HIERARCHICAL TENSOR APPROXIMATION OF MULTIDIMENSIONAL IMAGES

Qing Wu *Tian Xia* *Yizhou Yu*

Department of Computer Science
University of Illinois at Urbana-Champaign

ABSTRACT

Visual data comprises of multi-scale and inhomogeneous signals. In this paper, we exploit these characteristics and develop an adaptive data approximation technique based on a hierarchical tensor-based transformation. In this technique, an original multi-dimensional image is transformed into a hierarchy of signals to expose its multi-scale structures. The signal at each level of the hierarchy is further divided into a number of smaller tensors to expose its spatially inhomogeneous structures. These smaller tensors are further transformed and pruned using a collective tensor approximation technique. Experimental results indicate that our technique can achieve higher compression ratios than existing functional approximation methods, including wavelet transforms, wavelet packet transforms and single-level tensor approximation.

***Index Terms*—** Multilinear models, image compression, multi-scale analysis, adaptive bases, tensor ensemble approximation

1. INTRODUCTION

With advances in imaging technologies—-such as CCD, laser, magnetic resonance, and diffusion tensor—-visual data of multiple dimensions have been produced at an unprecedented rate and scale. These new technologies bring new challenges to existing multidimensional image compression techniques.

Visual data exhibit two important intertwined characteristics. First, they comprise of signals at many different frequencies. Second, these signals have spatially inhomogeneous magnitudes. Existing techniques, such as wavelet transforms, have successfully exploited both of the aforementioned characteristics to achieve a transformation that is local in both frequency and space domains. As a result of such transformation, the inherent structures of the original signal become better exposed to compression. One important aspect of such inherent structures is that even though the original signal appears inhomogeneous, its elementary components at different frequencies or local regions may exhibit similar patterns. Further operations performed on these components can remove redundancy and achieve compact representation. There are at least two possible operations we can perform. First, components with similar patterns can be grouped together and compressed collectively in a lower dimensional space to remove redundancy. Second, a component can be simply pruned if its magnitude is negligible. Such processing gives rise to significantly reduced size and dimensionality of a dataset and overall, reduced representational complexity.

Multilinear models based on tensor approximation have received much attention recently. They are capable of generating a more compact representation of multi-dimensional data than traditional dimensionality reduction methods. In this paper, we exploit the aforementioned characteristics of visual data and develop a compact representation technique based on a hierarchical tensor-based transformation. In this technique, an original multi-dimensional dataset is transformed into a hierarchy of signals to expose its multi-scale structures. The signal at each level of the hierarchy is further divided into a number of tensors with smaller spatial support to expose its spatially inhomogeneous structures. These smaller tensors are further transformed and pruned using a tensor approximation technique to achieve a highly compact representation. Our hierarchical tensor approximation can achieve far higher quality than wavelet transforms at large compression ratios. In comparison to a traditional multiresolution analysis which simply projects signals at various different resolutions onto a prescribed basis which was obtained without any specific knowledge of the data, our hierarchical approximation actually adopts bases specifically tailored for the characteristics of the data currently being approximated. We have successfully applied our new hierarchical tensor approximation to both medical and scientific multidimensional images.

2. BACKGROUND AND RELATED WORK

A real Nth-order tensor $\mathcal{A} \in \Re^{n_1 \times n_2 \times \ldots \times n_N}$, can be considered as an element of a composite vector space, $R^{n_1} \otimes R^{n_2} \otimes \cdots \otimes R^{n_N}$, where we call each R^{n_i} an elementary vector space, and $\otimes$ denotes the Kronecker product of vector spaces. The dimensionality of the i-th elementary vector space is n_i. Let us first review basic tensor approximation techniques, including rank-r approximation and rank-$(r_1, r_2, ..., r_N)$ approximation.

A rank-r approximation of $\mathcal{A}$ is formulated as

$$\hat{\mathcal{A}} = \sum_{j=1}^{r} b_j \times_1 \mathbf{u}_j^{(1)} \times_2 \mathbf{u}_j^{(2)} \times \cdots \times_N \mathbf{u}_j^{(N)}, \quad (1)$$

where b_j is a scalar coefficient, each $\mathbf{u}_j^{(i)}$ is simply a column vector of length n_i, and $\times_k$ represents k-mode product of a tensor by a matrix [1]. The column vectors, $\{\mathbf{u}_j^{(i)}\}_{j=1}^{r}$, are not necessarily orthogonal to each other. An efficient algorithm for rank-r approximation can be found in [1]. When r is small, the scalar coefficients along with their associated basis vectors give rise to a compact representation of the original tensor.

A rank-$(r_1, r_2, ..., r_N)$ approximation of $\mathcal{A}$ is formulated as

$$\tilde{\mathcal{A}} = \mathcal{B} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \cdots \times_N \mathbf{U}^{(N)}, \quad (2)$$

where each basis matrix $\mathbf{U}^{(i)} \in \Re^{n_i \times r_i}$, and the core tensor $\mathcal{B} \in \Re^{r_1 \times r_2 \times \cdots \times r_N}$. The column vectors of each $\mathbf{U}^{(i)}$ are orthonormal to each other. Once the basis matrices are known, $\mathcal{B} = \mathcal{A} \times_1 {\mathbf{U}^{(1)}}^T \times_2 {\mathbf{U}^{(2)}}^T \times \cdots \times_N {\mathbf{U}^{(N)}}^T$. When $r_1, r_2, \ldots, r_N$ are sufficiently small, the core tensor and the basis matrices together give rise

[1] The k-mode product of a tensor $\mathcal{A}$ by a matrix $\mathbf{U} \in^{J_k \times n_k}$, denoted by $\mathcal{A} \times_k \mathbf{U}$, is defined as a tensor with entries: $(\mathcal{A} \times_k \mathbf{U})_{i_1 \ldots i_{k-1} j_k i_{k+1} \ldots i_N} = \sum_{i_k} a_{i_1 \ldots i_N} u_{j_k i_k}$.

to a compact representation. The Alternative Least Square (ALS) algorithm was used in [2, 3] to solve the optimal basis matrices given their reduced ranks. In each iteration, ALS optimizes only one of the basis matrices, while keeping others fixed. It has been demonstrated in [4] that rank-$(r_1, r_2, \cdots, r_N)$ tensor approximation can achieve smaller Root Mean Squared Errors (RMSE) than Principal Component Analysis (PCA) given the same compression ratios.

Note that existing tensor approximation methods consider the input data as a single-resolution multi-dimensional array without exploiting its multi-scale structures. On the other hand, wavelet analysis is inherently a multi-scale analysis tool and has been frequently applied to multidimensional image compression [5, 6]. Wavelet packet techniques have also been developed to recursively decompose both the low-frequency and high-frequency components at each scale [7, 8]. When wavelets are applied to multi-dimensional signals, the bases are typically formed as tensor products of the one-dimensional bases. There has been work on developing more powerful oriented wavelet bases [9]. However, such work has been primarily focused on two-dimensional images while this paper focuses on images with a dimensionality higher than two.

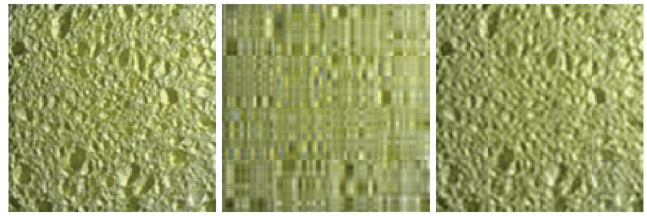
3. HIERARCHICAL TENSOR APPROXIMATION

Given a collection of multi-dimensional images with the same size and dimensionality, our hierarchical approximation algorithm produces a compact hierarchical representation based on tensor approximation by removing the redundancies among different images as well as within each one. At each level of the hierarchy, our representation keeps an incomplete approximation of a list of (subdivided) tensors. At an intermediate level, these subdivided tensors represent the residual errors of the accumulated approximation at higher levels. Once the tensors representing the residual errors are passed to the next lower level, each of them is further spatially subdivided into up to 2^N smaller tensors where N is the order of the original tensors. In the following, we first discuss tensor approximation at each level, and then introduce hierarchical transformation and approximation.

3.1. Tensor Ensemble Approximation

In many situations, we need to simultaneously approximate an ensemble of tensors, and most often, these tensors have strong correlations. For example, a multi-dimensional array of color values or velocity vectors gives rise to three scalar tensors for the three color channels or three components of the vectors. As we know, color response curves have much overlap with each other and velocity components need to satisfy certain physics-based equations. As a result, these scalar tensors have strong correlations with each other. As will be discussed in the next section, we also subdivide a large tensor into smaller ones and approximate them collectively because these subdivided tensors have local spatial support and may share similar basis matrices among each other.

Suppose the list of tensors at level l is $\mathcal{A}_1^l, \mathcal{A}_2^l, \cdots, \mathcal{A}_{m_l}^l$, where m_l is the number of tensors and $\mathcal{A}_i^l \in \Re^{n_1^l \times n_2^l \times \ldots \times n_N^l}$, and we look for a rank-$(r_1^l, r_2^l, ..., r_N^l)$ approximation of each $\mathcal{A}_i^l$, which is denoted as $\tilde{\mathcal{A}}_i^l$. Because of correlations and redundancies among this list of tensors, approximating each of them separately is suboptimal. We move one step further and approximate all these tensors collectively. To achieve this goal, we organize these m_l N-th order tensors into a $(N+1)$-th order tensor $\mathcal{G}^l \in \Re^{n_1^l \times n_2^l \times \ldots \times n_N^l \times m_l}$, and obtain a rank-$(r_1^l, r_2^l, ..., r_N^l, r_{N+1}^l)$ tensor $\tilde{\mathcal{G}}^l$ as its approximation using the ALS algorithm. Note that $r_{N+1}^l \leq m_l$. This approximation is compactly represented using $N+1$ basis matrices, $\mathbf{U}^{(1)}, \cdots, \mathbf{U}^{(N)}, \mathbf{U}^{(N+1)}$, and a core tensor $\mathcal{H}$. That is,



(a) Original (b) Individual PSNR 20.13 (c) Ensemble PSNR 26.17

Fig. 1. A comparison of a reconstructed SPONGE texture from both ensemble and individual tensor approximations. (a) Original image, (b) a reconstructed image from individual approximation, (c) a reconstructed image from our ensemble approximation. (b)&(c) share the same compression rate which is 87.5%.

$$\tilde{\mathcal{G}}^l = \mathcal{H} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times \cdots \times_N \mathbf{U}^{(N)} \times_{(N+1)} \mathbf{U}^{(N+1)}, \quad (3)$$

where $\mathbf{U}^{(1)} \in \Re^{n_1^l \times r_1^l}, \cdots, \mathbf{U}^{(N)} \in \Re^{n_N^l \times r_N^l}$ and $\mathbf{U}^{(N+1)} \in \Re^{m_l \times r_{N+1}^l}$ and $\mathcal{H} \in \Re^{r_1^l \times r_2^l \times \cdots \times r_N^l \times r_{N+1}^l}$. When necessary, it is actually quite convenient to extract the core tensor $\mathcal{B}_i^l$ of each N-th order subtensor $\tilde{\mathcal{A}}_i^l$ out of this ensemble representation. Let $\mathbf{u}_i^{(N+1)}$ be the vector representing the transposed i-th row of $\mathbf{U}^{(N+1)}$. Then,

$$\mathcal{B}_i^l = \mathcal{H} \times_{(N+1)} \mathbf{u}_i^{(N+1)^T}. \quad (4)$$

We have compared our tensor ensemble approximation against individual tensor approximation. Fig. 1 shows one of such comparisons on texture images. In this example, the original texture image is partitioned into 16 blocks each of which has three color channels. Our ensemble approximation models the data as a list of 48 subtensors, approximates them collectively, and achieves a peak signal-to-noise ratio (PSNR) of 26.17 at 87.5% compression rate. On the other hand, individual approximation needs to store a distinct set of basis matrices for each color channel and each subtensor even when these bases are similar, and can only achieve a PSNR of 20.13 at the same compression rate.

3.2. Hierarchical Transformation and Approximation

In this section, let us first introduce a lossless hierarchical transformation of multi-dimensional matrices, or tensors. This transformation decomposes the original data into multiple levels and removes the redundancy at each level by exploiting the similarity among different spatial regions. To exploit spatial inhomogeneity of the original data, further lossy approximation (quantization and pruning) is performed on the resulting multilevel data. These two steps together give rise to a very compact representation.

We perform the lossless hierarchical transformation from top to bottom. The original input tensors are placed at the top level, which is also the first level. At each level l, there is an initial list of N-th order tensors, $\mathcal{A}_1^l, \mathcal{A}_2^l, \cdots, \mathcal{A}_{m_l}^l$. We exploit similarity among these tensors and remove redundancy by performing tensor ensemble approximation as discussed in the previous section. We only perform an incomplete approximation in the sense that the ranks of the truncated basis matrices are set to be smaller than necessary and the residual error is not necessarily reduced to the desired level. From this incomplete approximation, we can obtain an approximated version, $\tilde{\mathcal{A}}_i^l$, of each tensor $\mathcal{A}_i^l$. A residual tensor, $\mathcal{E}_i^l = \mathcal{A}_i^l - \tilde{\mathcal{A}}_i^l$, is subsequently defined for each tensor. Each residual tensor is then

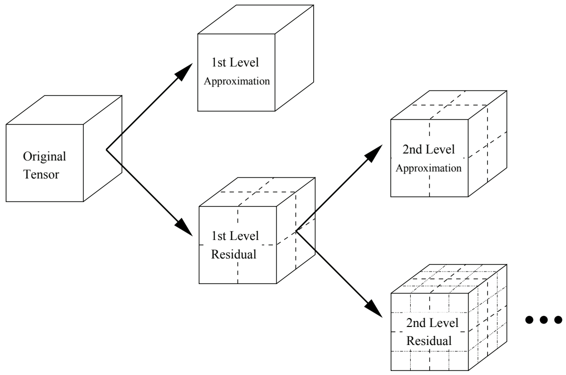


Fig. 2. In our hierarchical tensor transformation, an original tensor is represented as the summation of incomplete tensor approximations at multiple levels. The tensors at each level are subdivided residual tensors passed from the higher level.

subdivided into up to 2^N smaller tensors by dividing the dimensionality of each elementary vector space in half unless the dimensionality of an elementary vector space has already been reduced to one. These subdivided residual tensors are passed to the next lower level in the hierarchy for further approximation. Therefore, there will be at most $2^N m_l$ smaller tensors in the next lower level. And the approximation process repeats on these subdivided tensors. Such tensor subdivision and approximation can be repeated until the dimensionality of all elementary vector spaces has been reduced to one. Note that at the bottom level, all the tensors have only one scalar element, and further subdivision and approximation have become unnecessary. Even though we perform incomplete tensor approximation at every level except the bottom one, it can be easily verified that the original tensors at the top level can be faithfully reconstructed by first reconstructing the (residual) tensors at each level from their corresponding core tensors and basis matrices, and then accumulating the tensors at all levels together (Fig. 2).

To achieve a more compact representation, we need to perform further lossy approximation of the original data from the above hierarchical transformation. We achieve this goal by performing quantization on the core tensor coefficients followed by a tensor pruning step. Coefficients with a magnitude smaller than the quantization step are set to zero. The elements of the basis matrices are also quantized. In our experiments, we always use 8 bits per element for the basis matrices, and 8-20 bits per coefficient for the core tensors. After quantization, we further perform a pruning step on core tensors by introducing a separate pruning threshold which can simply be zero. For each core tensor $\mathcal{B}_i^l$ defined in (4), we compute the summation of squared coefficients. If the summation is less than the pruning threshold, the entire core tensor is eliminated. If the pruning threshold is set to zero, a core tensor is eliminated only when all of its coefficients have been quantized to zero. This tensor pruning step bears resemblance to coefficient pruning in wavelet-based image compression [5, 10]. Since the input data has spatially varying details, the coefficients of the core tensors corresponding to smooth regions of the data are likely to be small. Thus, these core tensors are more likely to be pruned. The tradeoff between the compression ratio and PSNR is achieved by adjusting the tensor pruning threshold. Given a PSNR and a quantization step, we perform a search for the tensor pruning threshold that can achieve the desired PSNR. When it is necessary to code the coefficients of the remaining core tensors, there are many existing techniques, such as arithmetic coding, entropy coding and zero-tree coding [10], from which one can choose.

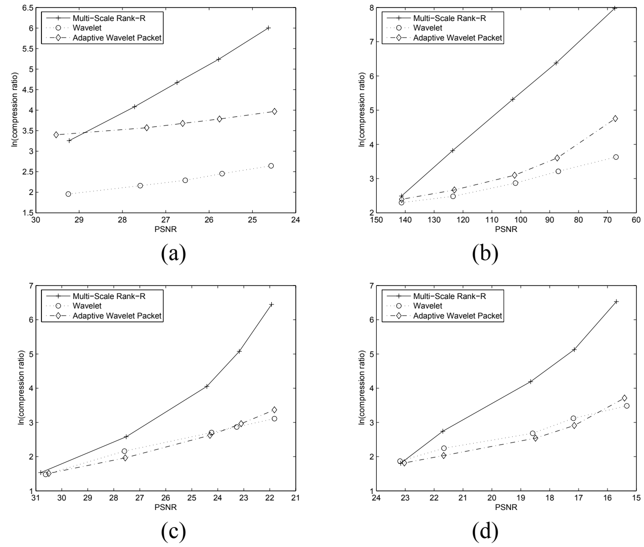


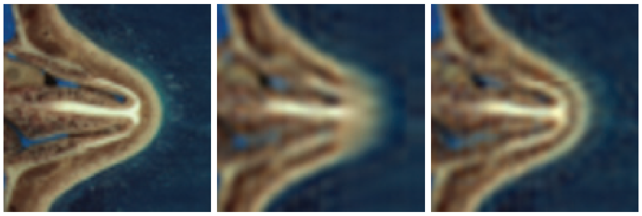
Fig. 3. Comparisons of data compression ratios achieved on four datasets by a bio-orthogonal wavelet transform (dotted), a corresponding wavelet packet transform (dash-dotted), and our hierarchical tensor approximatin (solid). The datasets include (a) a subset of the Visible Human dataset, (b) the velocity field of the 4D time-varying volume dataset, (c) a SPONGE BTF, and (d) a LICHEN BTF.

Meanwhile, in our hierarchical transformation, we still need to determine the reduced ranks of the basis matrices at each level. Since achieving an optimal selection of the reduced ranks at all levels requires expensive run-time optimization, we have designed a fast scheme that can achieve a suboptimal solution. At the first level, this scheme lets the user choose the set of desired ranks, $r_1^1, r_2^1, ..., r_N^1$, for the basis matrices. At subsequent levels, each of the ranks follows a geometric progression. Though being suboptimal, this scheme has been very effective in our experiments.

4. EXPERIMENTS

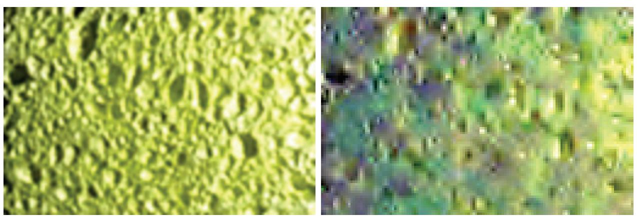
We have conducted experiments on a 4D time-varying scientific dataset, 3D medical images in the Visible Human dataset, and 4D bidirectional texture functions (BTFs). The 4D time-varying dataset is a sequence of simulated 3D velocity fields surrounding five jets. BTFs represent the changing appearance of a rough surface under a large number of combinations of lighting and viewing directions [11]. We have compared our hierarchical tensor approximation against wavelet transforms used in JPEG 2000 [12], wavelet packet transforms presented in [8], and a single-level tensor approximation technique from [4].

Fig. 3 shows comparisons of compression ratios that can be achieved by three techniques over a wide range of PSNR values. In our hierarchical approximation, there are 4-5 levels in the hierarchy. At the five data points on each curve corresponding to our compression scheme, the reduced ranks used for the top-level approximation are respectively $1/2, 1/4, 1/8, 1/16$, and $1/32$ of the original rank, and the common ratio between the ranks at two adjacent levels is always 0.5. Except for very few large PSNR values, our hierarchical tensor approximation achieved the highest compression ratios. And in most cases, the compression ratio it can achieve is at least one order of magnitude larger than that achieved by the wavelet trans-

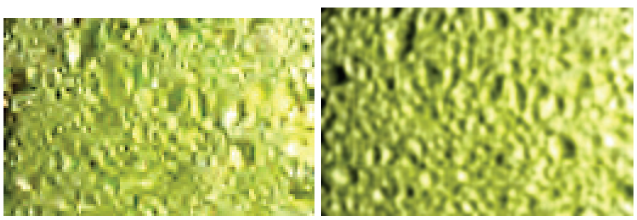


(a) Original (b) Single-level PSNR 29.89 (c) Hierarchical PSNR 31.05

Fig. 4. A comparison of the reconstructed Visible Human dataset from the single-level tensor approximation in [4] and our hierarchical tensor approximation. (a) A magnified view of a cross section of the nose region. (b) A reconstructed image from the single-level tensor approximation. (c) A reconstructed image from our hierarchical tensor approximation. (b)&(c) share the same compression ratio which is 15.7.



(a) Original (b) Wavelet, PSNR=16.77



(c) Wavelet Packet, PSNR=20.54 (d) Hierarchical, PSNR=25.21

Fig. 5. A comparison of reconstructed BTF images from a bio-orthogonal wavelet transform, an adaptive wavelet packet transform, and our hierarchical tensor-based representation. (a) An original BTF image. (b)-(d) Reconstructed images from the wavelet transform, the wavelet packet transform, and our hierarchical tensor approximation, respectively. The compression ratio for (b)-(d) is 55.

form. Meanwhile, our hierarchical technique also maintains a fairly significant improvement over the state-of-the-art single-level tensor approximation technique in [4]. In fact, on the same four datasets used in Fig. 3, the compression ratios achieved by our technique are respectively 41.0%, 52.9%, 93.9% and 121.8% higher than those achieved by the single-level tensor approximation under the same PSNR values. Note that in our hierarchical tensor approximation, we estimate the compression ratios according to the storage required by all remaining tensors and basis matrices in the hierarchy.

Fig. 4 shows a visual comparison between the single-level and hierarchical schemes on a local region from the Visible Human dataset. The original data has an extruding feature which the single-level method has failed to approximate well while the reconstruction from our hierarchical method still preserves the important details. Fig. 5 shows a visual comparison between wavelet transform, wavelet packet transform and our multi-level tensor scheme. We can also conclude that our technique can effectively preserve the fine details of the BTF and achieve the best visual quality.

5. CONCLUSIONS

In this paper, we developed a compact data approximation technique based on a hierarchical tensor-based transformation. Resembling a multiresolution analysis such as wavelet transform, our approximation represents significant and typically low frequency components at higher levels of the hierarchy and less important (high frequency) components at lower levels. Because high frequency components have smaller spatial support, they can be approximated using shorter basis vectors. That is one of the reasons we keep subdividing the residual tensors from level to level and use increasingly shorter basis matrices to approximate them. Shorter basis matrices impose less overhead on compression ratio.

More importantly, traditional multiresolution analysis simply applies scaled versions of a prescribed basis to signals at various different resolutions while our hierarchical approximation extracts basis matrices specifically tailored for the data being approximated. Therefore, our method is much better at removing redundancies in a specific dataset. In practice, we have found that the gained efficiency of our method in approximating data surpasses the storage overhead for the adaptively extracted basis matrices.

References

[1] T. Zhang and G.H. Golub, "Rank-one approximation to high order tensors," *SIAM J. Matrix Analysis and Applications*, vol. 23, no. 2, pp. 534–550, 2001.

[2] P. Kroonenberg and J. de Leeuw, "Principal component analysis of three-mode data by means of alternating least squares algorithms," *Psychometrika*, vol. 45, pp. 324–1342, 1980.

[3] L. De Lathauwer, B. de Moor, and J. Vandewalle, "On the best rank-1 and rank-$(R_1, R_2, ..., R_n)$ approximation of higher-order tensors," *SIAM J. Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[4] H. Wang, Q. Wu, L. Shi, Y. Yu, and N. Ahuja, "Out-of-core tensor approximation of multi-dimensional matrices of visual data," *ACM Transactions on Graphics*, vol. 24, no. 3, 2005.

[5] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, 1992.

[6] S. Muraki, "Volume data and wavelet transforms," *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 50–56, 1993.

[7] F.G. Meyer, A.Z. Averbuch, and J.-O. Stromberg, "Fast adaptive wavelet packet image compression," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 792–800, 2000.

[8] A. Jensen and A. la Cour-Harbo, *Ripples in Mathematics: The Discrete Wavelet Transform*, Springer, 2001.

[9] E.J. Candès and D.L. Donoho, *Curvelets – a surprisingly effective nonadaptive representation for objects with edges*, Vanderbilt University Press, 1999.

[10] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[11] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real world surfaces," *ACM Transactions on Graphics*, vol. 18, no. 1, pp. 1–34, 1999.

[12] M. Boliek, C. Christopoulos, and E. Majani (editors), "Jpeg 2000 image coding system (jpeg 2000 part i final committee draft version 1.0)," ISO/IEC FCD15444-1, http://www.jpeg.org/jpeg2000/CDs15444.html, March 2000.