THE HIERARCHICAL SIGNAL DEPENDENT TRANSFORM: CREATING ORTHONORMAL BASIS THAT MATCH LOCAL SIGNAL CHARACTERISTICS

Vanessa Testoni[†], *Max H. M. Costa*[‡] *and Dinei Florêncio*[*]

[†] ECE, University of California at San Diego, California, USA
[‡] DECOM-FEEC, University of Campinas, São Paulo, Brazil
[*] Microsoft Research, Washington, USA

ABSTRACT

Hierarchical transforms are widely used in image and video coding to produce multilevel decomposition of signals. After applying these transforms, same level signals are typically uncorrelated. However, there is often still significant cross level information. Traditionally, this cross-level information is exploited at the entropy coding step, but not at the transform step. The main contribution of this work is an energy compaction technique/transform that can also exploit these cross-resolution-level structural similarities. The core idea of the technique is to include in the hierarchical transform a number of adaptive basis functions derived from the lower resolution of the signal. A full image codec was developed in order to measure the performance of the new transform. Results are presented in terms of transform coding gain, energy concentration and distortion versus rate curves compared with standard JPEG, JPEG 2000 and JPEG XR.

Index Terms— Signal dependent transform, Hierarchical block-based transform, Image coding

1. INTRODUCTION

Linear transforms are a key tool in signal compression. They remove correlation between adjacent samples, concentrating the signal energy in a few coefficients, with associated gains in compression efficiency. Most modern codecs use hierarchical transforms, which allow to capture image correlation at different resolution levels, with reasonable complexity. One of the characteristics of these hierarchical transforms is that after its application, although the dependency among the subbands in the same level is highly reduced, there is still dependency among the subbands across different levels. This dependency has been successfully exploited at the entropy coding codec stage since the *zerotrees* concept was introduced by the Embedded Zerotrees Wavelet (EZW) algorithm [1]. However, this dependency has not yet been effectively exploited at the transform codec stage. Some recent work makes great progress in exploiting some of the cross-band, cross-region, or cross-color similarities [2, 3, 4]. However, they are tipically target at exploiting one particular aspect of these similarities, and it is hard to group them, or extend to other types of similarities.

In this paper we propose a framework that allows any type of similarity to be exploited in an unified way. The new transform presented here is hierarchical, adaptive and signal dependent. Based on these features, it was named HSDT (Hierarchical Signal Dependent Transform). The characteristics of the HSDT provide several advantages. As it is hierarchical, both spatial and quality scalability are naturally provided.

Additionally, the framework is flexible enough that to allow both overlapping or block transforms to be designed. In this paper we discuss mostly the block-based HSDT construction, which is simpler. Unlike other hierarchical transforms, the block-based HDST can be designed to allow unrestricted data access to previous hierarchical level or other regions of the current level. Yet, it does not significantly increase complexity, because it is applied in a block-based fashion with blocks of fixed size 8x8. The common blocking artifacts resulting from the applications of block transforms, specially at low bit-rates, are less perceptible as they are spread among the several hierarchical levels. As the bit-rate decreases, the distortion takes the form of a gradually increasing blurriness or loss of details. Additionally, the adaptation scheme exploits the signal dependency between levels by taking advantage of both directionality and neighborhood correlation properties of most images.

As expected by the adaptation process, each block in each level will be coded with a different and optimized basis. The results of a set of adaptive functions applied to the lower resolution signal will compose this optimized basis. These results are ordered as basis vectors in a way that the results of the best functions, i.e., the functions that give the maximum projections, are placed in the initial basis positions. To order the functions appropriately, several statistical measures, such as standard deviation and local correlation, are initially computed for the block itself and its neighborhood. These measures define block classes, e.g., uniform blocks in directional neighborhoods. The classes are then associated with the best functions to represent blocks with the specific characteristics described by the classes. This association between classes and best functions is defined by a previous extensive machine learning process employing several blocks from several images.

Like the basis pursuit coding scheme [5], the set of adaptive functions in the HSDT can be seen as a dictionary formed and ordered by a learning process. However, the results of the adaptive functions are the basis vectors, not the functions. Unlike the basis pursuit scheme, all basis vectors are orthonormalized, resulting in a complete basis, not in an overcomplete basis. Additionally, all coefficients resulting from the application of the basis to each block are scanned and entropy coded. Therefore, there is no sparse coding.

The transform process generation will be detailed in Section 2. Sections 3 discusses the results in terms of transform coding gain, as well as details of the associated image codec. Finally, Section 4 makes concluding remarks, and plots future directions.

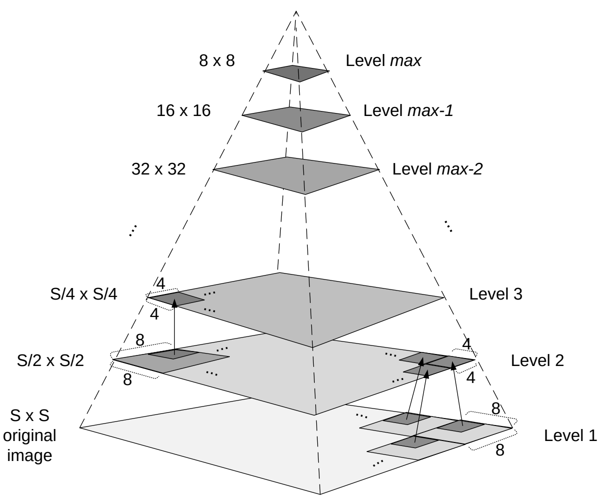

Fig. 1. 8x8 HSDT application process arranged in the form of a *max*-level pyramid.

2. THE HIERARCHICAL BLOCK-BASED SIGNAL DEPENDENT TRANSFORM

Coefficients of hierarchical transforms (e.g., wavelets) are often divided into two groups, and called “approximation” and “detail” coefficients. After each stage in the hierarchy, the transform is then applied again to the “approximation” coefficients, generating the multiresolution versions of the image we are used to see. Thus, the role of the approximation coefficients is to produce a lower resolution image, while the role of the ”detail” coefficients is to represent the fine details removed while producing such lower resolution image. It is important to our discussion to note that - because traditional transforms are linear and fixed - it does not matter in which order we compute the coefficients. Thus, for simplicity, traditional hierarchical decompositions are computed level by level, typically from highest resolution to lowest. In contrast (and similar to some lifting approaches), the HSDT is computed in two cycles: first, all levels of the “approximation” coefficients are computed, and only then are the detail coefficients computed, starting at the lowest resolution. We now give some details of the construction of the HSDT.

The HSDT is applied in this work in a block-based fashion with blocks of fixed size 8x8. Furthermore, we adopt the 2-D DCT basis for the lowest 4x4 coefficients. More elaborate designs are certainly possible, and will produce additional gains. However, we chose the 8x8 block design based on the DCT because it is simple, flexible, and able to show most properties of the proposed framework by contrasting to one of the most widely used transforms.

More specifically, at each level the lowest 4x4 coefficients of an 8x8 2-D DCT are computed. An 4x4 inverse DCT is then applied to these coefficients, thus producing an reduced resolution (4x4 pixels) version of the original (8x8 pixels) block. These procedure will be repeated hierarchically until the lowest desired resolution is achieved. This imply that, at the next level, four of these 4x4 reconstructed blocks will be concatenated, producing an 8x8 pixel block, which undergoes the same computation. This procedure is graphically illustrated in the form of a pyramid in Fig. 1.

The objective of this first step is simply to produce a multiresolution image pyramid. Any scaling function could be used, with the only constrain that the lower resolution image preserve enough information to avoid dimensionality expansion. The choice of the two-dimensional DCT allow us to obey this constrain for each 8x8 block. The same would also apply to any other non-overlapping block transform. Overlapping transforms could equally be used, but would require independence and orthogonality considerations over the whole image, which would make the analysis on this paper unnecessarily complex.

After the scaling functions of the hierarchical transform are applied in each decomposition level, next step is to compute the detail coefficients. In the end, the approximation coefficients can be seen as an intermediate computation level, since (except for the lowest level) the detail coefficients are the only ones actually transmitted. The adaptation process of the HSDT starts at the highest decomposition level. For each block, at each level, we will design an orthogonal basis to code the residual information down the pyramid structure as efficiently as possible. We will do this by designing an orthonormal basis using all information available at that point. This includes the full lower resolution image, as well as all previous blocks at the current resolution level. A set of adaptive functions, which will detailed in Section 2.1, is applied to each 4x4 full block, i.e. the block with dimensionality 64, of the lower resolution image. These functions should produce 8x8 full blocks as similar as possible to each 8x8 full block of the current level. If this similarity is successfully achieved, the representation of these computed 8x8 full blocks with dimensionality 48 will match the residual block being coded. Therefore, an optimized basis for this residual block can be obtained by including these 8x8 computed blocks as basis vectors in the transform.

As all basis vectors in the transform are applied to the whole residual 8x8 block, the HSDT is not separable and must be arranged in its 64x64 matrix form. Since the adaptive basis is generated only to code the residual blocks, the first 16 rows of the transform matrix are never modified, while the other 48 rows are adapted. Therefore, the first 16 rows of the HSDT transform matrix are exactly the scaling functions of the multilevel DCT which are already linearly independent. However, each basis vector resulting from the adaptive functions that is included in the matrix is linearly dependent and must be orthonormalized with respect to the first 16 rows and to the other basis vectors previously included into the matrix. This orthonormalization process is performed row-by-row until the basis is complete. Consequently, at the end of the HSDT matrix generation process, a complete orthonormal optimized basis is obtained.

It is important to note that the same basis can be generated both at the encoding and decoding processes, once the adaptation depends only on the lower resolution image computed at the previous decomposition level. So, there is no overhead.

In the linear algebra view of the HSDT application, the residual block is initially read by columns in order to form a block vector. Then, the inner product of this block vector and the basis vectors is computed to obtain the resulting coefficients. The approximation coefficients (first 16) will be null, once the residual block has dimensionality 48, and the first residual coefficient (17^{th} coefficient) will concentrate most of the residual block energy, if the adaptive functions are successfully designed.

2.1. Adaptive Functions

The performance of the HSDT is highly dependent on the performance of the adaptive functions. If the adaptive functions can produce basis vectors that, after the orthonormalization process, are similar to the residual block, the transform will achieve its goal of high energy concentration. In this initial study, two natural image

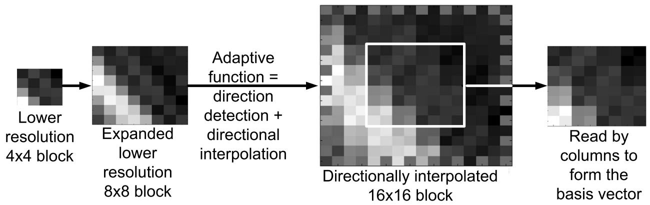

Fig. 2. Application of a directional adaptive function to an expanded 8x8 lower resolution block (to reduce blocking artifacts) in order to produce a basis vector.

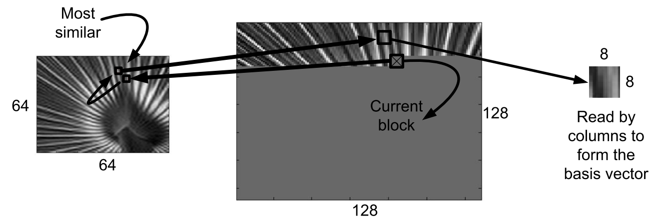

Fig. 3. Application of a block matching adaptive function to a 4x4 lower resolution block in order to produce a basis vector.

properties are exploited by the adaptive functions: directionality and correlation. We believe many higher level aspects of image can be exploited in the framework introduced in this paper. For example, by looking at similar blocks at the current level in relation to previous level, the sharpness of the image at current level can be predicted and incorporated in the adaptive basis. Similar more can be done in this respect, and will significantly improve performance.

2.1.1. *Directional Adaptive Functions*

For most images, correlation changes locally and is typically stronger along a certain direction, not necessarily being horizontal or vertical. Based on these considerations, several directional transforms have been proposed [6, 7, 8, 9]. These transforms can be classified into two categories. In the first category, new transforms are introduced to incorporate directional bases, like curvelets [6]. The transforms of the second category are modified from existing transforms, such as the directional wavelet transform [7], the directional DCT transform [8] and the directional lapped transform [9].

The scheme proposed to design directional adaptive functions in this work is from the first category and consists of two steps. All adaptive functions are applied to the 4x4 full blocks of the lower resolution image and must produce 8x8 full blocks. Therefore, in the first step, the dominant direction of the 4x4 block is identified, provided that the block has a strong correlation along a certain direction. This step is referred to as *direction detection*. In the second step, the block is interpolated in the dominant direction to generate the 8x8 block from the 4x4 block. This step is, then, referred to as *directional interpolation*. An example of the results obtained with these techniques is shown in Fig. 2.

The combination of direction detection and directional interpolation techniques is suitable to produce directional adaptive functions for the HSDT based on the assumption that lower resolution blocks with strong directional patterns show the same strong directional patterns in the corresponding residual block.

Several direction detection techniques are employed by HSDT and can be separated into three categories:

1) Hough transform using the principle of template matching [10];
2) Texture descriptors that compute spatial activity in specific directions, such as LAS (Local Activity Spectrum) [11] and SASI (Statistical Analysis of Structural Information) [12];
3) Orthogonal regression using principal components analysis.

The dominant direction of the block is determined by one of the direction detection techniques and the interpolation is performed employing directional and Euclidean weights enclosing a 12-point or 16-point neighborhood. Each combination of the techniques forms a directional adaptive function, resulting in a set of 30 different functions.

2.1.2. *Block Matching Adaptive Functions*

While the directional adaptive functions exploit the natural blocks directionality, the block matching adaptive functions exploit the natural local and non-local blocks correlations. It is well recognized that pixels in images usually have a strong correlation to their neighboring pixels, known as local correlation. In addition, across an image, there are potentially similar structures and repeated patterns, exhibiting strong non-local correlations.

The scheme proposed to design block matching adaptive functions in this work takes advantage of the correlation among the 4x4 full blocks of the lower resolution image to obtain the basis vector to be included in the transform matrix. As indicated by Fig. 3, the block matching algorithm is applied only to the full blocks of the lower resolution image. Initially, the corresponding lower resolution block of the current block being encoded/decoded is localized. Then, the most similar block to the lower resolution block is identified. Finally, the corresponding full 8x8 block of this most similar block is localized and read by columns to form the basis vector. It is important to note that, unlike performed by the directional adaptive functions, the blocks read as columns to form the basis vectors are not produced by directional interpolation. These blocks are directly extracted from the already encoded/decoded blocks of the current full image.

The block matching algorithms in this work employ a small window search of 3x5 blocks, i.e., 12x20 values and the matching computations can be performed in two stages using SAD (Sum of Absolute Differences) as the distance metric. In the first stage, only the central 4x4 areas of connected blocks, are compared with the current block. In the second stage, only the best match area is expanded to a 8x8 area and new comparisons are performed pixel-by-pixel in a full search procedure. A variation of this technique is to not only select the best match block, but compute the mean of the 5 best match blocks.

Like described for the directional adaptive functions, the block matching adaptive functions also employ texture descriptors. The difference is that, unlike the directional functions, here all the values computed for each descriptor, not only the largest autocorrelation coefficient considered for the direction detection, are taken into account. The descriptors are composed of a set of values and several distance metrics are employed in order to measure the similarity among the descriptors computed for each block. Another difference is that not only descriptors that detect directional features in the blocks are computed, such as LAS and SASI. Texture descriptors that describe the neighborhood of each value inside the block are also employed, such as the LBP (Local Binary Pattern) [13]. Consid-

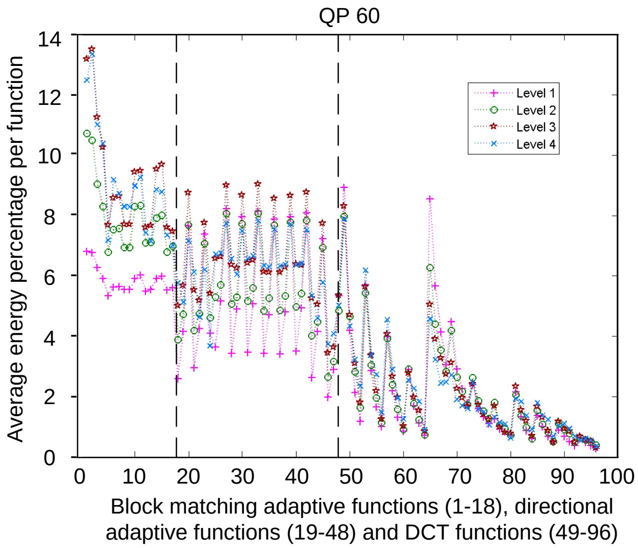

Fig. 4. Performance evaluation for adaptive and DCT functions in terms of average energy percentage per function.

ering all the possible combinations, a total of 18 different blocking matching adaptive functions arises.

2.1.3. Performance Evaluation of Basis Functions

As 30 directional functions and 18 blocking matching functions resulted from the combination of the techniques previously described, an initial experiment was conducted to evaluate the average performance of the functions over several images from a large database of perceptually diverse content [14]. The evaluation was performed with several QPs (Quantization Parameters) and also included the 48 residual wavelet functions of the multilevel two-dimensional DCT. The results obtained with QP 60 are shown in Fig. 4. The metric employed to evaluate the functions was the average energy percentage represented only by the first residual coefficient with respect to the total residual block energy.

Several conclusions can be taken from the graph in Fig. 4. The performance of the block matching adaptive functions in the initial 18 positions is better, in general, than the performance of the directional adaptive functions in the following 30 positions. The adaptive functions indeed achieve, in the average, higher energy concentrations than the DCT functions. Besides, the adaptive functions achieve more energy concentration in higher decomposition levels (3 and 4), while the DCT functions achieve their best energy concentration results in lower levels (1 and 2).

An additional important result is a best-in-the-average scanning order for the two-dimensional DCT functions. An interesting aspect of this result is that the first 10 functions read according to the best-in-the-average order are exactly the lower-frequency functions of the residual two-dimensional DCT subbands. This order is referred to as *adaptive order* in Fig. 5, while the traditional zig-zag scanning order is referred to as *fixed order*. It can be seen that the *adaptive order* compacts more energy in the first residual coefficients and provides a smoother decay curve. Readers are directed to the thesis in [15] for more details.

2.2. Basis Vectors Ordering

It is important to note that the performance of the adaptive functions in Fig. 4 is the average performance across several images. Individual performances per block can achieve much better results, such as 78% of the total block energy represented only by the first residual coefficient. The considerable variance in the functions performance raised the need to identify the "goodness" of each adaptive function with respect to the specific characteristics of each residual block. For instance, an uniform residual block localized in a correlated image area could be successfully coded with a block matching adaptive function. On the other hand, a directional residual block localized in a non-correlated area probably would be better coded with a directional adaptive function. And finally, a non-directional residual block in a non-uniform area would not benefit much from adaptive functions and should be coded with the standard two-dimensional DCT.

Therefore, it is clear that the characteristics of each residual block must be measured and the best adaptive or DCT functions to code each type of residual block must be identified. This process was performed in three steps in this work. In the first step, the following statistical measures considering not only the residual block itself but also its neighborhood were computed:

1) Standard deviation;
2) Spatial activity to measure block correlation;
3) Standard deviation of the spatial activities to differentiate uniform blocks from directional blocks;
4) SAD to measure neighborhood correlation;
5) Standard deviation of the SADs to differentiate uniform neighborhoods from directional neighborhoods.

In the second step, the best 15 adaptive functions and the best 10 DCT functions were selected according to the average performance results shown in Fig. 4. And finally, in the third step, the relation between the energy concentration performance of each selected adaptive/DCT function and the statistical measures per block is found through an extensive training process employing multiple linear regression. For all blocks of the several images in [14], the five statistical measures were stored and the performance of the adaptive/DCT functions was also stored. At the end of the training process, multiple linear regression was applied to minimize the sum of the squares of the distances between the hyperplanes and the statistical measures. One hyperplane is determined for each of the 10 selected DCT functions and each of the 15 selected adaptive functions. The equation of the hyperplane is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 \qquad (1)$$

where x_{1-5} are the 5 statistical measures, y gives the approximate energy concentration achieved by each function for a block with characteristics defined by the statistical measures x_{1-5} and β_{1-5} are the *coefficient regressors*. The *coefficient regressors* are the main result of the training process and are stored to be used on-the-fly in the encoding/decoding processes of each block. The function chosen to encode/decode the block on-the-fly is the function which returns the highest y value.

Not only the best function for the first transform matrix position must be given through Eq. (1), but also the best functions for all other matrix positions. If the first basis vector represents 50% of the total residual energy, but the other 50% of the energy is spread across the remaining 47 basis vectors or even concentrated in the last basis vectors, the overall performance of the transform will not be satisfactory. It is important not to place functions that exploit similar characteristics of the blocks together. For instance, if the first and the second basis vectors are very similar, after the orthonormalization, the second basis vectors will become too randomized and most of the energy it could represent would have already been represented by the first basis vector.

This method of ordering the basis vectors is referred to as *regression hyperplanes order* in Fig. 5. The other method applied in this work to order the basis vectors employs the Karhunen-Love transform (KLT) and is referred to as *KLT order* in Fig. 5. The KLT is applied to the selected adaptive functions and decomposes them as a linear combination of the *principal components* of the functions. Therefore, the functions are uncorrelated and the common good characteristics of the best functions are reinforced, while the uncommon undesirable characteristics are weakened. After the eigendecomposition, the eigenvalues are sorted in descending order and only the initial corresponding eigenvectors are kept in order to form the final basis. The remaining positions of the transform matrix are filled with the functions of the default two-dimensional DCT ordered according to the best-in-the-average scanning order presented in Section 2.1.3.

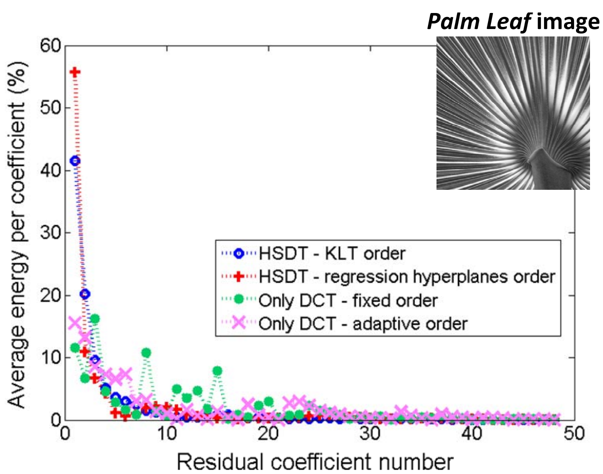

Fig. 5. Residual coefficients average energy concentration for the *Palm leaf* image.

3. RESULTS

We evaluate the results of our HSDT by using two methods: comparing transform coding gain (TCG), and implementing a simple image codec.

3.1. Transform coding gain

The transform coding gain (TCG) is a measure developed for comparing energy concentration performance of various transforms. It is defined as the ratio between the arithmetic and geometric means of the coefficients energy, and gives an approximation of the coding gain of a transform, when followed by an appropriate entropy coder. As expected, the coding gain of the HSDT was significantly superior to the baseline (hierarchical DCT). For example, for the *Palm Leaf* image in the right upper corner of Fig. 5, the HSDT outperforms the DCT by 4.1 dB. For the other 16 images on the Kodak database, the HSDT TCG gains over DCT vary from 0.9 dB to 5.2 dB, with an average of 2.2 dB.

Besides providing high energy concentration in few coefficients, it is desirable that the transform coefficients be ordered in a "decreasing in the average" order. The average percentage energy per coefficient is shown in Fig. 5 for the 48 residual coefficients of the *Palm leaf* image. It can be seen that approximately 55% of the energy is concentrated only in the first residual coefficient. Results for other images presenting different contents also shown that the HSDT outperforms the DCT providing both smoother decay curves and higher energy concentrations in the initial residual coefficients.

3.2. A simple codec

The structure of the HSDT does not easily fit into that of traditional codecs, and developing a full blown codec that matches the level of efficiency and tuning of existing commercial codecs would be too time consuming. Yet, it would be important to evaluate performance on a full codec, even given the caveats mentioned above. Thus, as a proof of concept, a simple image codec was implemented to evaluate the potential gain from the HSDT. At each stage, the transform coefficients are quantized by a uniform scalar quantizer with a deadzone. The quantizer is not tuned to the application, and the quantization step (QP) is pre-defined and used to code all residual coefficients from blocks in all decomposition levels. The scanning of the quantized coefficients can be done sequentially or progressively and the entropy coding stage employs an arithmetic coder. The coder statistical model for the residual coefficients was not optimized, and starts with an uniform distribution for each decomposition level. This initial (uniform) statistical model is then updated only on-the-fly after each symbol encoding/decoding. After all decomposition levels are entropy coded, the final bitstream is produced, written in the coded file and the coding process is completed. As the main purpose of the full image codec is to evaluate the HSDT, the entropy coding stage, as well as the quantization stage, were kept as simple as possible.

The distortion versus bit-rate performance comparisons are performed among JPEG, JPEG XR, JPEG 2000 and the HSDT codec using the two basis vector ordering methods described in Section 2.2: regression hyperplanes and KLT. The HSDT codec has more comparable characteristics with JPEG XR than with JPEG and JPEG 2000. Among other comparable features, JPEG XR also employs a block transform and its *frequency mode* enables quality and spatial scalability functions. The Y-PSNR versus bit-rate curves are shown in Fig. 6 for the *Palm leaf* image. Even with the HSDT codec constraints (quantization and entropy coding stages were not optimized), it outperforms JPEG XR by approximately 1dB on average. As expected, JPEG 2000 outperforms all other image codecs. We believe that by extending the HSDT to an overlapping transform (as used in JPEG 2000), similar gains can be expected.

Note in Fig. 7 that, while JPEG XR introduces more visible blocking artifacts, HSDT produces more blurred images for comparable bit-rate scenarios. This is an interesting result obtained for the HSDT because, although it also employs a block transform as the JPEG XR, the blocking artifacts are weakened through the various decomposition levels. It is important to note here that JPEG XR has as an optional second transform lapped stage to reduce blocking artifacts at low bit-rates.

4. CONCLUSIONS

In this paper we proposed a new framework to design adaptive transforms. The framework allows one to exploit intra- and inter-level

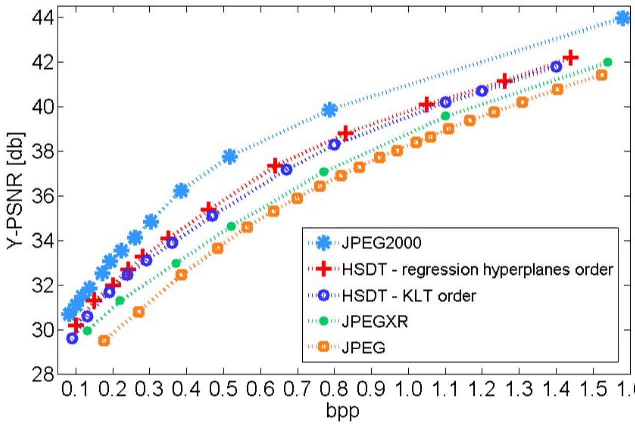

Fig. 6. Y-PSNR versus bit-rate curves for the *Palm leaf* image.

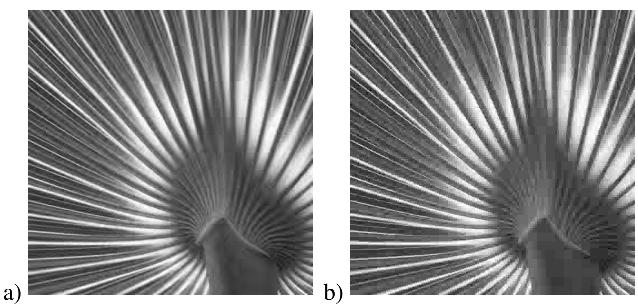

Fig. 7. Reconstructed *Palm leaf* image previously encoded with: *a*) HSDT at 0.25 bpp (achieving Y-PSNR of 32.5 dB) and *b*) JPEG XR at 0.22 bpp (achieving Y-PSNR of 31 dB).

similarities of an image. Yet it avoids overcomplete or other dimensionality expanding approaches: at each stage of each block, the resulting transform forms an orthornormal basis.

Although we believe most of the gain is still to be explored, initial results already show an average coding gain of 2.2 dB over the underlying transform (the hierarchical DCT). We implemented a simple codec to verify how these gains reflect into a more complete codec, and compared results to standard codecs. Experiments showed that HSDT is comparable with JPEG XR. Furthermore, for images with high directional content, results were up to 3 dB better than the equivalent results with JPEG XR. Recent research [2, 4] has shown that local image similarities can be used to increase the energy compaction of local transforms by making them adaptive to the signal. The HSDT significantly extends the recent research by creating a framework that allows other types of similarities to be exploited.

We have shown the HSDT effectively exploits image similarity, and direction content in images. We believe that even better results will be achieved with the development of new basis functions, and after tuning of the quantization and entropy coding stages.

In summary, this work shows that there is still room for further improvement in image coding exploiting cross-level structural similarities, and creates a framework in which this can be exploited.

5. REFERENCES

[1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 3445–3462, 1993.

[2] C. Lan, J. Xu, G. Shi, and F. Wu, "Exploiting Non-Local Correlation via Signal-Dependent Transform (SDT)," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1298–1308, 2011.

[3] U. Bayazit, "Adaptive Spectral Transform for Wavelet-Based Color Image Compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 7, pp. 983–992, 2011.

[4] C. Yeo, Y. Tan, Z. Li, and S. Raharkja, "Mode-Dependent Transforms for Coding Directional Intra Prediction Residuals," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 4, pp. 545–554, 2012.

[5] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaires," *IEEE Transactions on Signal Processing*, pp. 3397–3415, 1993.

[6] J. L. Starck, E. J. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *Proceedings of the IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 758–763, 2002.

[7] W. Ding, F. Wu, X. Wu, S. Li, and H. Li, "Adaptive directional lifting-based wavelet transform for image coding," *Proceedings of the IEEE Transactions on Image Processing*, pp. 416–427, 2007.

[8] H. Xu, J. Xu, and F. Wu, "Lifting-based directional DCT-like transform for image coding," *Proceedings of the IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1325–1335, 2007.

[9] J. Xu, F. Wu, J. Liang, and W. Zhang, "Directional lapped transforms for image coding," *Proceedings of the Data Compression Conference*, pp. 142–151, 2008.

[10] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Elsevier Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[11] B. Tao and B. W. Dickinson, "Texture recognition and image retrieval using gradient indexing," *Journal of Visual Communication and Image Representation*, vol. 11, no. 3, pp. 327–342, 2000.

[12] A. arkacioglu and F. Yarman-Vural, "Sasi: a generic texture descriptor for image retrieval," *Elsevier Pattern Recognition*, vol. 36, no. 11, pp. 2615–2633, 2003.

[13] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[14] S. He and D. Kirovski, "A novel visual perceptual model with an application to high-fidelity image annotation," *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, pp. 92–97, 2006.

[15] V. Testoni, "Contributions in Image and Video Coding," 2011, Ph.D. Thesis.