Manuscript Number: SIGPRO-D-17-00185R1

Title: Even/Odd Decomposition Made Sparse: A Fingerprint to Hidden Patterns

Article Type: Research Paper

Keywords: Even/odd transform; Hierarchical Decomposition; Sparsity; Feature extraction

Corresponding Author: Dr. Fabrizio Guerrini, Ph.D.

Corresponding Author's Institution: University of Brescia

First Author: Fabrizio Guerrini, Ph.D.

Order of Authors: Fabrizio Guerrini, Ph.D.; Alessandro Gnutti, M.Sc.; Riccardo Leonardi, Full Professor

Abstract: The very fundamental operation of even/odd decomposition is at the core of some of the simplest information representation and signal processing tasks. So far most of its use has been for rearranging data to provide fast implementations of various types of transforms (Fourier, DCT, ...) or for achieving elementary data transformation, such as the Walsh-Hadamard transforms. This work proposes to look into the decomposition framework to obtain a richer perspective. In the context of an iterated even/odd decomposition, it is possible to pinpoint intermediate layered levels of symmetries which cannot be easily captured in the original data. In addition this determines a hierarchical fingerprinting for any sort of continuous finite support analog signal or for any discrete-time sequence which may turn out useful in several recognition or categorization tasks. It also may help to achieve sparsity within a natural hierarchical framework, which could be easily extented for many other types of orthogonal transformations. This paper also suggests a global measure of the energy imbalance across the hierarchy of the decomposition to capture the overall fingerprinting of this interpretation.

Even/Odd Decomposition Made Sparse: A Fingerprint to Hidden Patterns

Fabrizio Guerrini[1], Alessandro Gnutti[2], Riccardo Leonardi[2]

Department of Information Engineering, University of Brescia, Italy

Abstract

The very fundamental operation of even/odd decomposition is at the core of some of the simplest information representation and signal processing tasks. So far most of its use has been for rearranging data to provide fast implementations of various types of transforms (Fourier, DCT, ...) or for achieving elementary data transformation, such as the Walsh-Hadamard transforms. This work proposes to look into the decomposition framework to obtain a richer perspective. In the context of an iterated even/odd decomposition, it is possible to pinpoint intermediate layered levels of symmetries which cannot be easily captured in the original data. In addition this determines a hierarchical fingerprinting for any sort of continuous finite support analog signal or for any discrete-time sequence which may turn out useful in several recognition or categorization tasks. It also may help to achieve sparsity within a natural hierarchical framework, which could be easily extented for many other types of orthogonal transformations. This paper also suggests a global measure of the energy imbalance across the hierarchy of the decomposition to capture the overall fingerprinting of this interpretation.

Keywords: Even/odd Transform, Hierarchical Decomposition, Sparsity, Feature extraction

2010 MSC: 94A12

[1] EURASIP Member, Corresponding author, fabrizio.guerrini@unibs.it
[2] EURASIP Member

1. Introduction

The need to find particular signal characteristics, e.g. to make classification and/or labeling possible, is inherent in many signal processing applications. These include but are not limited to tasks such as event classification, anomaly detection, denoising, ... [16] [3] [14]. Such a broad class of applications adopts the most diverse technical solutions. It is however possible to enumerate a number of common approaches to such problems, all based on the search for possible hidden patterns in the data, for example the presence of (locally) periodic signals. In many cases, specific patterns are directly looked for thanks to pattern matching techniques [1] or indirectly through correlation-based measures [2]. Other possible approaches exploit specific stochastic properties found in natural data [11]. Feature extraction is a commonly found intermediate step, that is applied either in the original data domain or in a transformed one.

A particular approach, which is not always part of the aforementioned techniques, consists in finding symmetries of some kind that arise naturally for many classes of signals. Interest in symmetry detection exists for many different communities and it is aimed at various signal modalities. Symmetries can be either local or global in nature, and the search methods for these symmetry classes can be quite different (see [13] and [18]). It is well known that exploiting signals' inherent symmetries is an effective way to model the source, which may turn out useful for e.g. information compression [17]. Such alternative information descriptions have been widely adopted, since they allow to condense (sparsify) important properties of the original signal. Often that is why a reversible transformation of the original data is applied to reach this more compact description. Good examples of such transformations are omnipresent: see Fourier and multiresolution transforms [14] [15]. Non linear alternatives exist such as the iterative function system (IFS) paradigm for computing fractal dimensions or near-by signal regeneration [10].

One of the simplest decomposition proposed in signal processing, which is by its very definition constructed upon the signals symmetries, is the even/odd decomposition [15]. It turns out that, given the resulting intuitive geometrical interpretation and the parity preservation of the Fourier transform, even/odd decomposition is quite common in signal processing. It becomes then a natural proposition to try iterating this process to each half of the even/odd parts, which are necessary for reconstruction. The same geometrical interpretation can thus be preserved over the resulting decomposition tree. This in turn gives a peculiar characterization of the signal that is based on how its decomposition tree is shaped.

Therefore, this paper studies how to perform an iterative even/odd decomposition of 1-D signals around their midpoint, which in addition allows for a fast implementation. Such process is possible for both continuous and discrete-time signals and involves only very simple operations at each stage, without increasing the interval support or the number of samples necessary to represent the original signal. It will be shown that the recursive application of the even/odd decomposition for discrete sequences provides results identical to the application of a radix-2 implementation of the Walsh-Hadamard discrete transform (WHT) [5]. On another hand, the iterative nature of the transform allows for a deeper analysis of hidden symmetric patterns in the data during the computation. Such patterns do not correspond to local symmetries but are instead indicative of an even/odd relation between parts of the signal existing at a particular level of the decomposition tree, thus permitting to make decisions, such as to arrest the resulting decomposition tree at an earlier stage, without significantly impairing the quality of the representation. This approach therefore ensures both a fast implementation and an efficient way to detect these peculiar symmetric relations, leading thus to a naturally sparse representation of the decomposition tree. To prove how sparsity in the decomposition tree is an useful signal characterization, a tree sparsity measure is employed to classify broad 1-D signal types.

The rest of the presentation is organized as follows. Section 2 fixes some notation and presents some preliminary processing background by recapping briefly the even/odd decomposition of signals, along with how such operation motivates the considerations put forward in this paper. Section 3 provides a description of the iterative application of the even/odd decomposition to obtain a decomposition tree in the case of continuous time signals. The discussion is extended to the discrete time case in Section 4, leaving the resolution of some caveats that such domain causes till Section 6. The paper follows with some experimental simulation results described in Section 5 and ends by drawing the conclusions in Section 7.

1.1. Contributions

This paper introduces a decomposition tree for finite energy signals using the basic even/odd decomposition as the core decomposition step. It provides a blueprint for its fast implementation for finite, discrete sequences through the recursive application of a "butterfly"-like computation, similar to that employed in the fast computation of the Walsh-Hadamard Transform. As opposed to WHT, the iterative nature of the proposed decomposition allows to analyze every step of the process (the decomposition level), for example to extract features on the tree nodes. This paper is in particular focused on detecting sparsity in the decomposition tree as it is built from the root (the original signal) up to the leaves. Sparsity is measured during the iterative generation of intermediate results rather than relying on the L_0 norm of the leaves of the decomposition tree. Proving how such sparsity can provide a discriminating feature across different data types, it is suggested how such decomposition tree can find a particular type of symmetry in the data, that is not directly connected to local or global symmetries of the whole signal.

2. Motivation and Background

An iterative decomposition process can be undertaken based on the well-known even/odd decomposition basic signal manipulation, sometimes also referred to as the parity decomposition. The even/odd decomposition of a given energy signal $x(t) \in \mathcal{L}^2(\mathbb{R})$ states that $x(t)$ can be expressed as the sum of its even and odd parts, respectively $x_e(t)$ and $x_o(t)$, given by:

$$x_e(t) = \frac{x(t) + x(-t)}{2}; \quad x_o(t) = \frac{x(t) - x(-t)}{2}; \quad x(t) = x_e(t) + x_o(t) \tag{1}$$

The even signal is such that $x_e(t) = x_e(-t)$; the odd signal is such that $x_o(t) = -x_o(-t)$. Since $\mathcal{L}^2(\mathbb{R})$ is a Hilbert space, with inner product $< x(t), y(t) >= \int_{\mathbb{R}} x(t)y^*(t)dt$, such decomposition is possible $\forall\, x(t)$ and represents the vector $x(t)$ as the sum of two orthogonal vectors since the inner product $< x_e(t), x_o(t) >$ is obviously 0.

The energy E is defined as the squared Euclidean norm of the signal $x(t)$ and it is easy to see that:

$$\begin{aligned} E &= \int_{-\infty}^{\infty} |x(t)|^2 dt = \int_{-\infty}^{\infty} |x_e(t) + x_o(t)|^2 dt = \\ &= \int_{-\infty}^{\infty} |x_e(t)|^2 dt + \int_{-\infty}^{\infty} |x_o(t)|^2 dt = E_e + E_o \end{aligned} \tag{2}$$

where E_e and E_o are the energy of the even and odd parts respectively. The last step exploits the orthogonality of $x_e(t)$ w.r.t. $x_o(t)$.

The motivation under our interest in this elementary operation is its ability to readily detect inherent symmetries in the data. In fact, if the original signal $x(t)$ is of an inherently even (resp. odd) shape, the most part of its energy will be carried by its even (resp. odd) component. For example, in Figure 1 the latter case applies: the odd part carries around 70% of the total energy, or more than twice as much as that of the even part.

This characteristic can be generally useful for a number of tasks in signal processing, e.g. it favors a compact representation of the original signal. In fact, if the signal is reconstructed not by summing both parts but just by retaining the one which carries the most energy, the signal can be represented without introducing too much distortion. Obviously, in general for a given signal it is unlikely that such a condition holds after a single decomposition step, unless the signal possesses a very obvious symmetric/antisymmetric shape. For example, in Figure 1 one could hardly imagine to represent $x(t)$ using only $x_o(t)$.

For finite support signals, let us now propose to iterate the analysis, by applying the decomposition on the resulting informative part of the even and odd signals (i.e. their causal part which is recentered around the origin), thus constructing a decomposition tree. As the signals are decomposed again and again through Eq. (1) into pairs of orthogonal vectors, they can be analyzed in turn to exploit the symmetry content description which is inherent in the energy they carry. For example, to continue with the example above regarding the compact representation of a given signal, after a certain number of decompositions it can happen that (at least) one of the constituting signal can be safely discarded because it has energy below a certain tolerance threshold.

Of course, there are many possible strategies to handle the signals resulting from the iterative application of the even/odd decomposition, i.e. the decomposition tree nodes, depending on what is the intended objective. For example, for strongly symmetric data it could happen that, ending the process after a certain number of decompositions, just a small fraction of the tree leaves carry almost all of the energy of the original signal, thus retaining just them to reconstruct the signal leads to a compact representation of it. Hence, we can set such a threshold B, which can be possibly adapted to the decomposition level l being considered. If some of the leaves or nodes carry less energy than B they are

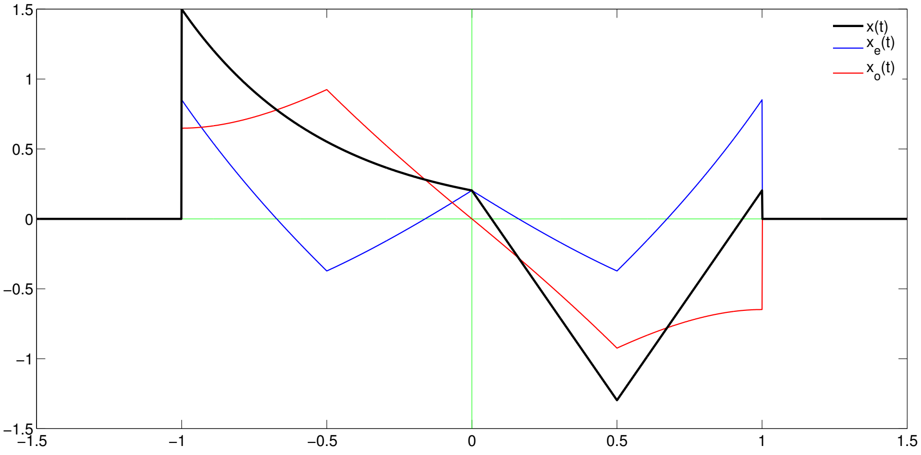


Figure 1: An example of standard parity decomposition. $x(t)$ (black) is the sum of $x_e(t)$ (blue) and $x_o(t)$ (red).

omitted from the reconstruction. If one of the leaves is missing because it has been discarded, only the surviving companion leaf is used to reconstruct the parent tree node.

On the other hand detecting an energy imbalance in some nodes of the decomposition tree, corresponds to a hidden symmetric relation which is present at the corresponding level of the decomposition tree.. The tree nodes can be analyzed as they are created from the top (root) down to the leaves to identify an energy concentration in some of the tree nodes. In this case a threshold S should be set on the ratio between the energies of the children nodes generated after a decomposition step and that of the associated parent node.

In the experimental results that will be shown in Section 5 we show how these strategies work to highlight how to reflect 1-D symmetries in the data. In the same section, we will give more examples of potential applications, for example the one presented in [6], before we draw the conclusions in Section 7.

3. Iterative Even/Odd Decomposition for Continuous Time Signals

In this section we discuss the even/odd decomposition of a finite support, finite energy signal in the continuous-time case, and discuss how it is possible to iterate such an operation to form a decomposition tree of the original signal. The objective is to introduce such a tree detailing the resulting processing. The derived more practical discrete-time case is described in Section 4.

To begin, let us consider a real-valued signal $x(t) \in \mathcal{L}^2(\mathbb{R})$, i.e. a finite energy signal. We shall assume that the signal has finite support, ignoring the case of infinite support signals for reasons that are to be discussed in what follows. We will also assume it is time-centered, i.e. with support $[-T, T]$, without losing generality considering the possibility of zero-padding the signal. The even/odd decomposition has been defined in Section 2 and an example of such decomposition is shown in Figure 1, with $T = 1$.

The pair of signals $x_e(t)$ and $x_o(t)$ obtained through the above decomposition step constitute an alternative representation of $x(t)$. Following Eq. (1), both

$x_e(t)$ and $x_o(t)$ have support $[-T, T]$. However, the new representation does not increase the temporal support needed to reconstruct the original signal. In fact, the parity property of the even and odd parts implies that even if they both have the same support of the original signal, only e.g. their causal part (respectively $x_e^{(c)}(t)$ and $x_o^{(c)}(t)$, both with support $[0, T]$ and respectively defined by $x_e^{(c)}(t) = x_e(t) \cdot \mathbb{1}(t \geq 0)$ and $x_o^{(c)}(t) = x_o(t) \cdot \mathbb{1}(t \geq 0)$) is as informative and thus sufficient to describe the entire signal. In other words, given just the causal part of both the even and odd parts, the anti-causal parts $x_e^{(ac)}(t) = x_e(t) \cdot \mathbb{1}(t < 0)$ and $x_o^{(ac)}(t) = x_o(t) \cdot \mathbb{1}(t < 0)$ can be readily obtaining by suitably mirroring the causal parts[3]:

$$\begin{aligned} x_e(t) &= x_e^{(c)}(t) + x_e^{(ac)}(t) = x_e^{(c)}(t) + x_e^{(c)}(-t) \\ x_o(t) &= x_o^{(c)}(t) + x_o^{(ac)}(t) = x_o^{(c)}(t) - x_o^{(c)}(-t) \end{aligned} \quad (3)$$

The original signal $x(t)$ is then reconstructed as in Eq. (1). By retaining just the causal part of the even and odd parts, their energy is respectively $E_e/2$ and $E_o/2$, and their sum gives $E/2$, because for both signals the anti-causal part has the same exact energy as the causal part, being just a mirrored copy of the latter (plus a further inconsequential change of sign for the odd part).

Figure 2 provides an approximate depiction of the effects of the decomposition process adopting a vector space representation (the approximation derives from representing infinite-dimensional vectors in a 3-D space). The even and odd signals $x_e(t)$ and $x_o(t)$ are orthogonal and sum up to $x(t)$, then multiplying them by the indicator function to take just the causal part again separates them into orthogonal signals, i.e. $x_e^{(c)}(t)$ and $x_e^{(ac)}(t)$ for the even part, because they have disjoint supports. However, what is really worth noting is that the causal parts $x_e^{(c)}(t)$ and $x_o^{(c)}(t)$ are no longer orthogonal.

As mentioned before, our objective is to iterate the even/odd decomposition of Eq. (1). This way we can build a decomposition tree by repeating the decom-

[3] The second equality in Eq. (3) is not exact for $t = 0$, however it is intended to highlight the concept that the anti-causal part can be readily obtained by flipping the causal part.

position process for every sub-even and sub-odd component. This operation is carried only on the causal part of each signal (except of course the root of the tree), i.e. without increasing the alternative representation support. The de-

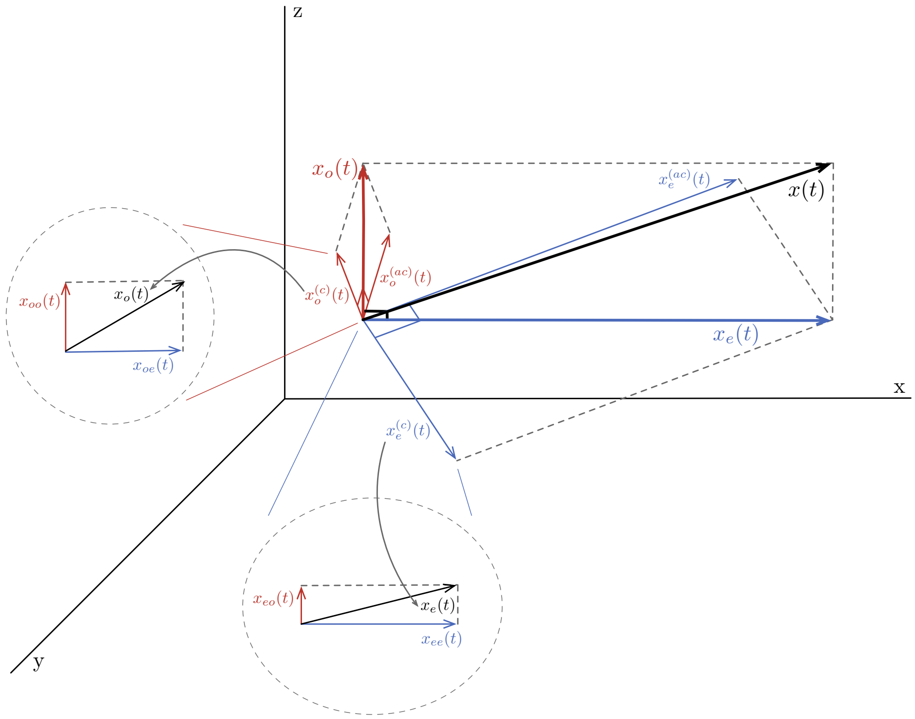
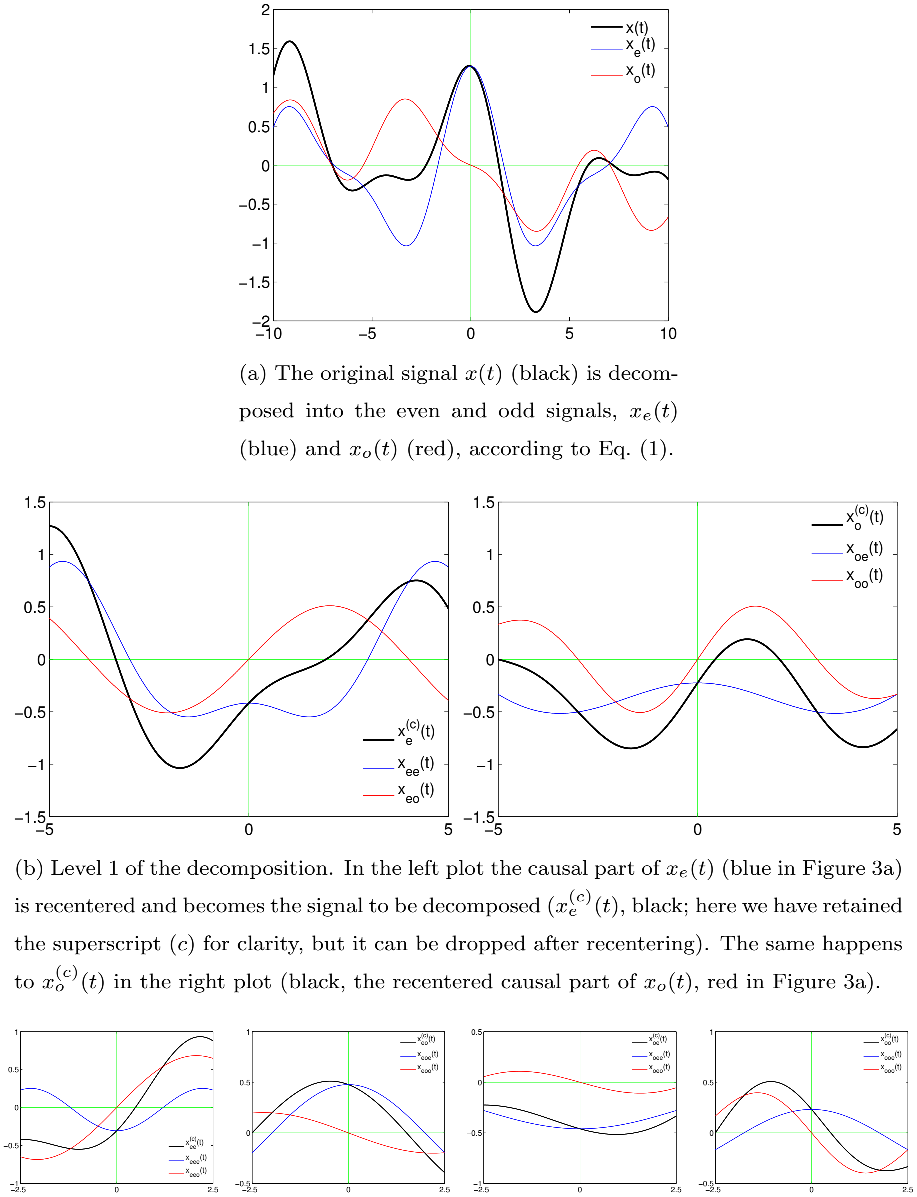


Figure 2: Even/odd decomposition iteratively presented on a 3D plot. The original signal $x(t)$ (black) is decomposed into two orthogonal vectors, the even signal $x_e(t)$ (blue) and the odd signal $x_o(t)$ (red). We have decided to align the x-axis and the z-axis along the direction of $x_e(t)$ and $x_o(t)$ respectively. They are in turn split into their causal and anti-causal part, which are again orthogonal to each other, as described by Eq. (3): the causal parts represent the orthogonal projections on the causal subspace of the original even/odd parts. On such subspace they represent the nodes of the decomposition tree. In the ovals, we sketched the next iteration of the decomposition, applied to the causal parts of $x_e(t)$ and $x_o(t)$ (the superscript is dropped in the ovals showing the next level of decomposition since the signals must be recentered).

composition after a single iteration applied just to the causal parts in abstract vector form is shown in the ovals of Figure 2. As we mentioned, the causal parts determine orthogonal projections on the causal subspace of the original even/odd parts. However, in such a subspace they are no longer orthogonal to each other and they can be considered as new, “independent” signals.

To start the iterative process, Eq. (1) constitutes the first level of the transform, and $x(t)$ is the root of the tree. The iteration then goes as follows: the signals $x_e^{(c)}(t)$ and $x_o^{(c)}(t)$, the causal parts of the output of Eq. (1) and the first level ($l = 1$) children nodes, can be further decomposed as well into their even and odd parts. Of course, before each decomposition of the causal part of each output signal (i.e. a node), the latter has to be time-recentered by shifting it by $T/2^l$ (to the left if the causal part is retained as we assumed) before iterating the decomposition. Thus, after iterating the decomposition again four new children nodes in the second level of the transform ($l = 2$) are obtained, constituted by respectively $x_{ee}(t)$, i.e. the even part of the time-recentered causal even part, $x_{eo}(t)$, $x_{oe}(t)$ and $x_{oo}(t)$ (the subscripts having the appropriate meaning, i.e. listing which component to retain after each decomposition read from left to right, and dropping the (c) superscript after recentering) and then just the causal part is again retained. In turn, they can be decomposed again into their own even and odd parts (after recentering) and so on. The new representation obtained after each decomposition still does not increase the temporal support needed to reconstruct the original signal. In fact, as it was the case with a single decomposition, only the causal parts of each obtained signal after the decomposition step is necessary. Figure 3 depicts an example of an iterative decomposition process, stopped at the second level, using a signal defined in the finite support $[-10, 10]$. Note how at each level the number of signals doubles but the support is halved by retaining just its causal part.

It is evident that in the case of continuous-time signals the iterative decomposition goes on forever, with ever-shrinking node supports. In general, level l of the transform tree is constituted by 2^l signals, or tree nodes, with support $[0, T/2^{l-1}]$. Theoretically, there is no limit on the number of decomposition lev-



(a) The original signal $x(t)$ (black) is decomposed into the even and odd signals, $x_e(t)$ (blue) and $x_o(t)$ (red), according to Eq. (1).

(b) Level 1 of the decomposition. In the left plot the causal part of $x_e(t)$ (blue in Figure 3a) is recentered and becomes the signal to be decomposed ($x_e^{(c)}(t)$, black; here we have retained the superscript (c) for clarity, but it can be dropped after recentering). The same happens to $x_o^{(c)}(t)$ in the right plot (black, the recentered causal part of $x_o(t)$, red in Figure 3a).

(c) Level 2 of the decomposition. From left to right, the decomposition of respectively $x_{ee}^{(c)}(t)$, $x_{eo}^{(c)}(t)$, $x_{oe}^{(c)}(t)$ and $x_{oo}^{(c)}(t)$ is depicted. Each of these is again the causal, recentered part of the corresponding signals in the first level of Figure 3b.

Figure 3: Even/odd decomposition tree of a signal $x(t)$, illustrated from the root (top) to the second level (down).

els, although the decomposition process can be of course arrested at any desired level, *e.g.* if the support becomes smaller than a chosen quantity.

Supposing to arrest the decomposition at a given level, the complementary reconstruction process, which iterates a single reconstruction step, consists of first mirroring the even and odd signals around the time origin (retaining and changing the sign respectively, as in Eq. (3)) and then adding them to obtain the parent node on the precedent level. The reconstruction process can also be observed from the decomposition shown in Figure 3: starting from a given level, it consists of summing the blue and red signals to obtain the black signal and then copying it to the above level to form the new blue or red signal (depending on the particular node considered), after it has been shifted and mirrored accordingly.

As a side note, if the original signal has infinite temporal support, the initial decomposition of Eq. (1) produces even and odd parts with infinite support as well, and again the informative part is contained in the causal support $[0, +\infty]$ that retains half of the energy of the original signal, so this principle holds for infinite support signals as well. However, it is not feasible to iterate the decomposition since it is impossible to recenter the even and odd parts around the time origin (and padding can not help either). Therefore, for the scope of this paper we are considering only finite support signals.

4. Iterative Even/Odd Decomposition for Discrete Time Signals

The discussion of Section 3 applies to the discrete case as well, provided that some additional care is used in handling how the energy is distributed (an important fact later) and interpreting the supports. To keep the presentation as streamlined as possible, we will discuss such details in Section 6. For now, let us briefly restate the problem from the beginning in the discrete-time setting. Consider a discrete-time, real-valued, finite-energy sequence $x[n] \in \mathcal{L}^2(\mathbb{Z})$, with support $-N, \ldots, N$. This choice implies that L, the length of $x[n]$, is odd, namely $L = 2N + 1$, which is still general if one assumes to pad with a single 0 even-length sequences. Now the even-odd decomposition is described as:

$$x_e[n] = \frac{x[n] + x[-n]}{2}; \quad x_o[n] = \frac{x[n] - x[-n]}{2}; \quad x[n] = x_e[n] + x_o[n] \qquad (4)$$

In this case, the informative support, that is the one which carries "unique" samples and not simply mirrored ones, for the even part is $0, \ldots, N$ and that of the odd part is $1, \ldots, N$ (since $x_o[0] = 0$ by definition, which is non informative). So, what is needed for reconstructing the original sequence $x[n]$ is $x_e^{(c)}[n]$ which is $x_e[n]$ limited to the $0, \ldots, N$ support and $x_o^{(c)}[n]$ which is $x_o[n]$ limited to the $1, \ldots, N$ support. The formal way to obtain $x_e[n]$ and $x_o[n]$ from $x_e^{(c)}[n]$ and $x_o^{(c)}[n]$ is discussed in Section 6, but for now it can be highlighted that even if the decomposition is unbalanced as the informative support of the even part is $N + 1$ samples versus the N samples for the odd part, still the decomposition does not increase the support needed for the original signal reconstruction.

Alternatively, for even-length sequences one can divide the original sequence into a "right" part and a "left" part of length $L/2$. The causal even sequence is then the semi-sum of the right part with the flipped version of the left part, while the causal odd sequence is the semi-difference of the two, and both therefore have $L/2$ samples. This abstraction, valid for L even, is simpler to understand and allows to perform the decomposition without zero-padding. Again, more details can be found in Section 6.

In any case, Eq. (2) still holds, provided the integration operator is substituted by the summation one:

$$E = \sum_{n=-N}^{N} |x_e[n] + x_o[n]|^2 = \sum_{n=-N}^{N} |x_e[n]|^2 + \sum_{n=-N}^{N} |x_o[n]|^2 = E_e + E_o \qquad (5)$$

The iteration of the basic even/odd decomposition is possible for discrete time signals as well, with an important difference. Given the discrete nature of the signal domain, the iteration is bound to stop when single-sample sequences are encountered, a situation not found for the continuous time signals considered in Section 3. Therefore, decomposition trees for discrete-time finite support signals are always finite.

Obviously, the need to iterate the even-odd decomposition can be controlled by checking suitable parameters, such as the energy of the sequence being decomposed. In fact, it is entirely possible to arrest a subtree if the energy involved are under a user-defined threshold, as we mentioned in Section 2. We will explore how to control the decomposition process with respect to the intended objective of the whole process in Section 5.

The iterative decomposition of a discrete-time signals and the complementary reconstruction process are as follows. First, let us assume that L is even and in particular a power of 2, $L = 2^m$. In this case the decomposition comprises m levels (assuming that no energy thresholding is applied during the decomposition process and thus the tree is completely developed all the way to one-sample leaves), and the nodes in the decomposition tree at each level have all the same length. After the first decomposition step, the even and odd parts, corresponding to the first level nodes in the tree, are $L/2$ samples long. In general, nodes of level l always have the same length $L/2^l = 2^{m-l}$, that is half of that of the parent node. At the m-th and last level there are 2^m leaves, i.e. nodes with unitary length, that cannot be decomposed anymore. In this case, the decomposition tree is perfectly balanced, i.e. it forms a binary tree.

For example, if $L = 4 = 2^m$, with $m = 2$, the decomposition tree is 2-levels deep. The first level is obtained dividing $x[n]$ into a left part (the first two samples, let us say $x[1]$ and $x[2]$) and a right part (the last two samples, $x[3]$ and $x[4]$). In this case, the (strictly) causal part of $x_e[n]$ is $[x(3){+}x(2)\ \ x(4){+}x(1)]/2$ and of $x_o[n]$ is $[x(3){-}x(2)\ \ x(4){-}x(1)]/2$ (the left part is flipped as we mentioned before). Therefore, the two first-level nodes have a length of $2^{2-1} = 2$. The second level of the transform takes the semi-sum and the semi-difference of the two samples of $x_e[n]$ and $x_o[n]$, which is what happens if the right-left separation is applied again, and outputs the final 4 leaves.

Even if it is not immediately apparent, one can show that in the above case of a discrete-time signal defined in $1, \ldots, L$ with L equal to a power of 2, the recursive even-odd decomposition is strictly related to the Walsh-Hadamard transform of the signal [5]. Both can be obtained through a fast implementation through a classic “butterfly” diagram, the only difference being a permutation of the output sequence that involves bit-reversing of the output indices. To prove that, it is beneficial to start from the basic operation on a 2-sample input. Let us assume that $x[n]$ is a column vector, $\mathbf{x} = (x[1]\ x[2])^T$ and that the output vector $\mathbf{y} = (y[1]\ y[2])^T$ is also a column vector with the even signal written before the odd one, that is $\mathbf{y} = (x_e^{(c)}[1]\ x_o^{(c)}[1])^T$. Remembering to take the “causal” (right) part, $x_e^{(c)}[1] = (x[1] + x[2])/2$ and $x_o^{(c)}[1] = (x[2] - x[1])/2$, namely the even signal is the semi-sum of the input samples and the odd signal is their semi-difference, we can write the decomposition step as:

$$\mathbf{y} = \frac{1}{2} \cdot \mathbf{D_2} \cdot \mathbf{x} \quad \text{with} \quad \mathbf{D_2} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \tag{6}$$

By iterating this operation, for an input $\mathbf{x}$ with $L = 2^m$ samples, the output is:

$$\mathbf{y} = \frac{1}{2^m} \cdot \mathbf{D_{2^m}} \cdot \mathbf{x} \quad \text{with} \quad \mathbf{D_{2p}} = \begin{bmatrix} \mathbf{D_p F} & \mathbf{D_p} \\ -\mathbf{D_p F} & \mathbf{D_p} \end{bmatrix} \tag{7}$$

where $\mathbf{F}$ is the flipping matrix. The order of the leaves in $\mathbf{y}$ reflects what is produced by Eq. (6). It is obtained by writing the index in binary form and considering the successive binary figures as 0 for the even node and 1 for the odd node. Comparing the matrix $\mathbf{D_{2^m}}$ with the Walsh matrix of the same order, with the rows taken in sequential order, we find that the former can be obtained from $\mathbf{W_{2^m}}$ through the following process:

- Write the row number of $\mathbf{W_{2^m}}$ and $\mathbf{D_{2^m}}$ in binary form, using m bits;
- Associate each row index of $\mathbf{W_{2^m}}$ with the bit-reversed row index of $\mathbf{D_{2^m}}$;
- Change the sign of the bottom half of $\mathbf{D_{2^m}}$.

This is true for every $m = \log_2 L$. An alternative way of stating this fact is to say the rows of $\mathbf{D_{2^m}}$ are a permutation of the rows of $\mathbf{W_{2^m}}$. Exactly like the rows of $\mathbf{W_{2^m}}$, the rows of $\mathbf{D_{2^m}}$ are linearly independent.

Setting L as a power of 2 guarantees that in every step of the decomposition the length of the resulting sequences is always even. For even sequences, the left-right separation always applies (see also Eq. (10) in Section 6). However, if the original sequence length is even but not a power of 2, or odd altogether, it is unavoidable that some nodes in the tree have odd length.

In particular, let us say that a particular node (even the starting signal) has odd length L. In that case, the decomposition tree is slightly unbalanced, since the even sequence is longer than the odd sequence by 1 sample. For example, if $L = 3$, the even sequence is of length 2, and can be further decomposed into two second-level leaves of length 1, whereas the odd sequence corresponds to a first-level leaf of length 1. Therefore, for a general value of L the decomposition tree is only approximately balanced since the penultimate level nodes may be leaves or length-2 nodes, and furthermore its nodes at some other level may be sequences of different lengths by 1 sample.

Since the children nodes have approximately half the samples of the parent node, the number of levels of the decomposition tree is $\lceil \log_2(L) \rceil$, and as mentioned above is exactly equal to $\log_2(L) = m$ in case $L = 2^m$. The expected number of levels is just an upper bound, as it has been hinted above, since setting a threshold B to the energy of a node may affect the depth of the tree.

5. Simulation Results

In this Section we describe a series of experiments meant to show how we propose to use the decomposition tree, exploiting the information carried by the constructed hierarchy.

The experiments have been carried out on a variety of 1-D digital signals obtained from various media. In particular, we have used a random set of rows from 4 standard images (*Lena*, *Cameraman*, *Mandrill* and *Tiffany*), in 512×512 format; various audio tracks consisting of (a) modern pop songs, (b) hip-hop beats and (c) vocal sound effects, sampled at CD quality (44.1kHz, 16 bits per sample); a selection of ECG signals taken from the PhysioNet database [7]; and seismic data from the IRIS database [9]. We have also used a number of texture patterns taken from [4], some clearly symmetric and some not quite so, as a testbed to exemplify different scenarios. To fairly compare these data, we have taken uniformly sized data windows having the same length of the images' rows, $N = 512$; in the case of audio, these windows were randomly picked at least $5s$ apart. The rationale behind such a diversified testbed is to analyze the decomposition tree for signals exhibiting diverse characteristics in terms of evolution in time/space.

First, let us show how compact is the representation obtained with the decomposition tree. Figure 4 depicts the performance for the proposed method versus that of a couple of basic transforms, the DCT and the DFT. The results are given in aggregate form, that is averaging the performance across the songs, ECG and image rows data categories (note that in the symmetric textures case the proposed method is clearly favored). The comparison has been done as follows: first, compute the whole decomposition tree (which is $\log_2 N$ levels deep) and then sort the N leaves of the last level according to their decreasing magnitude. Then, discard a given percentage T of the least significant leaves and compute the ratio of the energy of the reconstructed signal and that of the

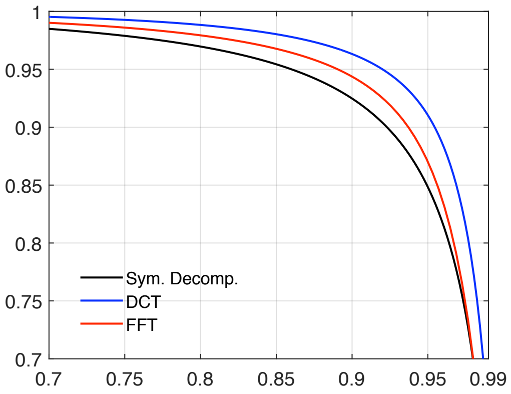


Figure 4: Comparison between the representation compactness of the proposed technique and those of the basic DCT and DFT transforms. The x-axis represents the percentage of discarded coefficients, T. The y-axis represents the energy ratio of the reconstructed signal w.r.t. the energy of the original sequence.

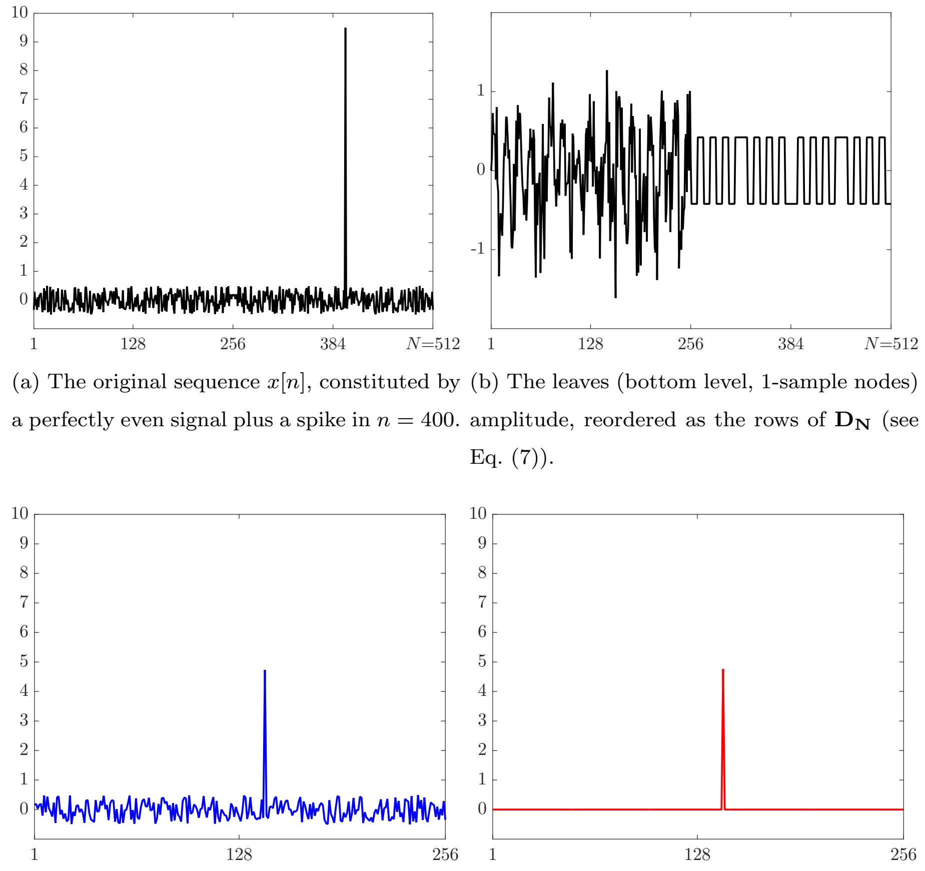
original signal. The same is performed for the other transforms, where instead of the leaves we discarded the least significant frequency coefficients in terms of their magnitude.

As Figure 4 shows, the compactness of the representation given by the leaves of the decomposition tree is slightly worse than that of the DCT or the DFT. Recall that when the decomposition tree has been constructed for all levels, the reached decomposition produces the Walsh-Hadamard transform, which is known not to be a decorrelating transform for the considered data.

However, the hierarchy in the decomposition tree can help to discover if some sort of hidden, inner sparsity is present in the original signal. To clarify, let us construct the following simple example as shown in Figure 5. Figure 5a depicts a purely even signal with a very obvious spike added at $n = 400$. If a frequency based representation is employed, e.g. DCT, such spike would be spread among all frequencies. Using instead the proposed decomposition scheme, this spike will likewise be spread among the leaves. In Figure 5b we have depicted the magnitude of the leaves at the bottom level of the tree, indexed in the same fashion as in Eq. (7), where the spread of the spike appears in the right subtree (which is the decomposition of the odd signal in the first level). However, intermediate-level nodes can actually, at some point, sparsely identify such a spike. In this example, this occurs in the first level of the decomposition. In fact, in Figure 5d the node representing the odd signal after the first decomposition has actually just a single non-zero sample.

It is therefore clear that, during the decomposition process, it is possible to detect if a node exhibits good sparsity by employing appropriate sparsity measures, e.g. the Gini index [8]. In such cases, the decomposition of a node can be avoided when the Gini index is greater than a prefixed threshold, effectively pruning the subtree to greatly increase the representation sparsity. This is also true if the sparsity properties of the symmetries within the original signal are not immediately visible as the one presented in Figure 5. An alternative strategy is to compute the Gini index for the entire decomposition tree to identify the sparser nodes: in this case, the objective is to make the decomposition sparsity even more effective, by verifying if any additional decomposition of a moderately sparse node can yield an even sparser representation at some level in its subtree, at the price of added complexity.

Instead of directly evaluating the sparsity of the decomposition tree analyzing the sparsity in node signals through the Gini index, we propose an alternative approach that is theoretically more connected to the essence of the



(a) The original sequence $x[n]$, constituted by a perfectly even signal plus a spike in $n = 400$.

(b) The leaves (bottom level, 1-sample nodes) amplitude, reordered as the rows of $\mathbf{D_N}$ (see Eq. (7)).

(c) The causal part of the even signal $x_e[n]$.

(d) The causal part of the odd signal $x_o[n]$, with just a non-zero sample corresponding to the spike.

Figure 5: A synthetic signal is used as an example to show how sparsity can be observed during the iterative even/odd decomposition.

decomposition. Let us analyze Figure 6 that shows the separated representation performance of the decomposition tree for different kinds of data (excluding symmetric textures, where their perfect symmetry usually means that very few leaves are non-zero). Evidently the decomposition capacity to well approximate the data is not uniform, which is indicative that fewer leaves are necessary to obtain a good reconstructed signal for certain data types with respect to others. This indicates that there is a variable degree of leaves sparsity among data types, as more leaves sparsity leads ultimately to a more compact representation. Having fewer leaves carrying more energy is an effect of having a decomposition tree where just a minority of nodes (and subtrees descending from them) possesses the most significant part of the original signal energy.

The sparsity of the decomposition tree in this sense ultimately translates into looking for nodes which are decomposed with a high energy imbalance, i.e. they are either highly symmetric or antisymmetric. Measuring how much sparsity is present in decomposition trees by looking for energy imbalances, and associating to it a scalar measure, represents a characteristic feature extracted from the signal. This is the subject of the further experiments that is elaborated upon in the followings.

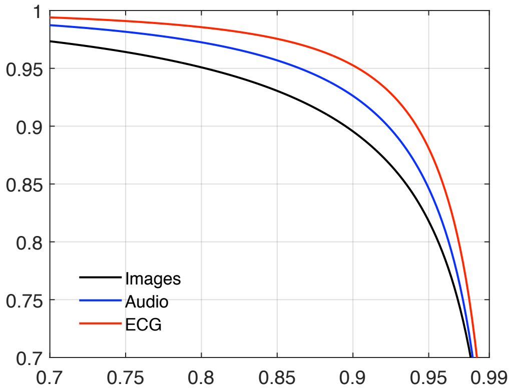
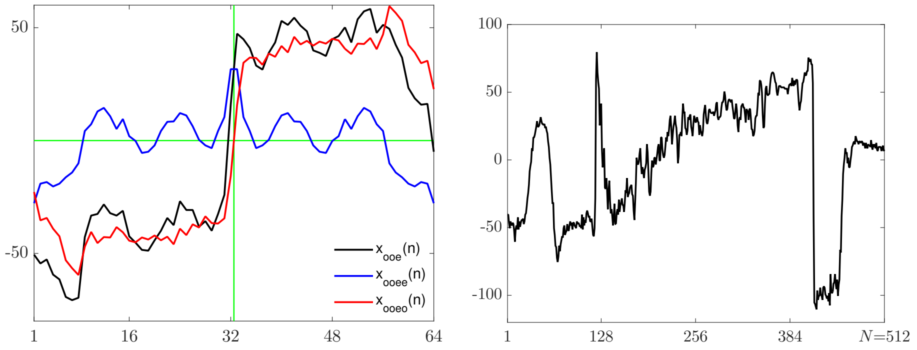


Figure 6: Approximation strength of the decomposition, disaggregating the results shown by the black curve in Figure 4. Information along the axes carries the same meaning as in Figure 4.

To recap the discussion so far, even if carrying the decomposition to the end yields the same coefficients (leaves) as the Walsh-Hadamard Transform, as previously shown, it is still possible to observe the presence of any even and odd symmetric node at any level of the decomposition. The identification can be readily performed by e.g. measuring, after the decomposition of a parent node, the ratio of the energy of the even part E_e w.r.t. the energy of the parent node E (or equivalently, that of the odd part E_o w.r.t. E, recall Eq. (5)): if it is close to 1 or 0, one of the two children nodes is preponderant, while if it is close to 0.5 no special (anti-)symmetry is present. It can be noted that such symmetric node, especially in deeper levels, is "buried" in the data, i.e. it is neither immediately perceptible by inspection nor it corresponds to a sort of local symmetry. It is only determined by a particular arrangement existing between signal values at specific locations relating to the particular level of the decomposition (i.e. which directly translate into what typically represents a pattern), thus determining the significance of the proposed measure. This is shown in Figure 7, which depicts the decomposition of a third level node into two children nodes (Figure 7a), where the odd part has the 91.9% of the energy of the parent node. Looking at the original sequence in Figure 7b, there is no local symmetry similar to the one in Figure 7a, however a particularly strong energy imbalance shows in the decomposition of this node.

To get an instant view of such "sparse nodes" occurrences in the data, we can draw a picture of the decomposition tree that highlights how the total energy distributes across the tree as it builds from the root to the leaves in a top-down fashion (see Figure 8). It is then possible to analyze how the energy of any node is divided among its two children nodes. The information carried by the sparsity of the nodes may in the end suggest different strategies to handle the decomposition based on the pursued objective. For example, the energy of the nodes can be used to interrupt the decomposition of nodes having energy ratio below a certain threshold B w.r.t. that of original sequence, effectively pruning subtrees which contains too little energy. On the other hand, significant energy imbalance, greater than another fixed percentage threshold S, may be signaled and determining the presence of a specific symmetry existing in the decomposition tree, thus expressing a good signature for the patterns present in the original sequence. In Figure 8 we have highlighted both occurrences.

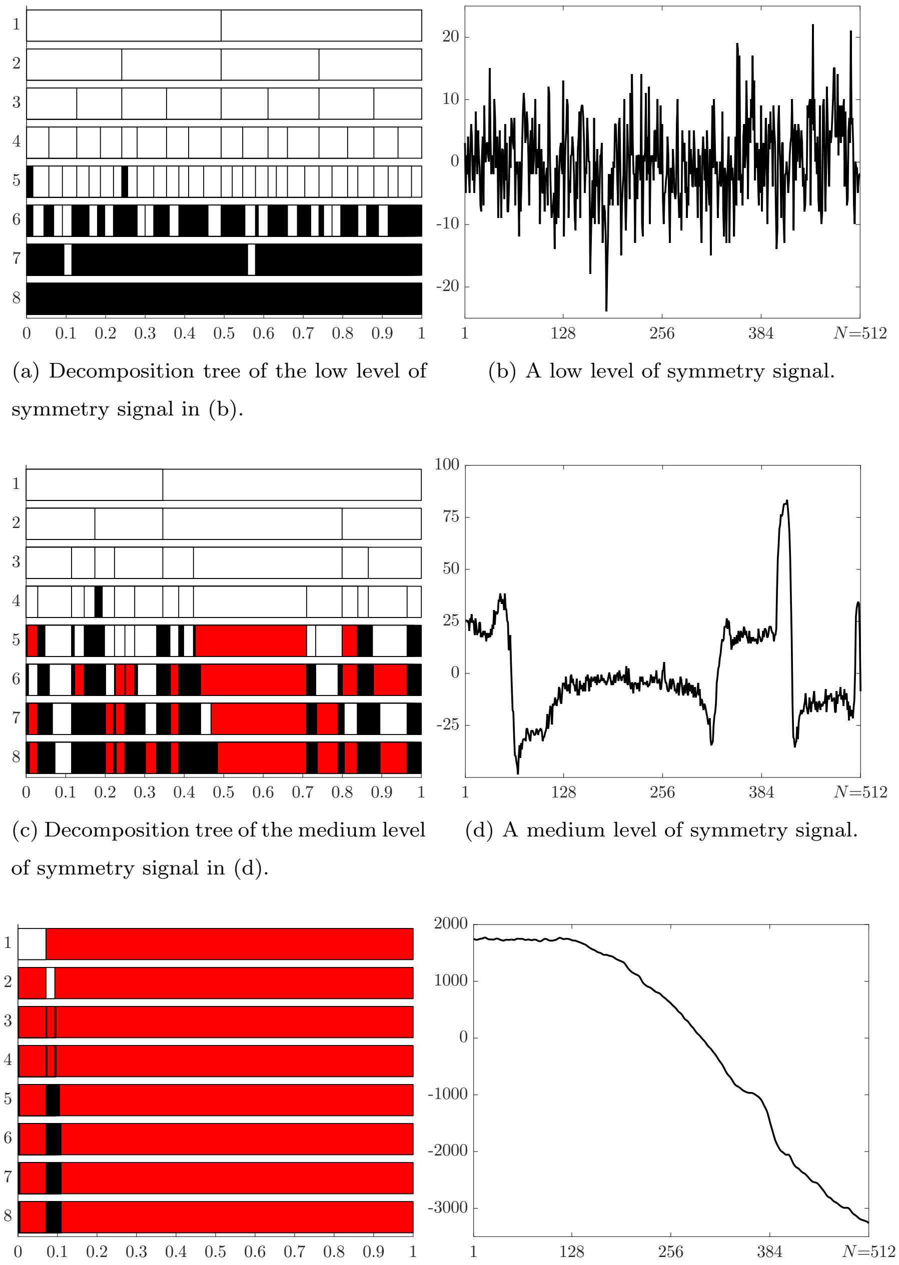
Recall that each node at a given level is orthogonal to every other node of the same level, so the sum of their energies is always equal to that of the original sequence, and the energies of children nodes always sum up to that of the parent node. In Figure 8 the decomposition levels are on the y-axis. For each row (level), there are a number of segments representing each node in the considered level, ordered from left to right as the rows of $\mathbf{D_1}$ (that is even part first, recall Section 4). The length of the segment represents the energy of the considered node w.r.t. the original signal: note how each segment is divided in two going from a level to the next. When the energy of a parent node goes mostly to one of the two children nodes (i.e. energy distribution greater than



(a) The decomposition of a third level node, corresponding to the even part of the odd part of the odd part of the original signal ($x_{ooe}[n]$, black). The odd signal ($x_{ooeo}[n]$, red) carries much more more energy than the even signal ($x_{ooee}[n]$, blue).

(b) The original sequence $x[n]$.

Figure 7: Part of a decomposition tree of an image row, which depicts how the presence of a highly symmetric node in lower levels cannot be readily perceived at the root level.



(a) Decomposition tree of the low level of symmetry signal in (b).

(b) A low level of symmetry signal.

(c) Decomposition tree of the medium level of symmetry signal in (d).

(d) A medium level of symmetry signal.

(e) Decomposition tree of the signal in (f).

(f) A signal exhibiting a high level of symmetry.

Figure 8: Three different examples of decomposition trees related to signals with respectively a high, medium and low measured symmetry.

S), such a child node is displayed in red. Instead when the energy of a node is smaller than B (as a percentage of the energy of the original signal), it is colored in black. The color white is used for all the other cases. In Figure 8, threshold values are $S = 0.9$ and $B = 0.02$.

Accordingly, let us build a measure on the amount of sparsity of the decomposition tree associated to a given signal. The energy imbalance d found during the decomposition of a certain node can be individually measured as:

$$d = \left| \frac{E_e - E_o}{2(E_e + E_o)} \right| = \max \left\{ \frac{E_e}{E_e + E_o}, \frac{E_o}{E_e + E_o} \right\} - \frac{1}{2} \tag{8}$$

The quantity d goes from 0 in the case of perfectly balanced even and odd parts (indicating absence of node sparsity) to 0.5 in the case of a purely even or odd node. For example, in the previous example shown in Figure 7, $d = 0.419$. We propose to sum this quantity for all the nodes, normalizing the measure of the nodes in each level by the number of nodes in that level, as in the following (valid for N equal to a power of 2 as is our case):

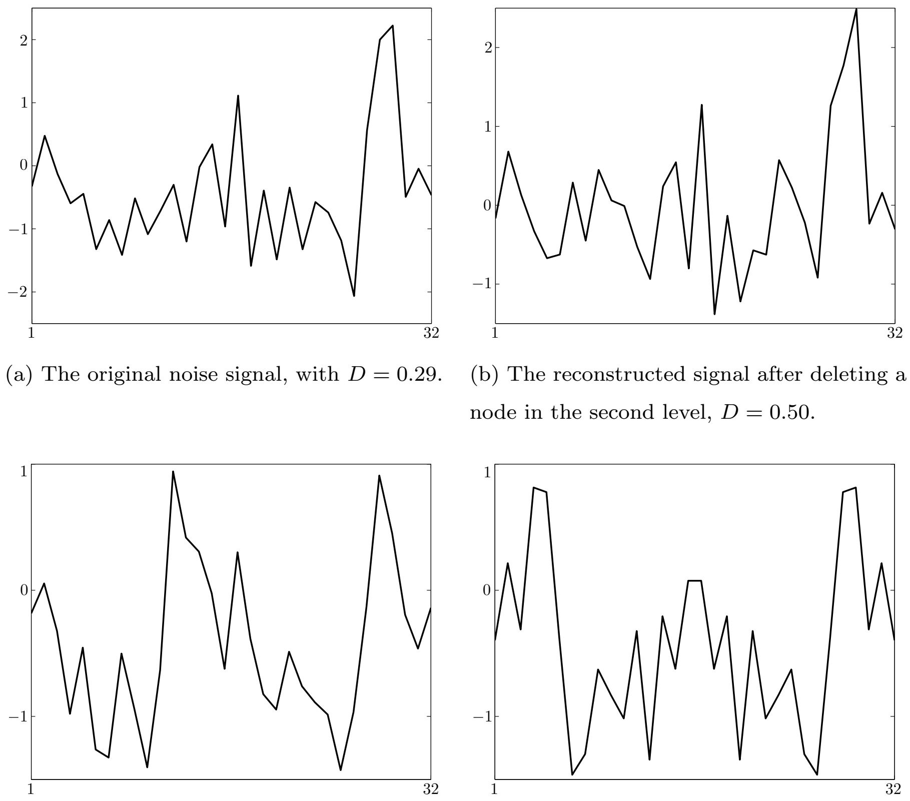
$$D = \frac{2}{\log_2 N} \sum_{l=1}^{\log_2 N} \frac{1}{2^{l-1}} \sum_{i=1}^{2^{l-1}} d_{li} \tag{9}$$

There are $\log_2 N$ levels, each with 2^{l-1} nodes. In the theoretical case in which each d_{li} takes the maximum value of 0.5 (each node is a pure even or odd signal), D would be equal to 1; if all d_{li} are equal to 0 (no node have any symmetry), then $D = 0$. The computation of such a measure is done similarly in the work by Kovesi [12] and successive publications in the same vein. However, in those works the objective is to measure the presence of a multi-scale local symmetry at a given point, where the scale is tied to the filter support. In our case, we measure the sparsity of an even/odd decomposition as computed at different scales, where this time the scale corresponds to a decomposition level, and each level analyzes the symmetry of different combinations of samples taken from the original sequence, thus losing the notion of locality on the process.

Further examples of the significance of D are shown in Fig 9. The original sequence shown in Fig 9a. is just a short excerpt of a Gaussian distributed white noise. Since there is no correlation between the samples, no particular symmetry is expected between any set of samples as considered for the construction of the nodes in the decomposition tree, in fact $D = 0.29$. It is possible to tinker with its decomposition tree, obviously changing the reconstructed signal in the process. The objective is to artificially increase D introducing a limited amount of changes in the decomposition tree to observe the relation (if any) between the signals found at the root of the various, slightly modified trees. The procedure consists in choosing a pair of child nodes and pouring all the energy of one into the other, effectively trimming the entire subtree of the deleted child node and putting the local d to 0.5. The effect on D is more pronounced if this procedure, that can be repeated at will, is done in lower levels. The results of this experiment are shown in Fig. 9b-9d.

In Figure 10 we have reported the distribution of the values of D across different data types in histogram form. Although discerning between them is a challenging classification problem (i.e. it is always possible to have an image row very much like an audio signal and viceversa), more so using just a scalar feature like D, one can observe how D varies between different data types. We have also fitted the histogram data using 4 different distributions, including the Gaussian one, and it can be observed that they fit very well the distribution of D values.

We have reported in Table 1 the mean and standard deviation of each distribution in Figure 10. Even if sometimes the signals cannot be clearly separated into different classes even by a human being, on the average the various data types possess a clearly different behavior in terms of D, suggesting that the even/odd symmetry measure translating into energy imbalances present in the nodes of their decomposition trees is indeed a discriminating feature. In particular, D distributes more towards higher values for data with on average smoother behavior and a tendency to self-replicate its time/space signal behavior, like ECG signals. In particular, this last property does not necessarily translate into the presence of local symmetries or otherwise periodic properties, instead the even/odd decomposition captures the presence of a possible sym-



(a) The original noise signal, with $D = 0.29$.

(b) The reconstructed signal after deleting a node in the second level, $D = 0.50$.

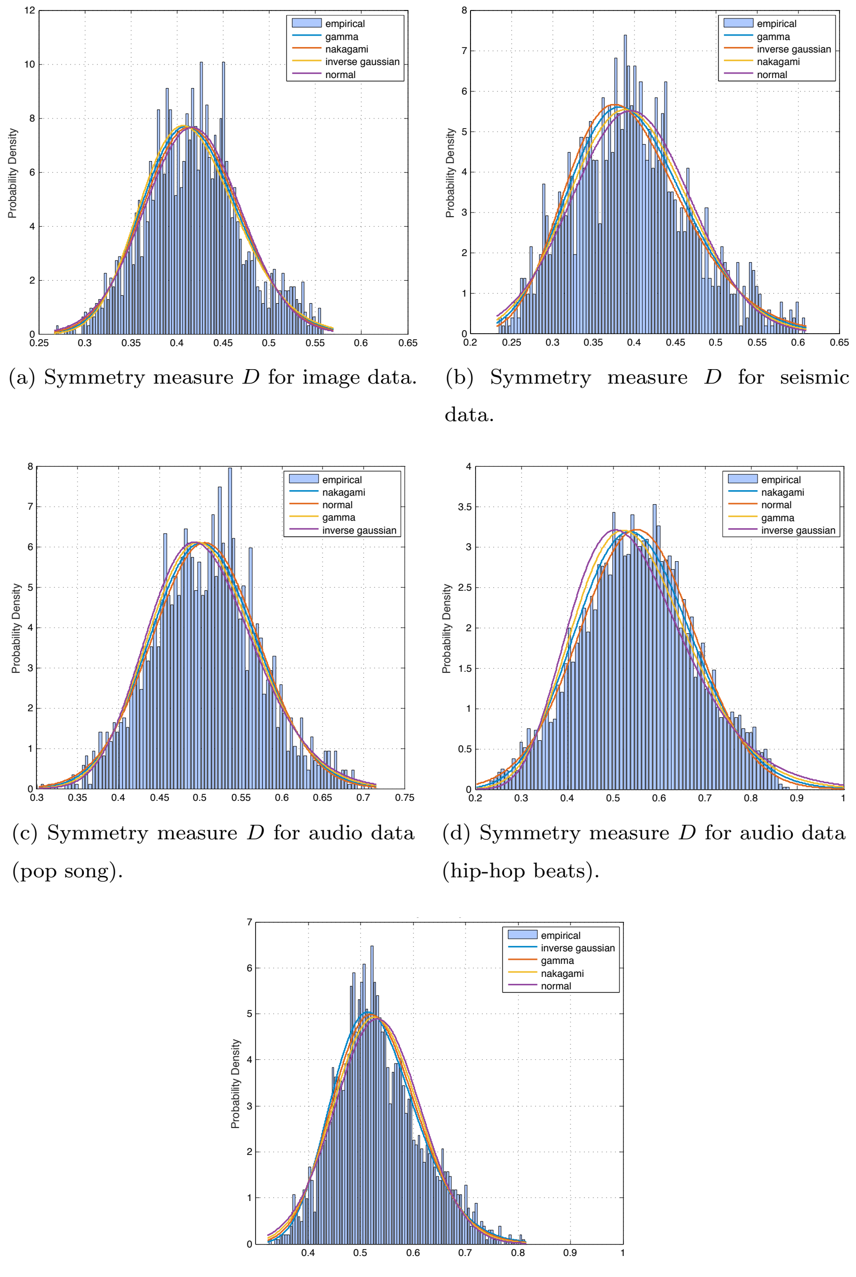
(c) The reconstructed signal after deleting a node in the second level and two nodes in the third level, $D = 0.65$.

(d) The reconstructed signal after deleting a node in the first level, $D = 0.78$ (of course, the signal is now globally symmetric).

Figure 9: Effect on D when deleting specific nodes in an example decomposition tree.

metric relation between combinations of different portions of the original signal at various scales.

In Figure 11 a pair of additional histograms are provided on two particular data types: vocal sound effects and texture patterns. Since such data may totally different waveforms in their class, ranging from quasi-periodic signals to erratic signals with low correlation akin to noise, a very different value distribution is observed, indicative of a strong inter-class variance in the decomposition



(a) Symmetry measure D for image data.

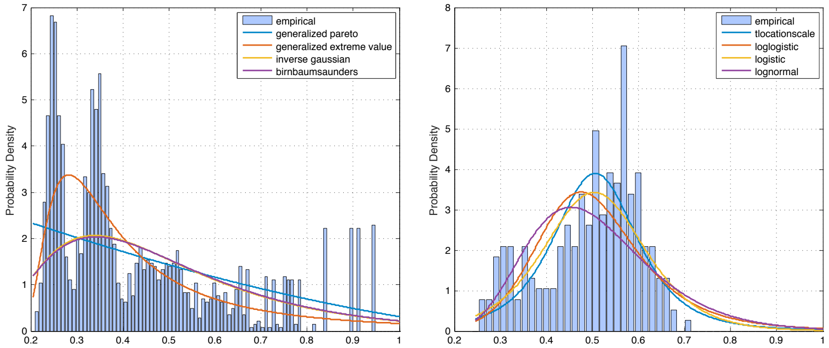
(b) Symmetry measure D for seismic data.

(c) Symmetry measure D for audio data (pop song).

(d) Symmetry measure D for audio data (hip-hop beats).

(e) Symmetry measure D for ECG data.

Figure 10: Histograms of the symmetry measure computed separately for each data type.



(a) Symmetry measure D for audio data.

(b) Symmetry measure D for audio data (vocal sound effects).

Figure 11: Additional histograms of the symmetry measure computed for more data types.

tree sparsity, coherently with the above argument. The mean and variance of the histograms for these distributions have been listed in Table 1 as well.

6. Details on the even/odd decomposition of discrete-time signals

Before concluding the paper, some extra details are provided in this section on how to perform the decomposition step of discrete sequences to allow its efficient and convenient iterative application.

To simplify the description, let us first modify some notations with respect to those we used in Section 4. Even if Eq. (4) is very similar to Eq. (1), to describe the iterative decomposition of sequences that follows it is simpler to change how the decomposition itself is performed, by shifting the original sequence and assuming that $x[n]$ be a strictly causal signal, meaning that $n = 1, \dots, L$ (excluding $n = 0$ from the support greatly simplifies the following discussion).

The even-odd decomposition for a strictly causal signal needs a slight modification in its definition, since the decomposition must be now performed around its midpoint, that does not coincide with the origin $n = 0$ but it is instead located at $n = \frac{L+1}{2}$. To flip $x[n]$ around the midpoint, the sequence $x[L+1-n]$

Data type	Mean	Std Dev
Image rows	0.4167	0.0521
Seismic data	0.3957	0.0052
Hip-hop beats	0.5513	0.0154
Pop song	0.5061	0.0654
ECG signals	0.5319	0.0816
Vocal sound effects	0.5084	0.0205
Texture patterns	0.5018	0.0687

Table 1: Mean and standard deviation computed on the histograms in Figure 10 and Figure 11.

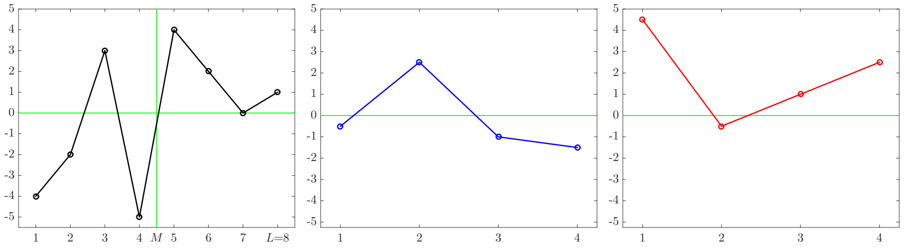
must be taken for $n = 1, \ldots, L$ and Eq. (4) becomes:

$$x_e[n] = \frac{x[n] + x[L+1-n]}{2}; \quad x_o[n] = \frac{x[n] - x[L+1-n]}{2}; \quad x[n] = x_e[n] + x_o[n] \tag{10}$$

If L is even, the midpoint $n = \frac{L+1}{2}$ corresponds to a half-integer point. Mirroring the sequence around this point is straightforward because the "causal" part goes from $n = \frac{L}{2} + 1$ to $n = L$ while the "anti-causal" part goes (backwards) from $n = \frac{L}{2}$ to $n = 1$. Both even and odd causal parts, $x_e^{(c)}[n]$ and $x_o^{(c)}[n]$, have therefore $\frac{L}{2}$ samples and furthermore possess energy $E_e/2$ and $E_o/2$ respectively, for the same orthogonality reasons given in Section 3. Figure 12 illustrates a simple example of a single decomposition step with $L = 8$.

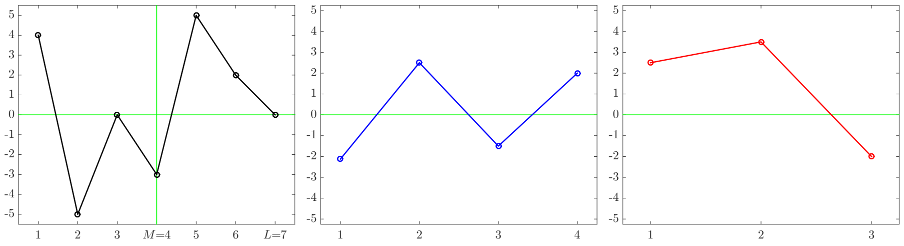
However, in the case of L odd, the situation is different. The midpoint $n = \frac{L+1}{2}$ around which to perform the decomposition is now an integer point. This is analogous to the time-centered case outlined at the start of Section 4, where the midpoint was $n = 0$. Applying Eq. (10), it turns out that $x_e[\frac{L+1}{2}] = x[\frac{L+1}{2}]$ and $x_o[\frac{L+1}{2}] = 0$. So again, the decomposition is unbalanced because the informative (i.e. causal) part of the odd sequence, $x_o^{(c)}[n]$, is just $\frac{L-1}{2}$ samples because of the discarded one, which is the one corresponding to $n = \frac{L+1}{2}$, is always 0. On the other hand, the causal part of the even sequence $x_e^{(c)}[n]$ has a sample more, for a total of $\frac{L+1}{2}$, and the first one corresponds to the midpoint sample of the original sequence as stated above. Figure 13 depicts the latter situation using the case of $L = 7$.

There is an important detail to note in such a decomposition with L odd. To keep Eq. (5) valid, that is to guarantee that the causal parts of the even and odd sequences carry exactly $E_e/2$ and $E_o/2$ energy respectively, the first sample of the causal part of the even sequence, which is equal to $x[\frac{L+1}{2}]$, must be normalized by $\sqrt{2}$ (so that it carries half of its original energy, compare the first



(a) Original sequence $x[n]$. (b) Causal even signal $x_e^{(c)}[n]$. (c) Causal odd signal $x_o^{(c)}[n]$.

Figure 12: When L is even, the midpoint (M in the left plot) is a half-integer point and the two sequences have the same length.



(a) Original sequence $x[n]$. (b) Causal even signal $x_e^{(c)}[n]$. (c) Causal odd signal $x_o^{(c)}[n]$.

Figure 13: When L is odd, the midpoint (M in the left plot) is an integer point and the causal part of the even sequence starts with the normalized original central sample. In this case it is longer by one sample with respect to the odd sequence.

sample of Figure 13b with the middle one of Figure 13a). It can be seen that during the decomposition process it is counted twice when the even sequence is mirrored around it, so the value is unchanged when divided by 2 and the whole energy of this sample stays on the even part. However, if we desire that the energy of the causal part carries half of E_e, the energy of the even component, as is the case for L odd, the normalization correctly handles this situation. This normalization does not change the substance under the decomposition, but must be kept in mind during the energy computations and the reconstruction process.

Given this difference in the decomposition process in the case of L even or odd, it follows that the reconstruction process is also different. For L even it is similar to what happens in the case of continuous time signals: the even and odd parts are mirrored (to the left since we retained the causal parts) and then summed. Instead for L odd the central sample of the reconstructed sequence is $x_e^{(c)}(1)$, multiplied by $\sqrt{2}$ to get rid of the normalization, and the other samples of $x_e^{(c)}[n]$ are mirrored to the left just like those of $x_o^{(c)}[n]$ and then added to obtain the original samples except the central one.

As an example of why the normalization keeps the energies of the children nodes balances, consider the sequence $x[n] = [2\ 2\ 1]$ with $L = 3$. The energy E is 9. To obtain the even child node, $x[n]$ and its flipped version are added, the result divided by 2 and then the two rightmost samples (thus including the middle one) are taken as the causal part. For the odd child node, the flipped version must be subtracted instead and only the rightmost sample is kept (the middle one is obviously 0). So, without normalization in this case $x_e[n] = [1.5\ 2\ 1.5]$ and $x_e^{(c)}[n] = [2\ 1.5]$, whereas $x_o[n] = [0.5\ 0\ -0.5]$ and $x_o^{(c)}[n] = [-0.5]$. Note how the causal children nodes can correctly reconstruct $x[n]$ using the process described above.

Now, let us analyze the energies. The energy E_o of $x_o[n]$ is 0.5 and the energy of $x_o^{(c)}[n]$ is 0.25 which is $E_o/2$, as expected. However, the energy E_e of $x_e[n]$ is 8.5 but the energy of $x_e^{(c)}[n]$ is 6.25 which is not $E_e/2$. That happened because the central sample has been counted twice during the decomposition as explained above. To let the energy of $x_e^{(c)}[n]$ be $E_e/2$, that sample must be normalized by $\sqrt{2}$, so that the sequence is now $x_e^{(c)}[n] = [\sqrt{2}\ 1.5]$ with energy $4.25 = E_e/2$. Using the normalization allows a simpler, unified implementation of both the energy imbalance analysis and the reconstruction process.

7. Conclusions

In this paper we proposed the iteration of the even/odd decomposition of a 1-D energy signal to construct a decomposition tree, for either the continuous-time and discrete-time settings, and studied how such tree could be exploited for a variety of signal processing tasks. We limited our scope to limited support signals to make the recentering operation possible, enabling the iteration of the basic decomposition to take place. In particular, we pointed out that for continuous-time, finite support signals such tree is infinitely deep with ever-shrinking nodes support, while for discrete-time, finite support signals the decomposition tree is always finite and that a little care is required in the latter case to simultaneously not increase the representation support and guarantee correct energy balancing between children nodes.

In addition, we proved that for discrete sequences, for representation compactness sake, when the decomposition is complete the performance of the proposed method is equivalent to that of the Walsh-Hadamard Transform. However, we did show how such a decomposition tree can also be effectively exploited to detect hidden sparsity in the resulting decomposition tree from the data and how their presence is diversified between different signal kinds. In particular the identification of local symmetries in the decomposition directly represents a particular arrangement existing between signal values, thus determines an effective signature to describe signal patterns. Coherently a measure of such sparsity can be effectively detected during the decomposition process, prompting further processing of the signal.

Many extensions to this work are possible. In our present research, we are addressing the limitations of considering only the even/odd decomposition of the signal around its midpoint. In fact, by its nature, this transform is able to detect and effectively represent symmetries in the data but the computation is carried just around the time origin or the midpoint. To overcome this limitation, the decomposition step at the heart of the transform needs to be generalized.

On the other hand, by combining the latter generalization with the iterative process described here, it is possible to adapt the framework presented here to the search for local symmetries, by reducing the scope of the decomposition and interpreting the energy imbalance as a direct hint of symmetry presence. Some preliminary results in this vein have already been presented in [6].

Finally, we based our analysis on the Euclidean norm, i.e. the energy of the nodes. It is also possible to compute the symmetry-based distances using the L_1 norm of the children nodes, instead of the L_2 norm (energy) as we have done in the experiments shown here, to speed up the computing of the symmetry measure D. In this case, the price to pay is the loss of the orthogonality between the tree nodes, so more complex normalization factors than those of Eq. (9) are needed. This topic is also matter of undergoing research.

Acknowledgements - This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] A. Apostolico and Z. Galil. *Pattern matching algorithms.* Oxford University Press on Demand, 1997.

[2] J. Bendat and A. Piersol. *Engineering applications of correlation and spectral analysis.* Wiley-Interscience, New York, 1980.

[3] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[5] B. Fino and V. Algazi. Unified matrix treatment of the fast walsh-hadamard transform. *IEEE Trans. on Computers*, 11(100):1142–1146, 1976.

[6] A. Gnutti, F. Guerrini, and R. Leonardi. Representation of signals by local symmetry decomposition. In *Proc. of the 23rd European Signal Processing Conference (EUSIPCO '15)*, pages 983–987, 2015.

[7] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):215–220, 2000.

[8] N. Hurley and S. Rickard. Comparing measures of sparsity. *IEEE Trans. Inf. Theor.*, 55(10):4723–4741, 2009.

[9] Incorporated research institutions for seismology data. `https://www.iris.edu/hq/resource/bb_processing_matlab`. Accessed: 2017-04-27.

[10] A. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. Image Processing*, 1(1):18–30, 1992.

[11] G. John and M. e. a. Charles. *Algorithms for Statistical Signal Processing.* Prentice Hall, London, 2002.

[12] P. Kovesi et al. Symmetry and asymmetry from local phase. In *Tenth Australian joint conference on artificial intelligence*, volume 190, pages 2–4. Citeseer, 1997.

[13] J. Liu, G. Slota, G. Zheng, Z. Wu, M. Park, S. Lee, I. Rauschert, and Y. Liu. Symmetry detection from realworld images competition 2013: Summary and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 200–205, 2013.

[14] S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):674–693, 1989.

[15] A. Oppenheim, A. Schafer, and C. Jones. *Discrete-time signal processing.* Prentice Hall, 1999.

[16] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. Smeaton, and G. Quenot. Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proc. of TRECVID 2014*, page 52.

NIST, USA, 2014.

[17] V. Sanchez, R. Abugharbieh, and P. Nasiopoulos. Symmetry-based scalable lossless compression of 3d medical image data. *IEEE Transactions on Medical Imaging*, 28(7):1062–1072, 2009.

[18] C. Sun and J. Sherrah. 3-d symmetry detection using the extended gaussian image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:164–168, 1997.

We the authors thank the reviewers for their valuable observations that have helped to pinpoint the essence of the contribution and improve the clarity of the presentation.

Note: we refer with Lnnn to line numbers in the original submission.

***

Response to Reviewer #1 comment No. 1:

For continuous-time signals, there is no theoretical limit to the number of levels (L193). The support of a child node is always half that of its parent, so as the level number increases the support of the nodes at higher levels shrinks but never goes to 0. The experiments, on the other hand, has been conducted on discrete, finite sequences. In this case, the number of levels is always finite (L237). In particular, the number of levels is equal to ceil(log2(L)) for a general sequence length L. To improve clarity on this matter, we added such facts respectively in Section 3 and at the end of Section 4, where we discuss generic sequence lengths.

Although we hinted that it is in principle possible to choose to arrest the decomposition at any given level, in our experiments we always let the decomposition go on until the one-sample leaves are reached, that is after log2(512)=9 levels. So, in essence what drives the number of levels even across different data types is the length of the considered root sequence. We chose L=512 to be consistent with the length of the image rows.

The number of obtained levels may be actually lower because the threshold B would prevent to further decompose a given subtree, as expounded as well in the added paragraph at the end of Section 4. This is why we added ``(at most)'' in L247. We revised the sentence to be a bit more specific.

***

Response to Reviewer #1 comment No. 2:

Eq. (8) measures the energy imbalance in a single decomposition step, while the sparsity measure of the whole decomposition tree is in Eq. (9). The reason why this energy imbalance leads to sparsity measurement in tree representations is expounded upon in answer No. 1 to Reviewer #2.

The measure proposed by the reviewer is actually the same as that in Eq. (8), except for a factor 2. In fact, if we assume in Eq. (8) that E_e>E_o, and recalling that E=E_e+E_o, then d=E_e/E-1/2=(2E_e-E)/2E=(E_e-E_o)/2E (the case E_e<E_o is analogous). We used our version of Eq. (8) because we thought it highlighted that only either one of the even or odd symmetries can lead to energy imbalance, although the suggested one (that we added in the equation, divided by 2 for consistency) is interesting in its own right because it is similar to the reflection coefficient definition. It is no surprise, in the end, that the problem of determining if a sequence is symmetric is connected to determine the amount of reflection experienced by waves in physics.

***

Response to Reviewer #1 comment No. 3:

As suggested, we performed data fitting of the histograms in Fig. 9 (Figs. 10 and 11 in the revised version) on both Gaussian and a number of other distributions, all of which estimate well the obtained histograms. The reason why different data types are associated to different D distributions is ultimately the same as the reason why a particular decomposition tree is sparser than another, in the sense that (on the average) the realizations taken from those different signals more or less tend to fit in the sparse decomposition tree model embodied by Eq. (9). Please refer also to answers No. 1 and No. 5 to Reviewer #2.

***

Response to Reviewer #2 comment No. 1:

Sparsity in the iterative even/odd decomposition tree is measured in terms of energy imbalance during any single decomposition step. When the imbalance is high, the parent signal is strongly (anti-)symmetric and therefore one between the even and odd child nodes has much lower energy with respect to the other (but possibly not negligible with respect to the original signal energy). If the lower energy child node is discarded, it is equivalent to put the correspondent subtree to 0, leading naturally to a sparser representation of the decomposition tree in terms of represented subtrees. As Fig. 5 proves, this is not a matter of compactness in the last-level leaves representation (which is the same obtained using a WHT as reported in Fig. 4). Instead, taking advantage of the iterative decomposition, sparsity can manifest itself in any node of any level. The use of the L2 norm to detect sparsity in Eq. (8) for a single decomposition step, and by extension of the whole tree through Eq. (9), can only approximate the true, L0 norm sparsity as that displayed by Fig. 5d (which is obtained starting from a synthetic sequence), but has the great advantage of preserving orthogonality during the decomposition, thus allowing to precisely control the amount of distortion introduced in the reconstructed sequence by discarding any node.

Hence, the next step is to try to identify which characteristics a sequence should possess to be associated with a sparse decomposition tree, so that we may infer which classes of signals among those considered have sparser trees than the others. These characteristics should be derived in the time domain, since the decomposition works in that domain and it has a non-linear nature. To prove what these characteristics are, we included a set of new experiments on N=32 long sequences. In these experiments (Fig. 9 in the revised version), sequences that are associated to sparse (with sparsity appearing in different levels) and non-sparse decomposition tree are reported. These new results substantiate the conclusion that we offered in L428. Obviously, the conclusions learnt from these experiments may be carried on to longer sequences, recalling the iterative nature of the tree.

***

Response to Reviewer #2 comment No. 2:

As reported from L263 onwards, the recursive even/odd decomposition, when considered for L=2^m long sequences and for complete decomposition trees, represents an alternative implementation of the fast Walsh-Hadamard Transform (WHT). Of course, the WHT is related to the Haar wavelet transform (that is the same as the Daubechies wavelet transform 'db1'), in that every step of the WHT may be described as the application of the Haar wavelet on the intermediate result, keeping in mind that WHT has no shifting feature and thus a windowed input sequence has to be considered. The inverse process, namely obtaining a wavelet decomposition through the application of a WHT-like process, would have to do with modifying the decomposition step to actually output approximation and detail signals, and that would mean to insert suitable filters in the columns of D2 as well as a decimation step; however, all the nice properties in terms of conservation of support length would be lost.

***

Response to Reviewer #2 comment No. 3:

We added Subsection 1.1 that specifies the contributions of the paper right from the start.

***

Response to Reviewer #2 comment No. 4:

We consider the proposed transform as a novel contribution, although there are connections with existing processing tools, namely the basic even/odd decomposition and the WHT. Such preliminary background is well covered in classic signal processing textbooks, e.g. [15], or seminal papers, e.g. [14] (refs number in the revised version). We added a few more broad spectrum references in the introduction for those fields touched upon in the experimental section.

This paper is building on a classic concept like that of orthogonal block transforms and then making it "new again" through its original take on the the study of sparsity through the iterative generation of intermediate results rather than relying on the L0 norm of the coefficients in the transformed domain. We clarify our take in the new Section 1.1 which lists the paper contributions.

***

Response to Reviewer #2 comment No. 5:

We added more data types to our experimental section to strengthen the argument in Section 5. However, it is not our purpose to propose D as a categorization feature: indeed, a scalar feature may incur in many mis-classification results as its dimensionality is too small. Instead, the objective is to prove that the sparsity in decomposition tree is a discriminating feature in terms of the temporal evolution of the sequence, as the new experiments described in the answer No. 1 also show. In this case, the histograms representing the feature distribution show how on average different data types may be used to classify the whole data subset.

***

***Highlights (for review)**

- Hierarchical even-odd decomposition of signals able to detect the presence of hidden patterns in the data.
- Measuring an overall energy imbalance distribution in a recursive decomposition of a signal.
- Stopping a recursive transformation of data to favour sparsity of the tree representation.
- Discrete versus continuous-time operational implementations of the decomposition.