets: (a) with $\mu = 0.066$ and (b) $\mu = 0.8$. The first and third columns represent odd ternary wavelets. The second and fourth columns represent even ternary wavelets.

pression is to express an initial set of data using some smaller set of data, either with or without loss of information. Here, we compress some signals and compare the compression results with other existing wavelets.

Signal 1. Given a fractal-like signal generated by the Weierstrass function

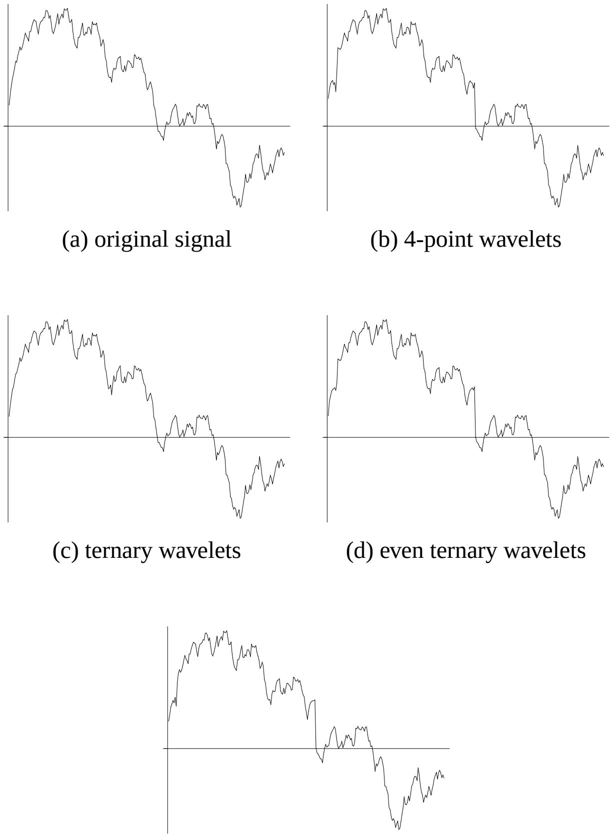
$$W(x) = \sum_{j=1}^{\infty} r^{(s-1)j} \sin(r^j x), \quad r > 1,$$

where $s \in (1, 2)$ is the box dimension of $W(x)$, we compress the signal by 4-point wavelets, ternary, even ternary, and odd ternary wavelets. For a fixed compression ratio of 100:18, the L^2 compression errors for 4-point, even ternary, and odd ternary wavelets are 13.30%, 13.60% and 12.79% respectively, but for compression ratio 100:17, the L^2 compression error for ternary wavelets is 6.50%. Figure 4 shows the compression results.

Signal 2. Taking a fractal-like signal, for a fixed compression ratio of 100:3, the compression error for 4-point, even ternary, and odd ternary wavelets are 0.76% , 0.77% and 0.75%, respectively, while the compression error for ternary wavelets is 0.08%. The compressed signals with $w = 0.05$ and $\mu = 0.066$ are shown in Fig. 5.

Signal 3. By taking a regular signal generated by the function

$$F(x) = \sum_{j=1}^{10} r^{(j-r)} \cos(r^{(j-r)} x), \quad r = 1.5,$$



(a) original signal (b) 4-point wavelets

(c) ternary wavelets (d) even ternary wavelets

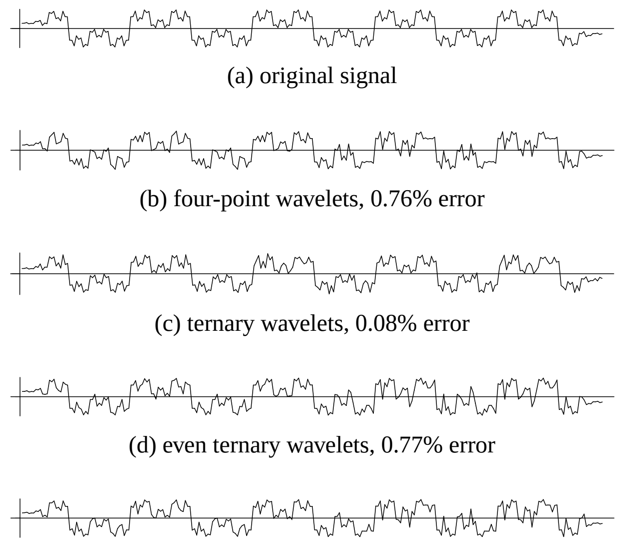
(e) odd ternary wavelets

Fig. 4. Compression results: (a) Original signal, $r = 1.5$ and $s = 1.3$ (b) Signal compressed by 4-point wavelets with weight parameter $w = 0.05$, L^2 error being 13.30% (c) Signal compressed by ternary wavelets with tension parameter $\mu = 0.066$, L^2 error being 6.50%. Moreover, (d) and (e) show the compression result by even ternary and odd ternary wavelets with $\mu = 0.066$, L^2 errors being 13.6% and 12.79%, respectively.

we compress it by 4-point, odd ternary, and even ternary wavelets with compression ratio 100:4 and get the penalty of errors 4.35%, 4.25% and 6.21%, respectively. Figure 6 shows the comparison of the compression results.

Signal 4. Here we compress the regular signal 7(a) by 4-point, odd ternary, and even ternary wavelets. The compression errors with compression ratio 100:50 are 0.33%, 0.25% and 0.46%, respectively. The results are shown in Fig. 7.

From the demonstration of Signals 1 and 2, we have shown that the error in compressing fractal-like signals by ternary wavelets is at least half less than that of the error given by 4-point wavelets.



(a) original signal

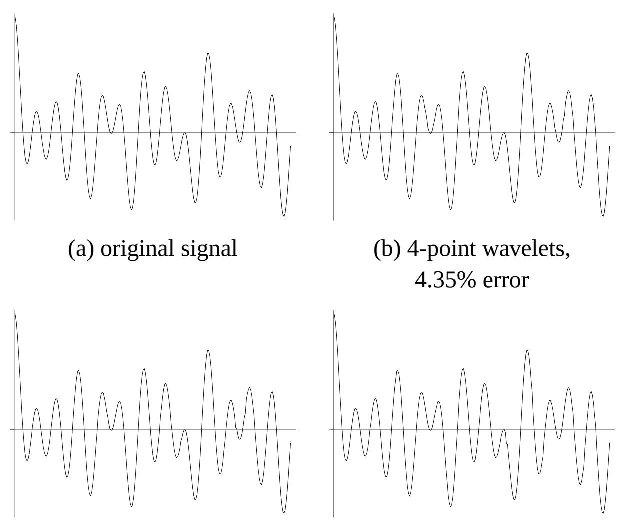
(b) four-point wavelets, 0.76% error

(c) ternary wavelets, 0.08% error

(d) even ternary wavelets, 0.77% error

(e) odd ternary wavelets, 0.75% error

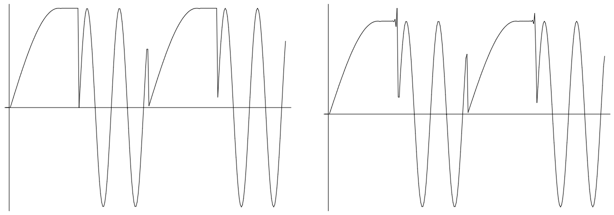
Fig. 5. Signal compression: (a) Original signal (b) Signal compressed by 4-point wavelets, 0.76% level of L^2 error (c) Signal compressed by ternary wavelets, 0.08% level of L^2 error. Moreover, (d) and (e) show the compression results by even ternary and odd ternary wavelets, with errors 0.77% and 0.75%, respectively.



(a) original signal (b) 4-point wavelets, 4.35% error

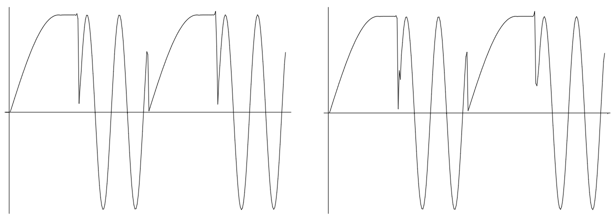
(c) odd ternary wavelets, 4.25% error (d) even ternary wavelets, 6.25% error

Fig. 6. Compression of a regular signal: (a) Original signal (b) Signal compressed by 4-point wavelets, 4.35% level of error (c) Signal compressed by odd ternary wavelets, 4.25% level of error. Moreover, (d) shows the compression result by even ternary wavelets, level of 6.25%.



(a) original signal

(b) 4-point wavelets, 0.33% error



(c) odd ternary wavelets, 0.25% error

(d) even ternary wavelets, 0.46% error

Fig. 7. Compression ratio 100:50: (a) Original signal (b) Compression results by 4-point wavelets, 0.33% level of errors. Moreover, (c) and (d) show compression results by odd ternary and even ternary wavelets with 0.25% and 0.46% errors, respectively.

5. Summary and Future Work

In this paper, based on an interpolating 4-point C^2 ternary stationary subdivision scheme (Hassan *et al.*, 2002), we have introduced ternary wavelets. They are designed for compressing fractal-like signals because of their tightly squeezing property. The error in compressing fractal-like signals by ternary wavelets constitutes at most half of the error produced by 4-point wavelets (Wei and Chen, 2002). However, for compressing regular signals we have further classified these wavelets into 'odd ternary' and 'even ternary' wavelets. Our odd ternary wavelets are better than 4-point wavelets in the sense of compressing both regular and fractal-like signals. There are still some problems for further investigation such as how to find suitable values of the parameter for compressing particular signals. The generalization of this work to higher dimensions for compressing images, fingerprint compression, denoising images, etc. can be considered as a possible direction of future work. As wavelets are widely used in computer graphics and many other areas, we may investigate some other applications of ternary wavelets in these areas.

Acknowledgment

This work is supported by an Outstanding Youth Grant of the NSF of China (No. 60225002), the TRAPOYT at the Higher Education Institute of the MoE of China and the Doctoral Program of the MoE of China (No. 20010358003).

References

Andersson L., Hall N., Jawerth B. and Peters G. (1993): *Wavelets on closed subsets of the real line*, In: Recent Advances in Wavelet Analysis (L.L. Shumaker and G. Webb, Eds.). — New York: Academic Press, pp. 1–61.

Daubechies I. (1988): *Orthogonal bases of compactly supported wavelets*. — Comm. Pure Appl. Math., Vol. 41, No. 7, pp. 909–996.

DeVore R., Jawerth B. and Lucier B. (1992): *Image compression through wavelet transform coding.* — IEEE Trans. Inf. Theory, Vol. 38, No. 2, pp. 719–746.

Dubuc S. (1986): *Interpolation through an iterative scheme.* — J. Math. Anal. Appl., Vol. 114, No. 1, pp. 185–204.

Dyn N., Levin D. and Gregory J. (1987): *A four-point interpolatory subdivision scheme for curve design*. — Comput. Aided Geom. Design, Vol. 4, No. 4, pp. 257–268.

Hassan M.F. and Dodgson N.A., (2001). *Ternary and three-point univariate subdivision schemes.* — Tech. Rep. No. 520, University of Cambridge, Computer Laboratory. Available at `http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-520.pdf`

Hassan M.F., Ivrissimitzis I.P., Dodgson N.A. and Sabin M.A. (2002): *An interpolating 4-point C^2 ternary stationary subdivision scheme.* — Comput. Aided Geom. Design, Vol. 19, No. 1, pp. 1–18.

Liu Z., Gortler S.J. and Cohen M.F. (1994): *Hierarchical space-time control.* — Computer Graphics Annual Conference Series, pp. 35-42.

Lounsbery M., DeRose T.D. and Warren J. (1997): *Multiresolution analysis for surfaces of arbitrary topological type*. — ACM Trans. Graphics, Vol. 16, No. 1, pp. 34–73.

Mallat S. (1989): *A theory for multiresolution signal decomposition: The wavelet representation*. — IEEE Trans. Pattern Anal. Mach. Intell., Vol. 11, No. 7, pp. 674–693.

Stollnitz E.J., DeRose T.D. and Salesin D.H. (1996). *Wavelets for Computer Graphics: Theory and Applications.* — San Francisco: Morgan Kaufmann.

Wei G. and Chen F. (2002): *Four-point wavelets and their applications*. — J. Comp. Sci. Tech., Vol. 17, No. 4, pp. 473–480.

Weissman A. (1990): *A 6-point interpolatory subdivision scheme for curve design*. — M. Sc. Thesis, Tel-Aviv University, Israel.

Received: 25 December 2003