Redundant Wavelets on Graphs and High Dimensional Data Clouds

Idan Ram, Michael Elad, *Senior Member, IEEE*, and Israel Cohen, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a new redundant wavelet transform applicable to scalar functions defined on high dimensional coordinates, weighted graphs and networks. The proposed transform utilizes the distances between the given data points to construct tree-like structures. We modify the filter-bank decomposition scheme of the redundant wavelet transform by adding in each decomposition level operators that reorder the approximation coefficients. These reordering operators are derived by organizing the tree-node features so as to shorten the path that passes through these points. We explore the use of the proposed transform for the recovery of labels defined on point clouds and to image denoising, and show that in both cases the results are promising.**

***Index Terms*—High-dimensional signal processing, image denoising, label recovery, redundancy, tree, wavelet.**

I. Introduction

SIGNAL processing problems may involve inference of an unknown scalar target function defined on nonuniformly sampled high-dimensional grid, a graph or a network. A major challenge in processing functions on topologically complicated coordinates, is to find efficient methods to represent and learn them. Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be the data set such that $\mathbf{x}_i \in \mathbb{R}^n$ are points in high-dimension, or feature points associated with the nodes of a weighted graph or network. Also, let $f : \mathbf{X} \to \mathbb{R}$ be a scalar function defined on the above coordinates, and let $\mathbf{f} = [f_1, \ldots, f_N]^T$, where $f_i = f(\mathbf{x}_i)$. A key assumption in this work is that under a distance measure $w(\cdot, \cdot)$ in $\mathbb{R}^n$, proximity between the two coordinates $\mathbf{x}_i$ and $\mathbf{x}_j$ implies proximity between their corresponding values f_i and f_j. The goal in this work is to develop a redundant wavelet transform that can efficiently represent the high-dimensional function $\mathbf{f}$. Efficiency here implies sparsity, i.e., representing $\mathbf{f}$ accurately with as few as possible wavelet coefficients.

In our previous work [1], we have introduced the generalized tree-based wavelet transform (GTBWT), which is a critically sampled (in fact, unitary) wavelet transform applicable to functions defined on irregularly sampled grid of coordinates. We have shown that this transform requires less coefficients than both the 1-D and 2-D separable wavelet transforms to represent an image, and is useful for image denoising. The main limitation of the GTBWT is sensitivity to translation. Indeed, in order to obtain a smooth denoising result in [1], we utilized a redundant representation obtained by applying several random variants of the GTBWT to the noisy image. This approach is effectively similar to applying a redundant transform to the image in a rather cumbersome and computationally intensive manner.

In this paper, we introduce a redundant tree-based wavelet transform (RTBWT), which extends the redundant wavelet transform [2]–[4] to scalar functions defined on high-dimensional data clouds, graphs and networks. This transform is obtained by modifying an implementation of the redundant wavelet transform, which was proposed by Shensa [3] and Beylkin [4], similarly to the way we modified the decomposition scheme of the orthonormal transform in [1]. This implementation employs a filter-bank decomposition scheme, similarly to the orthonormal discrete wavelet transform. However, in each level of this scheme none of the coefficients are discarded. We add in each decomposition level operators that reorder the approximation coefficients. These operators are data-dependent, and are obtained using tree-like structures constructed from the data points. Each reordering operator is derived by organizing the tree-node features in the corresponding level of the tree so as to shorten the path that passes through these points. The reordering operators increase the regularity of the permuted approximation coefficients signals, which cause their representation with the proposed wavelet transform to be more efficient (sparse). We note that our proposed transform shares some similarities with the easy path wavelet transform proposed in [5], which also employs permutations in order to enhance its sparsity. However, while we develop a redundant transform for functions on point clouds or high-dimensional graph-data, [5] concentrates on an orthonormal transform for images.

We explore the use of the proposed transform for the recovery of labels defined on point clouds, and show that it outperforms the 1-nearest neighbor (1-nn) classifier. We also explore the use of the transform to image denoising, and show that it outperforms the K-SVD based algorithm of Elad and Aharon [6], and achieves denoising results that are similar to those obtained with the BM3D algorithm [7]. We also show that the RTBWT and GTBWT achieve similar denoising results, while the former is computationally less-demanding.

The paper is organized as follows: In Section II, we introduce the proposed redundant tree-based wavelet transform. In Section III, we explore the use of this transform to the recovery of labels defined on point clouds, and in Section IV we explore the use of the transform to image denoising. We present experimental results that demonstrate the advantages of the proposed

Manuscript received November 19, 2011; revised February 12, 2012; accepted March 06, 2012. Date of current version April 03, 2012. This work was supported by the Israel Science Foundation (Grants 1130/11 and 1031/08) and in part by the Rubin Scientific and Medical research fund. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dimitrios Tzovaras.

I. Ram and I. Cohen are with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: idanram@tx.technion.ac.il; icohen@ee.technion.ac.il).

M. Elad is with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: elad@cs.technion.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2012.2190983

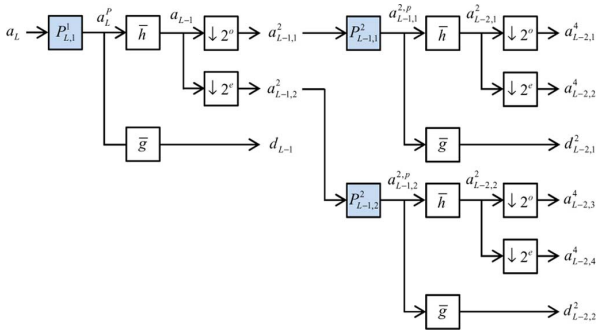


Fig. 1. Proposed redundant wavelet decomposition scheme.

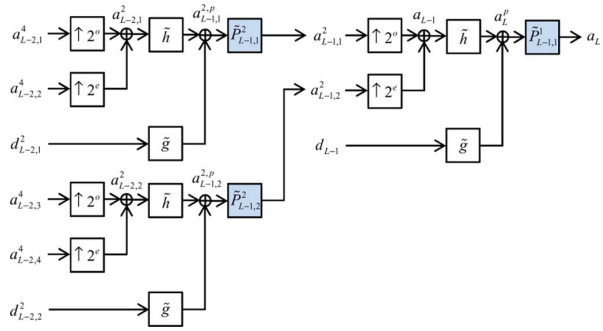


Fig. 2. Proposed redundant wavelet reconstruction scheme.

transform in both these sections. We summarize the paper in Section V.

II. Redundant Tree-Based Wavelet Transform

A. Decomposition and Reconstruction Schemes

We wish to develop a redundant wavelet transform that efficiently (sparsely) represents its input signal $\mathbf{f}$, defined on an irregularly sampled grid of coordinates. To this end, we extend the redundant wavelet transform, similarly to the way we extended the orthonormal transform in [1]. We note that we construct our proposed transform by modifying an implementation of the redundant wavelet transform as proposed by Shensa [3] and Beylkin [4], and not the well known algorithme à trous [2]. This implementation employs a filter-bank decomposition scheme, similarly to the orthonormal discrete wavelet transform. However, in each level of this scheme all the coefficients are retained since the highpass bands do not contain decimators, and the decimation in the lowpass bands is replaced by a split into even and odd sequences, which are further decomposed in the next decomposition level.

Fig. 1 describes the decomposition scheme of our proposed redundant wavelet transform. We denote the coarsest decomposition level $\ell = 1$ and the finest level $\ell = L = \log_2 N + 1$. $\mathbf{a}_\ell$ and $\mathbf{d}_\ell$ denote the approximation and detail coefficients in level ℓ, respectively. We start with the finest decomposition level, and apply to $\mathbf{a}_L = \mathbf{f}$ the operator $P^1_{L,1}$, which produces a permuted version $\mathbf{a}^p_L$ of its input vector. $P^t_{\ell,s}$ denotes an operator that operates in the sth band, out of t bands, in the ℓth decomposition level. It produces a permuted version of its input vector, and operates as a linear and unitary operator given that vector. The operators $P^t_{\ell,s}$ make the difference between our proposed wavelet decomposition scheme and the common redundant wavelet transform [3], [4]. As we explain later, these operators "smooth" the approximation coefficients in the different levels of the decomposition scheme. Next, we apply the wavelet decomposition filters $\bar{\mathbf{h}}$ and $\bar{\mathbf{g}}$ on $\mathbf{a}^p_L$, and obtain the vectors $\mathbf{a}_{L-1}$ and $\mathbf{d}_{L-1}$, respectively. Let $\downarrow 2^o$ and $\downarrow 2^e$ denote $2:1$ decimators that keep the odd and even samples of their input, respectively. Then we employ these decimators to obtain the signals $\mathbf{a}^2_{L-1,1}$ and $\mathbf{a}^2_{L-1,2}$. These two vectors are used as inputs for the next decomposition level.

We continue in a similar manner in the following decomposition levels. Let $\mathbf{a}^t_{\ell,s}$ denote an approximation coefficients vector, which is found in the sth band (out of t bands), in the ℓth decomposition level. This vector is obtained by starting from the sth sample in $\mathbf{a}_\ell$, and keeping every tth sample. Then in the ℓth decomposition level we decompose each of the vectors $\left\{\mathbf{a}^{2^{L-\ell}}_{\ell,s}\right\}_{s=1}^{2^{L-\ell}}$. We first apply on each vector $\mathbf{a}^{2^{L-\ell}}_{\ell,s}$ the operator $P^{2^{L-\ell}}_{\ell,s}$ and obtain a permuted version $\mathbf{a}^{2^{L-\ell},p}_{\ell,s}$. We then filter $\mathbf{a}^{2^{L-\ell},p}_{\ell,s}$ with $\bar{\mathbf{h}}$ and $\bar{\mathbf{g}}$ and obtain the vectors $\mathbf{a}^{2^{L-\ell}}_{\ell-1,s}$ and $\mathbf{d}^{2^{L-\ell}}_{\ell-1,s}$, respectively. Finally, we employ the decimators $\downarrow 2^o$ and $\downarrow 2^e$ to split each of the vectors $\left\{\mathbf{a}^{2^{L-\ell}}_{\ell-1,s}\right\}_{s=1}^{2^{L-\ell}}$ into even and odd sequences, respectively, and obtain the set of vectors $\left\{\mathbf{a}^{2^{L-\ell+1}}_{\ell-1,s}\right\}_{s=1}^{2^{L-\ell+1}}$.

In a similar manner, Fig. 2 describes the reconstruction scheme of our redundant wavelet transform. $\tilde{\mathbf{h}} = (1/2)\mathbf{h}$ and $\tilde{\mathbf{g}} = (1/2)\mathbf{g}$, where $\mathbf{h}$ and $\mathbf{g}$ denote the wavelet reconstruction filters, and the interpolators denoted by $\uparrow 2^o$ and $\uparrow 2^e$ place the samples of their input vector in the odd and even locations of their output vector, respectively. Finally, the operator $\tilde{P}^t_{\ell,s}$ reorders a vector so as to cancel the ordering done by $P^t_{\ell,s}$, i.e., $\tilde{P}^t_{\ell,s} = \left(P^t_{\ell,s}\right)^{-1}$. We next describe how the operators $P^t_{\ell,s}$ are determined in each level of the transform.

B. Building the Operators $P^t_{\ell,s}$

We wish to design the operators $P^t_{\ell,s}$ in a manner which results in an efficient (sparse) representation of the input signal by the proposed transform. The wavelet transform is known to produce a small number of large coefficients when it is applied to piecewise regular signals [2]. Thus, we would like the operator $P^t_{\ell,s}$, applied to $\mathbf{a}^t_{\ell,s}$, to produce a signal which is as regular as possible. We start with the finest level, and try to find the permutation that the operator $P^1_{L,1}$ applies to $\mathbf{a}_L = \mathbf{f}$. When the signal $\mathbf{f}$ is known, the optimal solution would be to apply a simple *sort* operation. However, since we are interested in the case where $\mathbf{f}$ is not necessarily known (such as in the case where $\mathbf{f}$ is noisy, or has missing values), we would try to find a suboptimal ordering operation, using the feature coordinates $\mathbf{x}_i$.

We recall our assumption that the distance $w(\mathbf{x}_i, \mathbf{x}_j)$ predicts the proximity between the samples f_i and f_j. Thus, we try to reorder the points $\mathbf{x}_i$ so that they form a smooth path, hoping that the corresponding reordered 1D signal $\mathbf{a}^p_L$ will also be smooth. The "smoothness" of the reordered signal $\mathbf{a}^p_L$ can be measured using its total variation measure

$$\|\mathbf{a}^p_L\|_{TV} = \sum_{j=2}^{N} \left|a^p_L(j) - a^p_L(j-1)\right|. \tag{1}$$

Let $\left\{\mathbf{x}^p_j\right\}_{j=1}^N$ denote the points $\{\mathbf{x}_i\}_{i=1}^N$ in their new order. Then by analogy, we measure the "smoothness" of the path through the points $\mathbf{x}^p_j$ by the measure

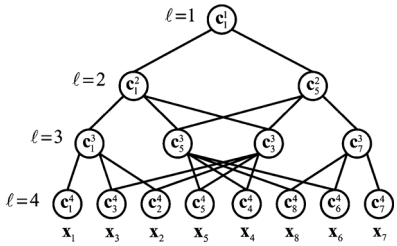


Fig. 3. Illustration of a "generalized" tree.

$$C_{TV}^{p} = \sum_{j=2}^{N} w\left(\mathbf{x}_{j}^{p}, \mathbf{x}_{j-1}^{p}\right). \tag{2}$$

Minimizing C_{TV}^{p} comes down to finding the shortest path that passes through the set of points $\mathbf{x}_j$, visiting each point only once. This can be regarded as an instance of the traveling salesman problem [8], which can become very computationally exhaustive for large sets of points. We choose a simple approximate solution, which is to start from an arbitrary point, and continue from each point to its nearest neighbor, not visiting any point twice. The permutation applied by the operator $P_{L,1}^{1}$ is defined as the order in the found path.

In order to employ the aforementioned method to find the operators $P_{L-1,1}^{2}$ and $P_{L-1,2}^{2}$ in the $L-1$th decomposition level, we again require feature points in order to predict the proximity between the samples of $\mathbf{a}_{L-1}$. Since $\mathbf{a}_{L-1,1}^{2}$ and $\mathbf{a}_{L-1,2}^{2}$ are obtained from $\mathbf{a}_L$ through filtering and subsampling, each approximation coefficient $a_{L-1}(i)$ is in fact calculated as a weighted mean of coefficients from $\mathbf{a}_L$, where the coefficients in $\bar{\mathbf{h}}$ serve as the weights. Thus, we calculate the feature point $\mathbf{c}_i^{L-1}$, which corresponds to $a_{L-1}(i)$, by replacing each coefficient $a_L(i)$ in this weighted mean by its corresponding feature point $\mathbf{x}_i$. We then employ the approximate shortest path search method described above to obtain the operators $P_{L-1,1}^{2}$ and $P_{L-1,2}^{2}$ using the feature points that correspond to the coefficients in $\mathbf{a}_{L-1,1}^{2}$ and $\mathbf{a}_{L-1,2}^{2}$, respectively.

We continue in a similar manner in the following decomposition levels. In level ℓ we first obtain the feature points $\mathbf{c}_i^{\ell}$ as weighted means of feature points from the finer level $\ell+1$. Then we use these feature points to obtain the operators $P_{\ell,s}^{t}$, running the approximate shortest path searches. Similarly to the GTBWT decomposition scheme [1], the relation between the feature points in a full decomposition can be described using tree-like structures. Each such "generalized" tree contains all the feature points which have participated in the calculation of a single feature point $\mathbf{c}_i^{1}$ from the coarsest decomposition level. Also, each feature point in the tree level ℓ is connected to all the points in level $\ell+1$ that were averaged in its construction. Fig. 3 shows an example of a "generalized" tree, which may be obtained for a data set of length $N=8$, using a filter $\bar{\mathbf{h}}$ of length 4 and disregarding boundary issues in the different levels. As the construction of these tree-like structures play an integral part in our proposed transform, we term it redundant tree-based wavelet transform (RTBWT).

We note that the computational complexities of both the RTBWT and the GTBWT are dominated by the number of distances that need to be calculated in their wavelet decomposition schemes. In [1], we employed the orthonormal transforms corresponding to several randomly constructed trees in order to apply a redundant transform. A full RTBWT decomposition, corresponding to redundancy factor of L, requires the calculation of $N^2 - 2N - (1/2)N(L-1) + 1$ distances. The method employed in [1] requires $L\left((2/3)N^2 - N - L + (4/3)\right)$ distance calculations in order to obtain a transform with a similar redundancy factor. Therefore for large N it requires about $(2/3)L$ times more distance calculations than the RTBWT. We next demonstrate the application of our proposed transform to missing label recovery and image denoising.

III. Label Recovery Using RTBWT

We next demonstrate the advantages of the RTBWT as a tool for representation and processing of functions on point clouds. We conduct several experiments on the white wine quality data set, described in [9]. The data set contains 4898 data points in $\mathbb{R}^{11}$ which describe physicochemical tests obtained for white variants of the Portuguese "Vinho Verde" wine. The target function is the wine quality, determined by experts, on a scale of 0 to 10. We first randomly select 2048 data points to be the points $\mathbf{x}_i$, and construct the vector $\mathbf{f}$ which contains the corresponding quality labels. We then calculate the RTBWT which corresponds to this data set, where we use the squared Euclidean distance to measure the dissimilarity between the feature points. Next, we carry out nine experiments in which we randomly select between 10 to 90 percent of the quality labels, discard them, and employ the dictionary which corresponds to the RTBWT in order to recover them.

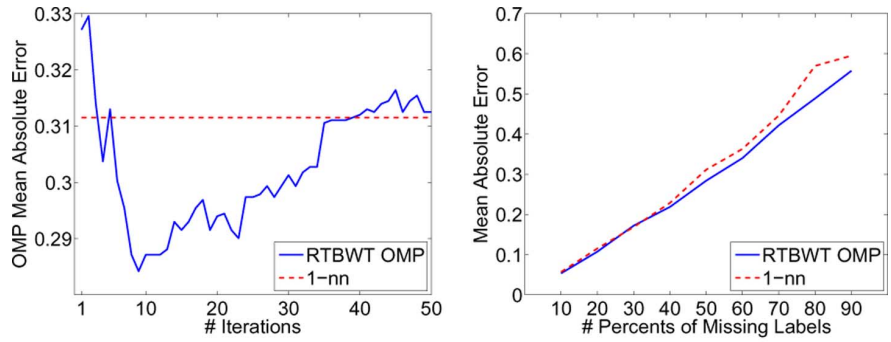
The signal $\mathbf{f}$ is recovered by finding the nonzero labels a sparse representation in the RTBWT dictionary, using the orthogonal matching pursuit (OMP) algorithm [10]. We compare our results to the ones obtained with the 1-nearest neighbor (1-nn) classifier. Fig. 4(a) depicts the mean absolute error (MAE) of the signal recovered by the OMP as a function of the iteration number, and the 1-nn classification error, in the case that 50 percents of the labels are missing. It can be seen that the RTBWT requires only nine dictionary elements to represent the best recovered signal which achieves lower error than the 1-nn classifier. Fig. 4(b) compares the MAE obtained with the two methods for different percents of missing labels. It can be seen that the RTBWT outperforms the 1-nn classifier in all the experiments but one.

IV. Image Denoising Using RTBWT

Let $\mathbf{F}$ be an image containing N pixels, and let $\tilde{\mathbf{F}}$ be its noisy version:

$$\tilde{\mathbf{F}} = \mathbf{F} + \mathbf{Z}. \tag{3}$$

$\mathbf{Z}$ denotes an additive white Gaussian noise independent of $\mathbf{F}$ with zero mean and variance σ^2. Our goal is to reconstruct $\mathbf{F}$ from $\tilde{\mathbf{F}}$ using the RTBWT. We choose the points $\mathbf{x}_i$ to be $\sqrt{n}\times\sqrt{n}$ patches, extracted from a padded version of $\tilde{\mathbf{F}}$, and the values f_i to be their middle pixels. The size of $\sqrt{n}$ ranges from 7 to 15. We use the squared Euclidean distance to measure the dissimilarity between the feature points in each level. We perform image denoising using the scheme proposed in [1], which employs subimage averaging, with the following differences. First, instead of using ten random variants of the GTBWT we employ a single nine-level RTBWT decomposition, which corresponds to a redundancy factor of 10. We also restrict the nearest neighbor searches performed for each patch to a surrounding



(a) (b)

Fig. 4. Label recovery results: (a) RTBWT OMP error vs. iteration number compared to 1-nn error (50% of the labels are missing). (b) RTBWT and 1-nn errors obtained for different percents of missing labels.

(a) (b)

Fig. 5. Denoising results for the image Barbara ($\sigma = 25$): (a) Noisy Barbara (20.18 dB). (b) Barbara denoised using RTBWT (30.76 dB).

TABLE I
DENOISING RESULTS FOR THE IMAGES BARBARA AND LENA OBTAINED WITH THE K-SVD (TOP LEFT), BM3D (TOP RIGHT), GTBWT (BOTTOM LEFT) AND RTBWT (BOTTOM RIGHT) ALGORITHMS. FOR EACH IMAGE AND NOISE LEVEL THE BEST RESULT IS HIGHLIGHTED

σ/PSNR	Lena		Barbara	
10/20.18	35.51	**35.93**	34.44	**34.98**
	35.87	35.88	34.94	34.93
25/28.14	31.36	32.08	29.57	30.72
	32.16	**32.17**	30.75	**30.76**

square neighborhood which contains $B \times B$ patches. Our experiments showed that this restriction both decreases the computational complexity of the transform and leads to improved denoising results. Finally, as the subimage averaging scheme described in [1] can be seen as a transform over that image patches, thresholding is performed by zeroing transform patches whose norm is smaller than a threshold T.

In order to assess the performance of the proposed image denoising scheme we apply it with the Symmlet 8 wavelet filter to noisy versions of the images Lena and Barbara, with noise standard deviations $\sigma = 10, 25$. The noisy and recovered Barbara images corresponding to $\sigma = 25$ can be seen in Fig. 5. For comparison, we also apply to the two images the K-SVD algorithm [6], the BM3D algorithm [7], and the GTBWT denoising scheme described in [1], with the search neighborhood and thresholding method described above. The PSNR of the results obtained with all the four denoising schemes are shown in Table I. It can be seen that the results obtained with the RTBWT and the GTBWT are almost identical, better than the ones obtained with the K-SVD algorithm, and close to the ones obtained with the BM3D algorithm. However, the RTBWT was about six times faster than the GTBWT since it required much less distance calculations.

V. CONCLUSION

We have proposed a new redundant wavelet transform applicable to scalar functions defined on graphs and high dimensional data clouds. This transform is the redundant version of the GTBWT introduced in [1]. We have shown that our proposed transform can be used for the recovery of labels defined on point clouds. We have also shown that the transform can be used for image denoising, where it achieves denoising results that are close to the state-of-the-art. In our future work plans, we intend to seek ways to improve the method that reorders the approximation coefficients in each level of the tree, replacing the proposed approximate shortest path search method. We also intend to apply the proposed transform to different image processing problems, and find new applications, which involve processing functions on graphs and point clouds.

ACKNOWLEDGMENT

The authors thank Dr. N. Rabin for the fruitful discussions and advices which helped in developing the presented work. The authors also thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] I. Ram, M. Elad, and I. Cohen, "Generalized tree-based wavelet transform," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4199–4209, 2011.
[2] S. Mallat, *A Wavelet Tour of Signal Processing, The Sparse Way*. New York: Academic, 2009.
[3] M. Shensa, "The discrete wavelet transform: Wedding the a trous and mallat algorithms," *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2464–2482, 1992.
[4] G. Beylkin, "On the representation of operators in bases of compactly supported wavelets," *SIAM J. Numer. Anal.*, vol. 29, no. 6, pp. 1716–1740, 1992.
[5] G. Plonka, "The easy path wavelet transform: A new adaptive wavelet transform for sparse representation of two-dimensional data," *Multiscale Model. Simul.*, vol. 7, pp. 1474–1474, 2009.
[6] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.
[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
[8] T. H. Cormen, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.
[9] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Dec. Support Syst.*, vol. 47, no. 4, pp. 547–553, 1998.
[10] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.