Optimal entropy encoders for mining multiply resolved data

R. A. DeVore, L. S. Johnson, C. Pan & R. C. Sharpley
Industrial Mathematics Institute, University of South Carolina, USA

Abstract

A prototype client-server implementation of image analysis and compression is described which is based on the recently developed theory of Cohen, Dahmen, DeVore, and Daubechies for optimal entropy data encoders. The class of algorithms resulting from this theory was developed for the analysis and synthesis of data and yields optimal (in an information-theoretic sense), progressive, universal encoders for purposes of compression, storage, and transmission of data which can be developed into a multi-resolution framework. Such data include photographic and sensor images, digital terrain maps, and multidimensional scientific data generated by computational simulators.

Two versions of the tree encoder have been implemented with a common client interface in order to demonstrate the applicability to diverse model problems. The first version processes the raw data in real time and is designed for relatively small data sets on single machines. The second version is designed as a network application to efficiently navigate archived data residing on remote databanks. Data structures which contain summary statistics, indexing and threading information generated by the analysis stage of the algorithm are utilized to navigate the data at adaptively-determined levels of resolution based upon the local depth of detail requested by the client application.

1 Introduction

There are diverse classes of data sets which must be efficiently analyzed, registered against databases, denoised, mined for information, and visually navigated, but which are too large to process effectively through a remote connection or interactively on a local workstation or PC. This paper describes our current work in scalable algorithm development and implementation for processing (i) large images (such as those that arise in medical and satellite imagery), (ii) data sets generated by large scale scientific simulators, and (iii) GIS digital terrain elevation maps for remote access. Our processing algorithms include several features which are important in data processing and analysis, such as

- progressive transmission of information from low resolution lossy to high resolution lossless data reconstruction,
- user-determined regions for local refinement,
- scalable algorithms suitable for both single and parallel processing,
- and multiple resolution display.

The multiresolution structure inherent in these wavelet-based algorithms provides a hierarchical spatial/frequency representation that is well suited for efficient operations throughout all scales of the data. At each node of this hierarchical representation, summary statistics may be used to provide information about features and information that may be desirable for further analysis. In Section 2 we describe the multiresolution framework provide by wavelet representations and client-server models which utilize these structures in order to implement optimal entropy progressive approximations of data which are useful for the analysis of extremely large stored data sets or remote transmission. A prototype implementation of these ideas is described in Section 3. Conclusions and additional applications are briefly discussed in Section 4.

2 Client-server models for local progressive transmission

Following idea from [1], several algorithms are developed in [2] for encoding wavelet decompositions of signals and higher dimensional data which are logically rectangular arrays. Generalizations of the multiresolution formulation to irregularly spaced data are given in [3,6]. Wavelet transforms applied to raw data are used for a variety of data processing tasks, including data assimilation, denoising, feature extraction, compression, and enhancements. The data is transformed to the wavelet domain by filtering the data with a discrete wavelet transform (DWT). Generic data processing tasks are performed by doing simple, yet powerful, operations on the transformed arrays and then transforming back using an inversion process (IDWT) (see Figure 1).

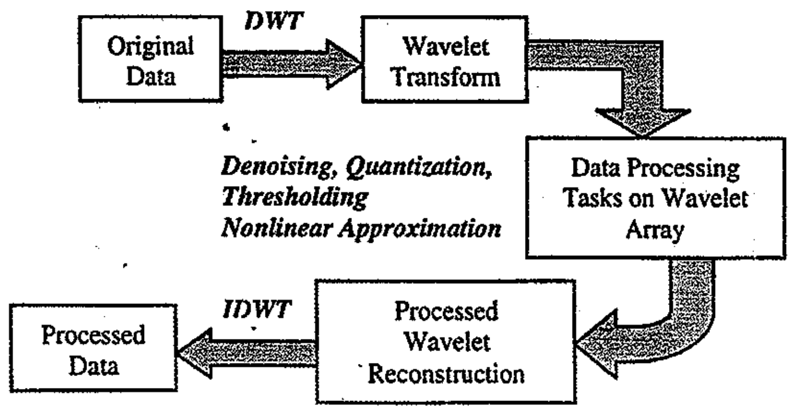


Figure 1: Generic schematic of end-to-end wavelet based data processing.

2.1 Overview of multiresolution decompositions

Primary operations of multiresolution analysis are projection filters that transform high-resolution data into coarser representations. These filters permit very large datasets to be represented at scales that are well suited for display and analysis, given the reference position of the viewer and the zoom factor selected. The wavelet transform operates on the same scale of high resolution data as the projection filters, but provides the detail which must be blended with the coarser representation in order to retrieve the original high resolution representation (see Figure 2). These complementary operations are then performed recursively on each scale (from highest resolution to the coarsest) in order to provide a multiresolution decomposition of the data into a doubly indexed collection of spatial and frequency information – the **discrete wavelet transform**. This latter stage is called the **decomposition** process while the filtering procedure to recover the original data is called the **reconstruction** process.

An important feature of these filters is their finite lengths (supports) which guarantees that each computed transform coefficient reflects only local information at the current scale. As shown in Figure 2, the filtering process

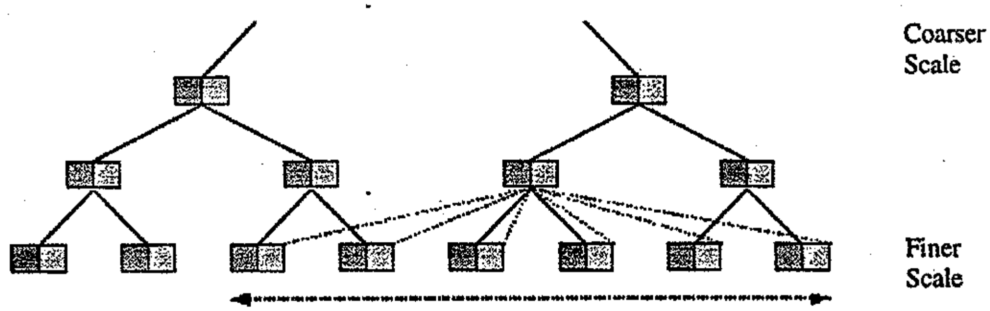


Figure 2: Sample hierarchical structure in 1D and the local support contribution (dotted lines) of fine scale data to coarser resolution of wavelet (dark) and scale (light) coefficients.

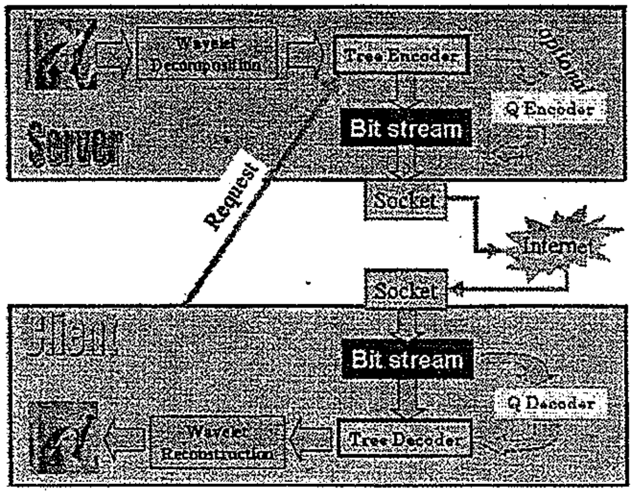


Figure 3: Schematic for client-server implementation.

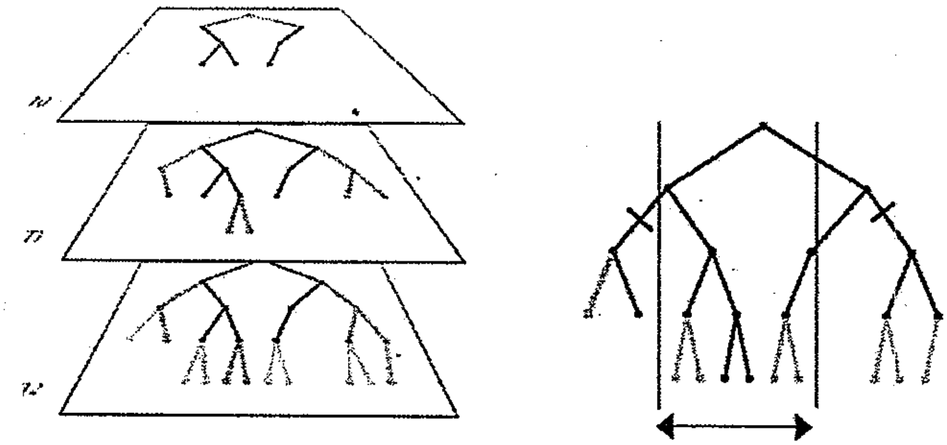
lends itself to a natural tree structure, where, in general, the transform coefficients may depend not only on direct descendants, but also upon a predetermined set of relatives. The inverse wavelet transform has an analogous filter. Moreover, by a simple geometric summing argument, the hierarchical representation shows that the complete decomposition process only requires a number of numerical operations on the order of the size of the original data, which then ensures that the multiresolution process is scalable with both computational advances and anticipated increases of data size

Typical data processing operations of compression, denoising, and feature extraction may then be applied to these wavelet transformed arrays, for example, by thresholding, weighting, and clustering the coefficients in various ways before taking the inverse transform. This is also a natural place to decouple the algorithm into server and client processes, since for large classes of typical images only a few significant coefficients remain at this intermediate stage which must be "transmitted". In Section 3 we will describe implementation of the client-server model, but for now we may assume that the client executes functions for the reconstruction and display of approximate images, while the server handles client requests such as providing (or transmitting) wavelet coefficients consistent with those requests (see Figure 3).

2.2 Progressive transmission

The inverse discrete wavelet transform (**IDWT**) performed by the client reconstructs the approximation of the original image from the wavelet coefficients (and miscellaneous other data). If this process is done on a different machine or by a different application than the one that performed the wavelet decomposition, then the coefficients (which have typically been quantized and entropy-encoded) must be transmitted to the reconstructive process based upon the client request.

The server could simply send all of the coefficients in some pre-set order chosen by the application programmer. In this case, the client does not begin the reconstruction process until all of the data has been received. This is adequate if



(a) (b)

Figure 4: Tree structure in 1-D indicating (a) bit plane layers T_j generated by minimal trees which contain the "significant nodes" obtained by thresholding with specified metrics, and (b) tree pruning operation determined by user-selected region.

2.4 Localization

A further enhancement in efficiency of transmission is to provide the capability for the client to specify regions for local refinement of the resolution, i.e. only transmit those coefficients which will provide finer detail to the selected region (Figure 4b). The local supports of the wavelet filters permit basic tree operations to mark a subtree of coefficients to be transmitted. For each bit layer of a user-selected region, the server is able to maintain the progressive updates.

If the server is constructing the bit layers on demand, then this added functionality of localization is easily implemented by elementary tree pruning operations. If the server model is based upon reading the bit plane layers of the decomposed image from a pre-processed file of bits, then additional information is needed in order to identify which bits need to be sent to fulfill the client's request and where they are located in the file. This indexing data comes in the form of several tree-based (or node-based) data structures. The server reads the indices and transmits the requested bits according to the index information.

2.5 Mathematical Foundation of Optimal Encoders

The performance of encoders is measured by the distortion versus the bit rate. Optimal bit rate performance in encoding a deterministic class $\boldsymbol{K}$ of functions in a given distortion metric is determined by the Kolmogorov entropy of that class (see [2]). The Kolmogorov entropy number N_ε for a class $\boldsymbol{K}$ and a given $\varepsilon > 0$ is the smallest number of balls of radius ε which cover $\boldsymbol{K}$. The $\log_2(N_\varepsilon)$ is the fewest number of bits which can encode each $f \in \boldsymbol{K}$ to a distortion $\leq \varepsilon$. While the Kolmogorov encoding gives a benchmark for the performance of encoders, it cannot be practically implemented. An encoder is given in [2] which allows for a choice of metric among the L^p spaces (p-power integrable) and is universal in that it applies to any function in L^p. The encoding is shown to be optimal (up to multiplicative constants) for a large collection of smoothness classes K (unit balls of Besov spaces).

The method developed in [2] employs the usual wavelet thresholding methods for nonlinear approximation [4], but avoids costly encoding of the significant coefficient positions by instead transmitting the minimal tree containing the significant nodes and the optimal number of bits for each bit layer of the corresponding wavelet coefficients in order to provide a progressive optimal encoder. Through a rigorous counting argument coupled with nonlinear approximation error estimates, it is shown that this tree-encoding method is optimal (in the sense denoted above) for a large class of smoothness spaces.

Since the constants in the asymptotic estimates degrade as p increases, when approximation is required in the maximum deviation norm (that is, $p = \infty$), new techniques are required. In this case, it is shown in [2] that the thresholding procedure must address all levels. This results from the fact that, locally, the significant wavelets from many levels have supports which may overlap which then adversely affects the error. To remedy this theoretically, in a coarse-to-fine procedure the threshold test for the significance of each node must be replaced by a test at each node to determine whether the weighted sum of coefficients over *any* possible subtree (using the node in question as the root) exceeds a certain threshold. The resulting computational cost however is prohibitive and does not lead to scalable algorithms. If instead at the previous stage of processing the data from fine to coarse scale to obtain wavelet coefficients, one stores (in appropriate structures at each node) the information at what layer future subtrees are to be grown in the progressive transmission, then this may be used to determine the current approximate to maintain control of the maximum deviation error. In addition, the algorithm will maintain the universality of encoding in terms of the metrics used.

This bottom-up (fine-coarse) hierarchical initial pass of the data on the server as coefficients are generated guarantees O(N) operation efficiency, and allows the structures at each node to be populated with appropriate information for later adaptive local processing by client applications. More importantly, the data may be decomposed and stored in a file, indexed according to the above strategy, so that it may be quickly accessed by clients for local resolution, independent of size of the total data set and depending only on the size of the subregion marked for further resolution. This same design is used to provide an optimal-entropy, universal encoding method for storage of images and similar data.

3 Implementations of Client-Server

The constructive theory provided in [2] and outlined in Section 2 provides the basic algorithms leading to optimal and universal entropy encoders for progressive transmission of data. The practical implementation of these ideas is naturally amenable to two client-server models. In our current implementations for image processing, the client performs the same functions for both models.

3.1 Client model

The client needs to be able to handle standard tasks such as establishing connection with the server, coordinating the transmission of the data, reconstructing the image, and displaying the results. Additionally, the client allows the user to interrupt the transmission in order to specify regions for local refinement, and allows the user to specify desired levels of detail, which are typically based on display capabilities.

As part of the task of reconstructing the image, the client must maintain information about which parts (updates for coefficient bits, subtrees, new coefficient and sign bits) and how much of the data has been previously received in order to handle the progressive nature of the transmitted data. This may double the memory requirements of the server as compared to the requirements of simply receiving the data all at once.

Once progressive transmission is achieved, an obvious next step is to allow the client to interactively interrupt the process in order to focus on a particular region (or regions) of the image. The process would continue, but only the bits needed to refine the selected region(s) would be sent to provide user desired detail. This sort of activity has been termed "burning in" or "tunneling in".

3.2 Server models and corresponding structures

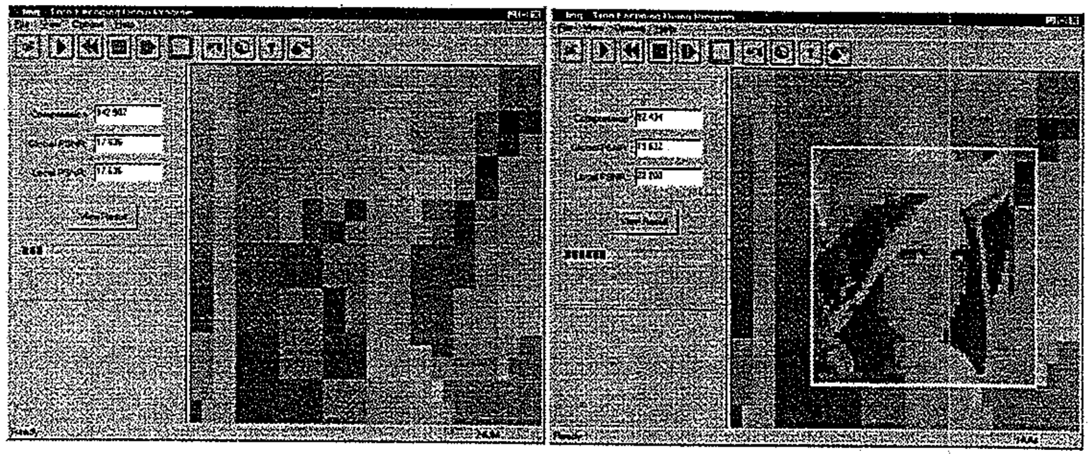
The server models in our two implementations are based upon either

(i) providing in-core processing
(ii) the ability to access index files to wavelet-based decompositions of data that have been previously generated.

The server must also track which data has been previously sent, unless it is simply serving a pre-processed bit file and doesn't allow the client to specify regions for local refinement. In that latter case, the server has no data memory requirements at all, save for the ability to remember its place in the file.

For a server to handle a pre-processed bit file and localization, it must be able to jump to the bit (in the file) corresponding to an arbitrary node in the current bit layer and then be able to construct the string of bits identifying the children to be "grown" from a previous layer after a localization request is made. This requires a collection of structures to dynamically build an index file based upon client requests for specified regions of data.

The structures required for these operations increase the disk space requirements of the server several-fold, but for large data, the corresponding increase in speed should justify the increase.



(a) (b)

Figure 5: Client interface illustrating (a) nonlinear transmission of optimal information and (b) user controlled local refinement in the progressive updates.

3.3 Sample implementation for image processing

To illustrate features of the progressive tree encoder, we have implemented a graphical user interface for the client which handles tasks such as network connections, loading images, setting parameters for encoding, and determining statistics such as compression rates and peak signal to noise ratio. VCR controls are also available to observe the nonlinear progressive approximations as detail is added (see Figure 5a). Specification of a subregion of the image by a click-and-drag mouse operation permits a local refinement of the selected area in future updates as the transmission proceeds (Figure 5b). Additional advantages of the multiresolution representation are that multiple views may be represented for scale up purposes using the coarsening operators while the detail operators may be used for simultaneously displaying extracted features such as edges, shapes, and more subtle features.

4 Conclusions and Future Directions

A class of multi-resolution based algorithms has been described which permit the analysis and navigation of large data sets through a client-server implementation. The basic design provides for remote transmission of medical images for diagnoses, localized high resolution processing for digital terrain elevation maps, and extensions to higher dimensional data. Applied to mammogram imagery for example, one may locally use an end to end wavelet processing system to transform the image to the wavelet domain, apply feature identification algorithms to automatically detect highly correlated data at appropriate scales and mark those regions suspected to contain pre-cancerous microcalcifications, and encode the data using the optimal entropy tree encoding algorithms described here. An expert radiologist may use the client application to request low resolution images marked up for the suspicious areas, which may then be displayed to full resolution. Similar applications are being investigated for military applications combining digital terrain maps with satellite and sensor imagery through automatic registration methods enabled by wavelet technology. These applications are important for realistic three-dimensional renderings for use in mission rehearsal and planning as well as post battlefield assessment. In this case, uniform approximation is a primary metric and will rely on the methods described in this paper.

Acknowledgements

This work has been supported in part by The Office of Naval Research and the Army Research Office Contracts #N00014-00-10470 and #DAAG55-98-1-0002 and by the National Science Foundation Grant #NSF KDI 578AO45.

References

[1] Cohen, A. et al., On the Importance of Combining Wavelet-Based Non-Linear Approximation with Coding Strategies, *IEEE Transactions in Information Theory*, submitted.

[2] Cohen, A., Dahmen, W., Daubechies, I. & DeVore, R.A. Tree Approximation and Encoding, *IMI Technical Report* **99:04**, Department of Mathematics, University of South Carolina, Columbia, SC 29208, (1999).

[3] Daubechies, I., Guskov, I., .Schroder, P. & Sweldens W. Wavelets on irregular point sets, *Phil. Trans. R. Soc. Lond. A.* (to appear)

[4] DeVore, R.A. & Lucier, B. Wavelets, *Acta Numerica*, **1** (1991), 1-56.

[5] Mallat, S., *A Wavelet Tour of Signal Processing*, Academic Press, New York, 1998.

[6] Dyn, N. & Ron, A. Radial basis functions approximation: from gridded centers to scattered centers, *Proc. London Math. Soc.*, **71** (1995), 76-108.

[7] Ron, A. et al., *The ideal data representation center*, Knowledge Distributed Intelligence Project. [http://www.waveletidr.org/].

SAMPLE SYMBOLS

N_ε
$\varepsilon > 0$
$\log_2(N_\varepsilon)$
$f \in \boldsymbol{K}$
distortion $\leq \varepsilon$
L^p