Facial Image Compression using Patch-Ordering-Based Adaptive Wavelet Transform

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

***Abstract*—Compression of frontal facial images is an appealing and important application. Recent work has shown that specially tailored algorithms for this task can lead to performance far exceeding JPEG2000. This paper proposes a novel such compression algorithm, exploiting our recently developed redundant tree-based wavelet transform. Originally meant for functions defined on graphs and cloud of points, this new transform has been shown to be highly effective as an image adaptive redundant and multi-scale decomposition. The key concept behind this method is reordering of the image pixels so as to form a highly smooth 1D signal that can be sparsified by a regular wavelet. In this work we bring this image adaptive transform to the realm of compression of aligned frontal facial images. Given a training set of such images, the transform is designed to best sparsify the whole set using a common feature-ordering. Our compression scheme consists of sparse coding using the transform, followed by entropy coding of the obtained coefficients. The inverse transform and a post-processing stage are used to decode the compressed image. We demonstrate the performance of the proposed scheme and compare it to other competing algorithms.**

***Index Terms*—Patch-based processing, redundant wavelet, compression.**

I. Introduction

In recent years, facial images are being extensively used and stored in large databases by social networks, web services, or various organizations such as states, law-enforcement, schools, universities, and private companies. Facial images are also expected to be stored in digital passports and ID cards. Thus, efficient storage of such images is beneficial, and their compression is an appealing application. The limitation to a specific and narrow family of images increases their combined spatial redundancy, and this allows algorithms that are specially tailored for the task of facial image compression, to surpass general purpose compression algorithms. More specifically, recent work [1]–[4] has shown that this kind of algorithms lead to performance far exceeding JPEG2000 [5].

In this paper, we introduce a novel algorithm for compression of facial images that exploits our recently developed redundant tree-based wavelet transform (RTBWT) [6]. This transform was originally designed to represent scalar functions defined on high-dimensional data clouds and graphs. However, we have shown in [6], [7] that this transform applicable to an image by converting it to a graph-structure, producing an image adaptive redundant and multi-scale decomposition that is highly effective for sparsifying its content. In this work, we bring this signal-adaptive transform to the realm of compression of aligned frontal facial images.

Given a training set of aligned face images, we construct a version of the RTBWT designed to sparsely represent these family of images. We compress an image by applying on it sparse coding using the RTBWT decomposition, quantizing the coefficients in the obtained representation, and then applying entropy coding. We decompress the image by placing the entropy decoded coefficients in a sparse vector, applying the RTBWT reconstruction, and applying a post-processing stage to the result. We demonstrate the performance of the proposed scheme both qualitatively and visually, and compare it to other competing algorithms.

The paper is organized as follows: In Section II, we explain how to calculate the RTBWT which sparsely represents a set of face images, and how to use it to obtain a sparse representation for such images. Section III introduces our proposed image compression scheme, and in Section IV we present experimental results that demonstrate its advantages.

II. The Sparsifying Transform

A. Sparse Representation of Facial Images

Let $\mathbf{y}$ be a column-stacked version of a face image, which contains N pixels. We assume that the image $\mathbf{y}$ follows the sparseland model [8], and therefore we can compress it by obtaining an efficient (sparse) representation for it. The sparseland model suggests that the image $\mathbf{y}$ can be sparsely represented using a redundant matrix $\mathbf{D}$ of size $N \times J$ $(J > N)$, which we term a dictionary. More specifically, let $\|\alpha\|_0$ denote the number of nonzero entries in a coefficient vector α. Then we expect that the solution of

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \ \|\alpha\|_0 \text{ subject to } \|\mathbf{D}\alpha - \mathbf{y}\|_2^2 \le \epsilon^2 \qquad (1)$$

should be sparse, i.e. $\|\alpha\|_0 \ll N$. This means that the image $\mathbf{y}$ can be represented using a small number of columns (termed atoms) from $\mathbf{D}$, with an error measured by the distance $\|\mathbf{D}\alpha - \mathbf{y}\|_2^2$. Naturally, there exists a tradeoff between the number of atoms used to represent $\mathbf{y}$ and the size of the representation error, i.e., the more atoms we use the smaller the error gets.

Problem (1) is an NP-hard problem, since it requires an examination of $O(J^N)$ possible non-zero supports for α. However, the matching and the basis pursuit algorithms [9]–[11] can be used quite effectively to obtain an approximation

I. Ram and I. Cohen are with the Department of Electrical Engineering, Technion – Israel Institute of Technology, Technion City, Haifa 32000, Israel. E-mail addresses: idanram@tx.technion.ac.il (I. Ram), icohen@ee.technion.ac.il (I. Cohen); tel.: +972-4-8294731; fax: +972-4-8295757. M. Elad is with the Department of Computer Science, Technion – Israel Institute of Technology, Technion City, Haifa 32000, Israel. E-mail address: elad@cs.technion.ac.il

The research leading to these results has received funding from the European Research Council under European Union's Seventh Framework Program, ERC Grant agreement no. 320649, and was supported by Robert H. Hillman Foundation for Global Security - Collaboration Technion and University Northeastern, and by the Israel Science Foundation (grant no. 1130/11), and by Japan Technion Society Research Fund.

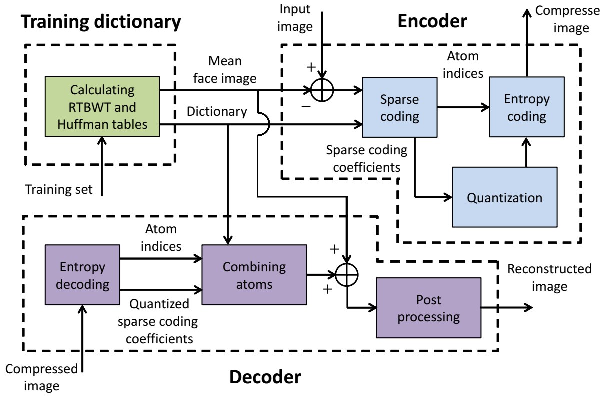


Fig. 1: Facial image encoding and decoding schemes.

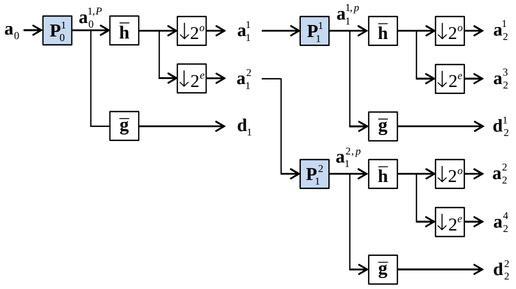


Fig. 2: RTBWT decomposition scheme

for the solution. In fact, these approximation techniques can be quite accurate if the solution is sparse enough to begin with [12]–[16]. In this work we make use of the Orthogonal Matching Pursuit (OMP) because of its simplicity [10] and efficiency. We next describe how we obtain the dictionary $\mathbf{D}$, which we use for sparsifying face images.

B. RTBWT-Based Dictionary for a Set of Images

Let $\mathbf{y}^g$, $g = 1\ldots, G$ be a training set, which contains the column-stacked versions of G aligned[1] face images, each containing N pixels. We wish to construct a redundant dictionary using the images in this training set, which will enable to sparsely represent them and similar facial images from a different test set. We note that while other methods usually train dictionaries which sparsely represent image patches, here we wish to find a dictionary that will be used to represent the entire image. To this end we make use of the redundant tree-based wavelet transform (RTBWT) [6], [7].

The RTBWT is a data-adaptive transform, providing a sparse and redundant representation for its input signal. In order to calculate the transform for an image $\mathbf{y}$, it requires that each pixel y_i will be associated with a feature vector $\mathbf{x}_i$, and it is assumed that under a distance measure $w(\mathbf{x}_i, \mathbf{x}_j)$ (e.g. Euclidean distance), proximity between two such feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ implies proximity between their corresponding pixels y_i and y_j. When working with an image, the features may be chosen to be patches centered around the pixel of interest [6], [7]. The transform is constructed by modifying the classical redundant wavelet transform [17], [18]. Figure 2 describes the decomposition scheme of the RTBWT. The filters $\bar{\mathbf{h}}$ and $\bar{\mathbf{g}}$ are the scaling and wavelet decomposition filters of a regular discrete wavelet transform, and they are applied using cyclic convolution. The $2:1$ decimators denoted by $\downarrow 2^o$ and $\downarrow 2^e$ keep the odd and even samples of their input, respectively. The signals $\mathbf{a}_\ell^s$ and $\mathbf{d}_\ell^s$ contain subsets of the samples in the approximation and detail coefficient vectors $\mathbf{a}_\ell$ and $\mathbf{d}_\ell$ in the ℓth scale, respectively, where $\mathbf{a}_0 = \mathbf{y}$.

The operators $\mathbf{P}_\ell^s$ make the difference between our proposed wavelet decomposition scheme and the common redundant wavelet transform [17], [18]. *Each such operator produces a permuted version* $\mathbf{a}_\ell^{s,p}$ *of its input vector* $\mathbf{a}_\ell^s$. This may be interpreted as a linear and unitary operator given that vector. These operators increase the regularity of the approximation coefficient signals in the different levels of the decomposition scheme and cause their representation with the RTBWT to be more sparse. The reordering operators are obtained by organizing the feature vectors, calculated from the patches, such that they are chained in the shortest possible path [6], [7], [19], [20]. Thus, essentially an approximation to the solution of the traveling salesman problem (TSP) [21] is obtained. For example, let $\{\mathbf{x}_j^p\}_{j=1}^N$ denote the patches $\{\mathbf{x}_i\}_{i=1}^N$ in their new order, then $\mathbf{P}_0^1$ is obtained by minimizing the measure

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

We note that the RTBWT reconstruction scheme is obtained in a similar manner by adding the operators $(\mathbf{P}_\ell^s)^{-1}$ into the redundant wavelet transform reconstruction scheme.

We now move to discuss the specifics of facial image compression. We first average the images in the training set and obtain a mean face image. This image contains information shared by all the images in the training set, and therefore we subtract it from every training image in order to obtain a more efficient representation for it. Next, since we want the transform to sparsely represent *all* the images in the training set, we need to associate a single feature point with every set of G pixels that are located in the same index in each of the training images. Thus, we construct a $G\times N$ matrix $\mathbf{Y}^G$, which contains in its rows the images $\mathbf{y}^g$, and choose its kth column to be the feature vector $\mathbf{x}_k$ associated with the kth pixel y_k^g in all of the training images. Note that this is different from the common practice mentioned above of using spatial patches, and in our scheme, *the same permutation operators are applied to all train images*. We choose the distance function w to be the Euclidean distance. Having defined the feature points and the distance function, we use them to construct the RTBWT according to the scheme described above, and in [6].

We next denote by $\mathbf{\Phi}$ and $\mathbf{\Psi}$ the matrices that apply the RTBWT decomposition and reconstruction, respectively, and choose our dictionary $\mathbf{D}$ to be a version of $\mathbf{\Psi}$, whose atoms have been normalized to have a unit norm. We note that because of its large size, the matrix $\mathbf{D}$ is not explicitly calculated nor stored. Instead, in order to multiply vectors

[1] Geometrical pre-aligning of the facial images is crucial to any method that aims for effective compression. Indeed, previous work [1], [2], [3] used this, and here we assume the availability of an aligned set.

by $\mathbf{D}$ and $\mathbf{D}^T$ within the OMP algorithm, it is required to apply the RTBWT reconstruction and decomposition schemes, respectively, on these vectors.

III. Facial Image Compression Scheme

Our proposed facial image encoding and decoding schemes are shown in Figure 1. We assume that we are given training and test sets containing aligned facial images. We average the images in the training set and subtract from them the mean image. We calculate a dictionary from the resulting images as described in the previous section, and then encode every one of these images using the following procedure: 1) We apply sparse coding to the image using the OMP algorithm to obtain a small set of coefficient values and their corresponding indices. 2) We replace the coefficient indices by the differences between the indices of consecutive coefficients, split the coefficient values into low and high ranges, and apply uniform quantization to the values in each range. 3) We calculate two different Huffman tables, one for the coefficient values and the other for their indices, relying on the statistics of their occurrences in the sparse representations of all the images in the training set. *All the aforementioned calculations are done offline, and we assume that the obtained RTBWT dictionary (along with its defining permutations) and Huffman tables are known both to the encoder and the decoder, along with the mean face image, and therefore they do not need to be transmitted as side information.*

Encoding an image from the test set starts with subtracting from it the mean face image, and applying the same encoding procedure that was applied above in order to find the significant representation coefficients. We then perform entropy coding by applying the corresponding Huffman tables to the resulting sets of coefficient indices and values, and obtain the compressed image. We decode such an image by first applying entropy decoding, thus obtaining the quantized coefficient values and the index differences. We recover the coefficient indices from their differences, and use them and the corresponding quantized coefficient values to construct a sparse representation. The image is reconstructed by simply applying the RTBWT reconstruction to this sparse vector, and adding the mean face image to the result.

In order to further improve the quality of the obtained image, we use a post-processing scheme, which is a variation on the one proposed in [3]. This scheme consists of applying to the reconstructed image N different filters of size 5×5, each centered around a different pixel. We use the training images to learn several different sets of N filters, each corresponding to a different range of bit-rates. These filters are obtained as follows. We apply the encoding and decoding schemes to each of the training images $\mathbf{y}^g$, and arrange the obtained images $\hat{\mathbf{y}}^g$ as the rows of a $G \times N$ matrix $\hat{\mathbf{Y}}^G$. Now, let $\mathbf{h}_k$ be a column stacked version of the 5×5 filter applied to a reconstructed image in the location of its kth pixel. Also, we denote by $\mathbf{e}_k$ and $\mathbf{R}_k$ a vector and a matrix, whose right multiplications with $(\hat{\mathbf{y}}^g)^T$ extract the kth pixel and the transposed column-stacked version of the surrounding 5×5 patch, respectively. Then the filter $\mathbf{h}_k$ is obtained by solving the following least squares problem

$$\begin{aligned}\hat{\mathbf{h}}_k &= \underset{\mathbf{h}_k}{\operatorname{argmin}} \ \|\hat{\mathbf{Y}}^g \mathbf{R}_k \mathbf{h}_k - \mathbf{Y}^g \mathbf{e}_k\|_2^2 \\ &= \left[\mathbf{R}_k^T (\hat{\mathbf{Y}}^g)^T \hat{\mathbf{Y}}^g \mathbf{R}_k\right]^{-1} \mathbf{R}_k^T (\hat{\mathbf{Y}}^g)^T \mathbf{Y}^g \mathbf{e}_k. \qquad (3)\end{aligned}$$

This process is repeated for each pixel, and it is part of the off-line training process.

IV. Experimental Results

We assess the performance of our compression scheme on a database[2] containing 4515 grayscale asian face images with 8 bits per pixel. These images are the same ones used in [2], and they undergo the same preprocessing stage as in [2] – alignment according to the scheme proposed in [1] followed by a scale-down by a factor of 2. We use a random subset of 4415 aligned images of size 221×179 as the training set, and the remaining 100 aligned images as the test set.

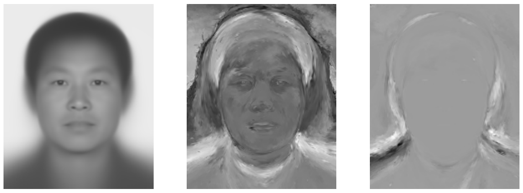
We start by calculating from the training images a mean face image, shown in Figure 3(a). We can see that this image contains a relatively sharp face, whose facial features are shared by all the training images. We subtract this image from all the images in the training set, and construct the RTBWT with the resulting images. We use a 13-level wavelet decomposition with the Symmlet 8 filter, and obtain a dictionary with redundancy factor of 14. Figures 3(b) and (c) contain examples of two atoms from this dictionary, which correspond to the coefficients with the second and fourth largest magnitudes, out of the ones used to represent the top-center image in Figure 5. It can be seen that the atoms are either images of complete faces, or images containing details around face edges. We then encode all the images in the training set and calculate one Huffman table for 128 coefficient values, and another for 1024 index difference values.

We next use our proposed compression scheme to encode and decode each image in the test set, and use the results (in PSNR) obtained with and without post-processing to calculate two rate-distortion curves. We compare these curves to the ones obtained by repeating this procedure with the common redundant wavelet transform (RWT) replacing the RTBWT in our scheme, and to rate-distortion curves obtained with JPEG2000[3], the algorithm described in [2] which is based on the K-SVD, and its improved version [3] which consists of a post-processing stage. We note that we used the matlab function "imwrite" to compress images in JPEG2000 format.

First it can be seen that the post-processing stage improves the performance of our scheme for all bit-rates both when the RTBWT and the RWT are used. Also, applying our scheme with the RTBWT improves its results by at least 8 dB compared to ones obtained with the RWT, both when post-processing is used and when it is not. Further, even without using post-processing our RTBWT-based algorithm outperforms JPEG2000, for all compressed-image sizes which

[2] We chose this database since it contains thousands of images of both men and women in varying ages, and it was used by previous papers by our group [2], [3], that provide results which serve as reference to compare against.

[3] Since we test performance on very low bit rates, for a fair comparison we removed a fixed header size of 100 bytes from the JPEG2000 curve.



(a) (b) (c)

Fig. 3: (a) Mean face image. (b) Two atoms from the RTBWT dictionary.

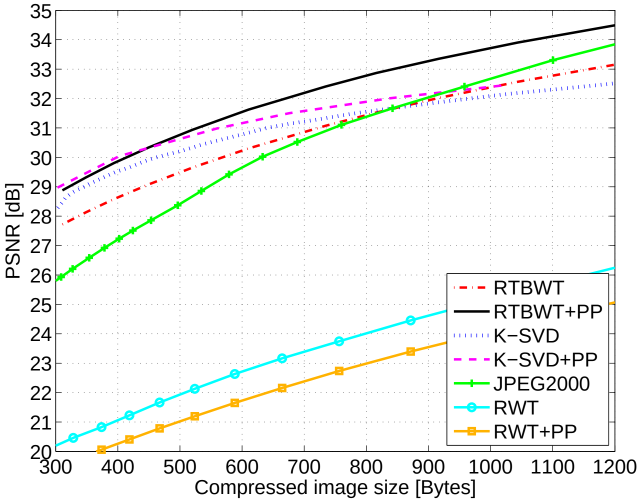
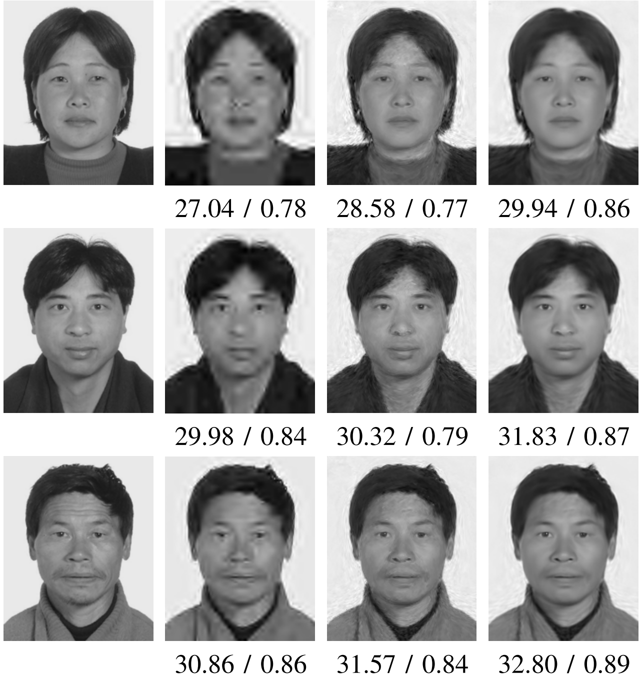


Fig. 4: Rate distortion curves obtained with JPEG2000, the methods in [2] (K-SVD) and [3] (K-SVD+PP), and our proposed scheme using the RWT and the RTBWT with and without post-processing (PP).

are lower than 1140 bytes. Our full RTBWT-based algorithm outperforms JPEG2000 for all bit rates: From a gain of 3 dB for low bit-rates to a gain of 0.6 dB for high bit-rates. Finally, without post-processing our RTBWT-based algorithm outperforms the algorithm in [2] for compressed-image sizes higher than 850 bytes, but obtains inferior results for smaller sizes. However, our full algorithm performs similarly to the algorithm in [3] for compressed-image sizes smaller than 450 bytes, but outperforms it for higher sizes even by more than 1 dB for compressed-image sizes higher than 1000 bytes.

We next demonstrate the visual quality of the results obtained with our RTBWT-based compression scheme when low bit-rates are used. Figure 5 compares both visually and in terms of PSNR and SSIM [22] the reconstructed images obtained for compressed-image sizes of 400, 600, and 800 bytes with our scheme, with and without post-processing, and with JPEG2000 with a header size of 100 bytes removed. It can be seen that our scheme obtains higher PSNR values than JPEG2000, and that post-processing further decreases these errors. In terms of SSIM, our scheme with post-processing outperforms JPEG2000. Without post-processing our results contain artifacts that look like paint-brush strokes, causing a reduced SSIM. These artifacts are greatly reduced using the post-processing, and the obtained images are of relatively high quality despite the low bit-rates.



27.04 / 0.78 28.58 / 0.77 29.94 / 0.86

29.98 / 0.84 30.32 / 0.79 31.83 / 0.87

30.86 / 0.86 31.57 / 0.84 32.80 / 0.89

Fig. 5: Facial image compression results (PSNR / SSIM) with compressed-image sizes of 400 bytes (first row), 600 bytes (center row), and 800 bytes (right row). The original images (first column) are compressed using JPEG2000 (second column), and our scheme without post-processing (third column) and with it (last column).

V. Conclusion

We have proposed a new face image compression scheme based on the redundant tree-based wavelet transform (RTBWT). We learn the transform from a training set containing aligned face images, and use it as a redundant dictionary when we encode images by applying sparse coding on them. Improved quality results are obtained by using a filtering-based post-processing scheme. We have demonstrated competitive performance compared to other methods, and managed to obtain results of high visual quality for low bit-rates.

There are several research directions to extend this work that are currently considered. A first direction is to learn a set of indices of leading coefficients, which is shared by all encoded images. These indices will be known to the decoder, and therefore only their corresponding values will be sent by the encoder, thus achieving better compression. A different direction is to train different dictionaries for different parts of the image or for different sub-groups of images in order to obtain dictionaries which are more adapted to the data. Such dictionaries may lead to a sparser image representation and improved quality of the reconstructed images. Finally, the performance of our scheme may also be improved by replacing the entropy coding technique it uses from Huffman coding to arithmetic coding. Then we may compare its results to those of the advanced HEVC compression scheme [23] which reduces the bit rate by about 20% compared to JPEG2000 [24].

Acknowledgements

The authors thank Ori Bryt, the associate editor and the anonymous reviewers for their helpful comments and help.

References

[1] M. Elad, R. Goldenberg, and R. Kimmel, “Low bit-rate compression of facial images,” *IEEE Trans. Image Processing*, vol. 16, no. 9, pp. 2379–2383, 2007.

[2] O. Bryt and M. Elad, “Compression of facial images using the k-svd algorithm,” *Journal of Visual Communication and Image Representation*, vol. 19, no. 4, pp. 270–282, 2008.

[3] ——, “Improving the k-svd facial image compression using a linear deblocking method,” in *Proc. 25th IEEE Convention of Electrical and Electronics Engineers in Israel*. IEEE, 2008, pp. 533–537.

[4] J. Zepeda, C. Guillemot, and E. Kijak, “Image compression using the iteration-tuned and aligned dictionary,” in *Proc. 36th IEEE Internat. Conf. Acoust. Speech Signal Process., ICASSP-2011*, 2011, pp. 793–796.

[5] D. S. Taubman and M. W. Marcellin, “Jpeg 2000: Image compression fundamentals, standards and practice,” 2001.

[6] I. Ram, M. Elad, and I. Cohen, “Redundant Wavelets on Graphs and High Dimensional Data Clouds,” *IEEE Signal Processing Letters*, vol. 19, no. 5, pp. 291–294, 2012.

[7] I. Ram, I. Cohen, and M. Elad, “Patch-ordering-based wavelet frame and its use in inverse problems,” *IEEE Trans. Image Processing*, vol. 33, no. 7, pp. 2779–2792, 2014.

[8] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Verlag, 2010.

[9] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[10] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *presented at the 27th Annu. Asilomar Conf. Signals, Systems, and Computers*. IEEE, 1993, pp. 40–44.

[11] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Rev.*, vol. 43, no. 1, pp. 59–129, 2001.

[12] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Trans. Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[13] ——, “Just relax: Convex programming methods for subset selection and sparse approximation,” *IEEE Trans. Information Theory*, vol. 51, no. 3, pp. 1030–1051, 2005.

[14] D. L. Donoho and X. Huo, “Uncertainty principles and ideal atomic decomposition,” *IEEE Trans. Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[15] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell 1$ minimization,” *Proc. National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[16] D. L. Donoho, M. Elad, and V. N. Temlyakov, “Stable recovery of sparse overcomplete representations in the presence of noise,” *IEEE Trans. Information Theory*, vol. 52, no. 1, pp. 6–18, 2006.

[17] M. Shensa, “The discrete wavelet transform: Wedding the a trous and mallat algorithms,” *IEEE Trans. Signal Processing*, vol. 40, no. 10, pp. 2464–2482, 1992.

[18] 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.

[19] I. Ram, M. Elad, and I. Cohen, “Generalized Tree-Based Wavelet Transform ,” *IEEE Trans. Signal Processing*, vol. 59, no. 9, pp. 4199–4209, 2011.

[20] ——, “Image processing using smooth ordering of its patches,” *IEEE Trans. Image Processing*, vol. 22, no. 7, pp. 2764 – 2774, 2013.

[21] T. H. Cormen, *Introduction to algorithms*. The MIT press, 2001.

[22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[23] G. J. Sullivan, J. Ohm, W. J. Han, and T. Wiegand, “Overview of the high efficiency video coding (hevc) standard,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, 2012.

[24] P. Hanhart, M. Rerabek, P. Korshunov, and T. Ebrahimi, “Subjective evaluation of hevc intra coding for still image compression,” in *Seventh International Workshop on Video Processing and Quality Metrics for Consumer Electronics-VPQM 2013*, 2013.