Compressed Representations of Molecular Structures

Chandrajit Bajaj

Computational Visualization Center
Department of Computer Sciences & Institute of Computational and Engineering Sciences
ACES 2.324A, 24th & Speedway, University of Texas at Austin, Austin, TX 78712
bajaj@cs.utexas.edu

Julio E. Castrillón-Candás

julio@ices.utexas.edu

Vinay Siddavanahalli

skvinay@cs.utexas.edu

Zaiqing Xu

zaiqing@cs.utexas.edu

Summary

We introduce a novel efficient representation for volumetric molecular structure and functions. This molecular representation allows us to interactively deal with large and small biomolecular datasets produced by increasingly accurate and better imaging techniques and simulations. A hierarchical basis wavelet based compression scheme is derived. Due to the C_0 continuity offered by these wavelets, we retain good accuracy of the molecular surfaces even at high compression ratios. Another important feature of our scheme is fast random access, which allows for fast rendering of large datasets which do not fit into main memory. Theoretical average, worst case access and computation costs for reconstruction are derived. As an extension, we also provide an algorithm for error bounded compression. Such an error bounded representation allows for high compression and fast random access can be used for both visualization and simulations of biomolecular complexes.

1 Introduction

Bio-molecular Complexes form the machinery responsible for most biological processes and are relevant to understanding many diseases such as cancer and metabolic disorders. Today, hybrid experimental approaches, utilizing X-ray crystallography, cryo-electron microscopy (Cryo-EM), and molecular tomographic imaging, together with structural proteomics have led to a rapid increase in the number of three-dimensional (3-D) structures and 3D electron density map data of bio molecules, in the Protein Data Bank (PDB) and at the European Bioinformatics Institute (EBI)). Supercomputers across the world are also now routinely deployed for computer generation of associated properties of these bio-molecules. These associated properties include large volumetric force fields and interaction potentials, including terms derived from volumetric Poisson-Boltzmann electrostatics potentials, or Generalized Born solvation energy estimations.

While molecular visualization software has developed over the years, today few tools if any are capable of interactive visual exploration of bio-molecules and their associated properties. An important challenge in achieving enhanced visualization of large datasets such as bio-molecules and their associated properties, with minimal interactive performance degradation is effective data processing and management, so that one can manipulate the large volumetric data in real time. Many 'large' datasets cannot all fit into the main memory of most desktops. To do large data analysis and visualization efficiently and even interactively, new algorithms need to be developed that process compressed data "in situ", using minimal additional memory space and time. Furthermore, interactive performance is dependent on rapid decompression of subvolumes, especially at random locations within the dataset.

Our approach to enabling real time interactive exploration of molecular maps and associated properties is to visualize directly from compressed representations of data with fast and random local decompression of this data. To achieve this goal, in this paper we introduce a new wavelet compressed volumetric representation of bio-molecules and associated properties, and address the tradeoff of excellent data compression, while maintaining rapid random and local decompression. These wavelet function spaces are custom built for rectilinear volumetric data, and are the span of a hierarchical basis (HB). Our HB wavelet representations of bio-molecules and associated properties preserves biological features at varying resolutions, while achieving a very high degree of compression, and yet allowing for efficient local decompression.

An important solution technique used in conjunction large data visualization algorithms is data compression (Bajaj, Ihm & Park (2000), Bajaj, Ihm, Park & Song (2000), Bajaj et al. (1999*b*), Bajaj, Ihm & Park (2001*b*,*a*), Bajaj et al. (1999*a*)). Compression yields maximal bandwidth for transmission of data from disk to compute servers, and from servers to display clients. Additionally, data compression allows for maximal memory utilization, wherein algorithms work directly from compressed data in-core, and use minimal incremental decompression for processing. Another obvious use of data compression is the archival storage of input volumetric density maps, as well as output iso-surfaces and/or volume renderings. Among the many compression techniques developed, wavelets (Daubechies (1998), Strang & Nguyen (1996)) have become an integral part for modern archival image storage and transmission as exemplified by the JPEG 2000 standard (Taubman & Marcellin (2001)). Moreover, extensions have been developed to the compression of geometric data and operator compression and stabilization (Castrillón-Candás (2001), Castrillón-Candás & Amaratunga (2003), Yserentant (1992)).

Applications of wavelets to volume rendering and geometric manipulation at interactive speeds have been quite limited. A visualization-specific compression scheme is presented in (Bajaj, Ihm & Park (2001*b*)), in which weights are assigned to voxels and combined with compression algorithms such that information about important features is preserved. In all of these data compression schemes, the objective is to allow rapid random access to compressed data while keeping a very high compression ratio (25:1 to 50:1), resulting in real-time visualization of extremely large datasets. However, due to the discontinuity of the basis, this scheme leads to biologically incorrect artifacts. A new representation is sought. In this paper, we deal with efficient representation and manipulation of tensor product representations of volumes over a grid using a hierarchical basis.

Numerous molecular representations have been proposed for the computer visualization of the structure of proteins, nucleic acids, and their properties (Leach (1996)). All these are different visual representation of an underlying hierarchical skeletal model of the positions of atoms, bonds, chains, and residues in the molecule. Hence, structural models aim mainly at representing the primary, secondary, tertiary and quaternary geometric structures of the molecule. A basic representation for molecular shape is the union of balls (Connolly (1983*a*)), also known as the CPK model, where each ball represents an atom of the molecule. In the presence of a solvent, the radius of the solvent atom increases the radii of the balls. This is called *solvent accessible surface* (SAS) (Connolly (1983*b*)) since it reflects the accessibility for a solvent.

The *solvent contact surface* (SCS), also known as the Connolly and Lee-Richards surface (Connolly (1983*b*,*a*), Lee & Richards (1971)), is computed by ”rolling” a solvent atom probe over the union of balls. Formally, a point p belongs to the *solvent contact surface* $\mathcal{S}_c$ of a molecule M with respect to a solvent with atoms of radius r if there exists a ball B_1 of radius r that contains p in its boundary and whose interior does not intersect M, and there is no ball B_2 of radius r whose interior contains p but does not intersect M. The solvent contact surface can be triangulated (Sanner et al. (1995), Varshney (1994)) for the purpose of visualization and other computations. Sanner and Olson (Sanner et al. (1995)) define a reduced surface for a molecule and give an algorithm to compute its triangulation in $n \log(n)$ time. (Varshney (1994)) introduces the concept of *interface surface* and develops an algorithm for simplification of the polygonal mesh representation of molecular surfaces. The simplified mesh has the properties: (a) every point is within a user-specified distance ϵ from the original mesh, (b) it is topologically consistent (has the same genus) with the original mesh, (c) its vertices are a subset of the vertices of the input mesh.

Representations for molecular surface models defined above often use spheres or piecewise linear approximation as their boundaries (Akkiraju & Edelsbrunner (1996), Connolly (1983*b*), Max (1983), Voorintholt et al. (1989), Sanner & Olson (1996), Varshney & Brooks, Jr. (1993)). Explicit representation of the boundary of the molecular surface using exact smooth trimmed NURBS (non-uniform rational B-splines) is presented in (Bajaj et al. (1997). Bajaj, Pascucci, Shamir, Holt & Netravali (2001)) introduces techniques for computing a molecular surface boundary representation as a set of NURBS patches and discusses maintaining such a NURBS representation under dynamic settings.

Other groups (Duncan & Olson (1993*a*), Leicester et al. (1988), Max & Getzoff (1988), Ritchie & Kemp (1999*b*,*a*)) represent molecular surfaces with expansions of spherical harmonic functions. This surface definition has fewer discontinuities than the hard sphere model and offers distinct advantages: realistic, and it is computationally easier to manipulate a suitably defined electron density function (Duncan & Olson (1993*b*), Koide et al. (1986)).

The molecular surface of bio-molecules can also be generated as an isosurface (implicit solvation model) of a smooth volumetric electron density function representation. The accurate computation of electron density representations from the PDB requires computations at the quantum mechanical level (Boys (1950)). One usually approximate the electron density distribution of the i^{th} atom with a Gaussian function (Mestres et al. (1997), Grant & Pickup (1995), Blinn (1982)) as $\rho_i(\mathbf{r}) = \exp\left(\frac{\mathcal{B}r^2}{R_i^2} - \mathcal{B}\right)$, where $\mathcal{B} < 0$ is the rate of decay parameter, R_i is the Van der Waals radius of the i^{th} atom and $r^2 = (x - x_{ci})^2 + (y - y_{ci})^2 + (z - z_{ci})^2$ ($\{x_{ci}, y_{ci}, z_{ci}\}$ is the center of the i^{th} atom.). A volumetric representation of the molecule may now be obtained by summing the contributions from each single atom, thus the electron density $I(x)$ for M atoms is described as

$$I(x) = \sum_{k=1}^{M} \rho_k(r) = \sum_{k=1}^{M} e^{(\frac{\mathcal{B}r^2}{R_k^2} - \mathcal{B})}. \qquad (1)$$

Volumetric maps of bio-molecules are additionally reconstructed from either tomographic electron microscopy reconstruction or from single particle reconstructions via Cryo-Electron Microscopy techniques. Several images are taken and averaged to produce high resolution of below 10 angstroms. The molecular surface of such reconstructed maps of bio-molecules can also be generated as an isosurface of this map. In Figure 1(f) an example of the Rice Dwarf Virus (RDV) is shown reconstructed from single particle Cryo-Electron Microscopy.

Isosurfaces can be computed and visualized from these volumetric maps by the well-known Marching cubes algorithm (Lorensen & Cline (1987)) which extract isosurfaces from scalar fields by traversing and triangulating each mesh cell. To accelerate searching for only contributing cells, lots of work have been exhaustively studied (Bajaj et al. (1996), Livnat et al. (1996), Wilhelms & Gelder (1992)) and later extended to time-varying fields (Shen (1998), Sutton & Hansen (2000), Bajaj, Sohn & Shamir (2001)). In Figure 1(a-c) examples of the isosurface of the electron density map of the ribosomal subunit 50S are shown. Several resolution maps are created.

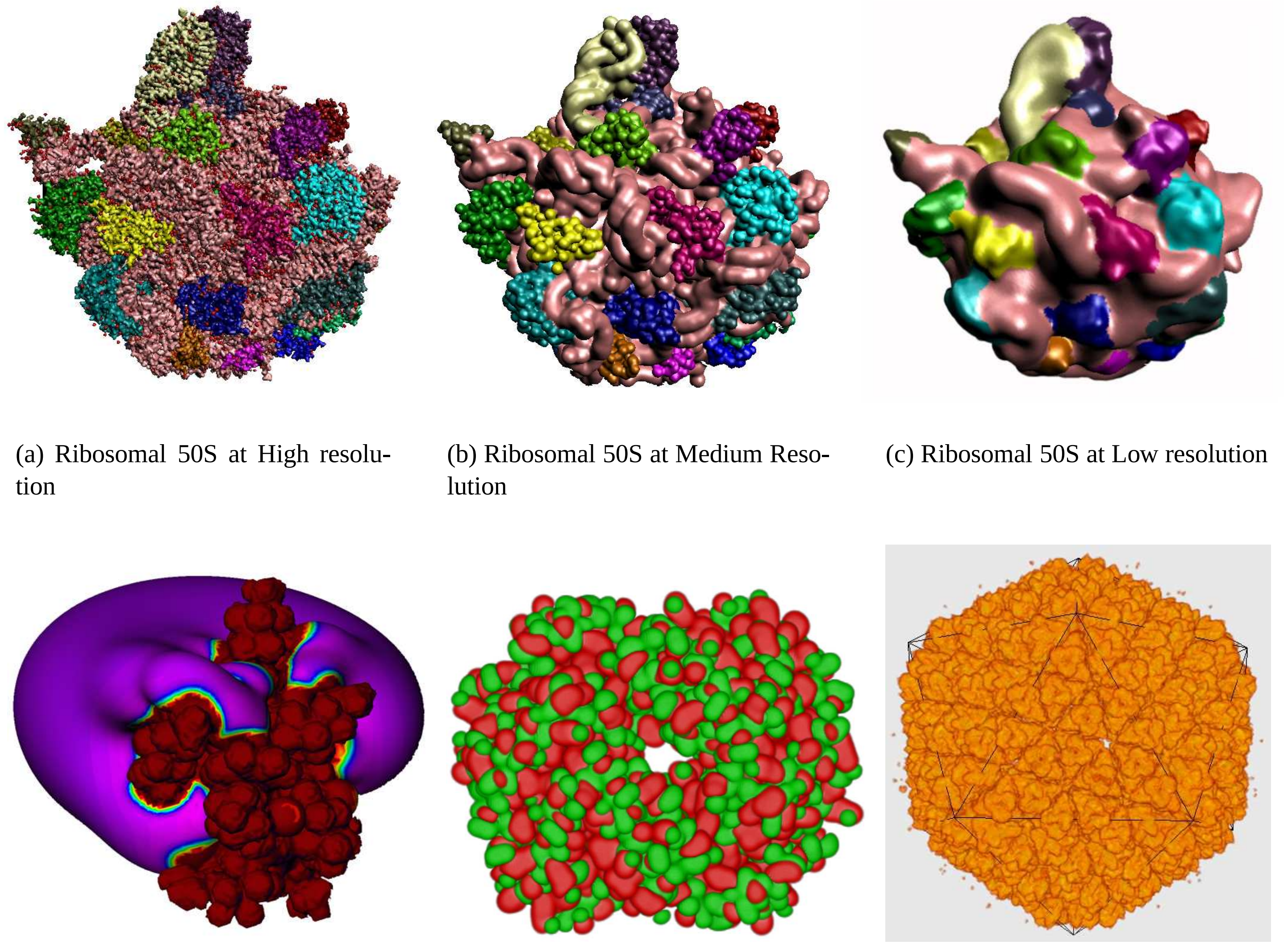
Some of the relevant molecular properties that we deal in this paper are hydrophobicity and electrostatics. We shall briefly describe how they where calculated.

The affinity towards water by different parts of the molecule plays an important part in its interaction with itself in folding and other proteins in docking. Hydrophilic / hydrophobic regions are those which have an affinity / dislike to water. It is generally seen that hydrophobic regions tend to align together, away from the solvent boundary. If we are given per atom hydrophobicity values h_i, for each atom i in a protein, one can define the hydrophobicity function ϕ as shown below

$$\phi(x, y, z) = \sum_{i=1}^{M} h_i e^{\alpha_i}$$

where $\alpha_i = \mathcal{B} \times \frac{(x - x_{ci})^2 + (y - y_{ci})^2 + (z - z_{c_i})^2}{{R_i}^2} - \mathcal{B}$. An example of the hydrophobicity map is shown in 1(e).

Associated properties of bio-molecules such as electrostatic potential fields are computed by solving the Poisson-Boltzmann equation. The numerical solutions to the equation we use here for as examples were produce by the Adaptive Poisson-Boltzmann Solver (APBS) code (Baker et al. (2001)). Again isosurface extraction and rendering provide one way of visualizing these maps. In Figure 1(d) an isosurface of the electrostatic map of glucose is produced. Electrostatic Poisson solvation energy estimations can also be created from the Generalized Born model (Lee et al. (2003)).



(a) Ribosomal 50S at High resolution

(b) Ribosomal 50S at Medium Resolution

(c) Ribosomal 50S at Low resolution

(d) Electrostatics Potential

(e) Hydrophobicity of hemoglobin. Green - hydrophobic. Red - hydrophilic.

(f) Rice Dwarf Virus Cryo-Em

Fig. 1. Visualization of Biomolecules and associated properties. (a)-(c) Electron density map at different resolution levels for the Ribosomal 50s Large Subunit. Each color represents a different chain in the molecule. For example, pink and light yellow correspond to the RNA.(d) Electrostatic potential field for glucose. (e) 256^3 Hydrophibicity map of hemoglobin, where green represents the hydrophobic part and red the hydrophilic. (e) Cryo-EM map of Rice Dwarf Virus. Note that (a)-(e) are synthetic and (f) is a natural map.

The rest of the paper is as follows: In our main section 2.1 we first show how to construct HB wavelets of all polynomial orders and then concentrate on a linear HB wavelet scheme as one possessing near very good trade-offs with compression rates and decompression times. An encoding and decoding scheme is described to achieve fast random access. Additionally, we derive error bounds on the accuracy of the linear HB wavelet representation for general molecular maps and synthetic Radial Basis Function (RBF) molecular maps. In the penultimate section 2.5 we present results of our volumetric compression results for various resolutions of bio-molecules and their associated properties along with rate distortion comparisons with non-HB wavelet methods. Finally, in the Appendix A we introduce a fully error bounded HB wavelets compression scheme.

2 Results and Discussion

In this section we construct our interactive wavelet HB compressed molecular representation and apply it to a series of natural and synthetic molecular maps. First, we discuss wavelet theory and how to construct the HB representation. Second, a zero-bit encoding scheme to achieve fast random access is derived for this basis. Furthermore, a series of error bounds are derived on the compressed format for molecular maps. Finally, we obtain timing and visualization results.

2.1 Wavelet Function Spaces with Hierarchical Basis

We construct Hierarchical Basis (HB) (Yserentant (1992)) over 3D domains in the context of *second generation wavelets* (Sweldens (1998)). In the following subsections 2.1.1-2.1.3 we show the construction of HB for the general case. In section 2.1.4, linear HB is explicitly constructed. In section 2.1.5 error bounds and our compression strategy are discussed, then followed by an appropriate encoding scheme in section 2.2.

2.1.1 Construction of the Hierarchical Basis

Our objective is to explicitly build wavelets in three dimensions with the following desirable properties:

- *Adapted to the local geometry.*
- *Easy to implement.*
- *high degree of accuracy.*
- *high degree of compression.*

To this end we construct a class of interpolating wavelet bases built from piecewise polynomials and of compact support from the class of second generation wavelets, introduced by Sweldens (Sweldens (1998)). Many of the details of the following section can be found in (Sweldens (1998), Castrillón-Candás & Amaratunga (2003), Castrillón-Candás (2001), Amaratunga & Castrillón-Candás (2001)). First we shall establish some useful notation.

Suppose we partition the domain $S_{0,0} = S \subset R^{\eta}$ into the disjoint union (partition) of K_0 parts $\{S_{1,0}, \dots, S_{1,K_0}\}$. Every single part $\{S_{1,i}\}$ is further divided into a partition of the form $\{S_{2,0}, \dots, S_{2,K_2}\}$ for some K_2. This process is repeated ad infinitum for every single partition. Now that we have partitioned our domain in a series of "multiresolution" patches we define a set of nodes on each $\{S_{j,v}\}$. Denote $x_{0,k}$ as the nodes on the level zero partition $S_{0,0}$ where $k = 1...N_0$, for some positive integer N_0. Similarly to the multiresolution partition decomposition, more nodes such as $x_{1,k}$ are added to all the level 1 partitions $\{S_{1,i}\}$. This process is then repeated for all $\{S_{j,v}\}$ ad infinitum.

Define $K(j)$ as the index set at each level j of refinement and $X_j = \{x_{j,k} \in S | k \in K(j)\}$ be the set of vertex values defined over the index set $K(j)$. The vertices's are constrained such that $X_j \subset X_{j+1}$ and $x_{j,k} = x_{j+1,k}$. The index sets are also constrained such that $K_j \subset K_{j+1}$ and the difference between the index sets from one level to another is given by $M(j) = K(j+1) \backslash K(j)$. Now that we have a multi resolution partitioning of our domain we are interested in constructing an efficient functional representation of an arbitrary function $f(\cdot)$ over the domain S. To this end we construct a series of basis function $\phi_{j,k}$ and nested *scaling function* subspaces V_j such that

$$V_j = \text{closure span}\{\phi_{j,k} \mid k \in K(j)\} \qquad \text{and} \qquad V_j \subset V_{j+1}, \;\; \forall j \in Z^*.$$

This implies that each basis function $\phi_{j,k}$ at level j can be written as a linear combination of $\phi_{j+1,k}$ at level $j+1$:

$$\phi_{j,k} = \sum_{l \in K(j+1)} h^0_{j,k,l} \phi_{j+1,l}, \;\; k \in K(j). \tag{2}$$

Now, let the *wavelets* subspaces $W_j = \text{closure span}\{\psi_{j,m} \mid m \in M(j)\}$ be the difference between V_j and V_{j+1}, i.e.

$$V_{j+1} = V_j + W_j.$$

Since $W_j \subset V_{j+1}$ then

$$\psi_{j,m} = \sum_{l \in K(j+1)} h^1_{j,m,l} \phi_{j+1,l}, \;\; m \in M(j). \tag{3}$$

The wavelet and scaling function subspaces are closely linked to a series of *dual* subspaces. Similar to wavelets and scaling functions we define a set of *dual scaling functions*$\tilde{\phi}_{j,k}$ such that

$$\tilde{V}_j = \text{closure span}\{\tilde{\phi}_{j,k} \mid k \in K(j)\} \qquad \text{and} \qquad \tilde{V}_j \subset \tilde{V}_{j+1},$$

for all $j \in Z$. Moreover, we define a series of *dual wavelet subspaces*

$$\tilde{W}_j = \text{closure span}\{\tilde{\psi}_{j,m} \mid m \in M(j)\},$$

such that

$$\tilde{V}_{j+1} = \tilde{V}_j + \tilde{W}_j.$$

It can now be seen that the dual scaling and wavelet functions satisfy

$$
\begin{aligned}
\tilde{\phi}_{j,k} &= \sum_{l \in K(j+1)} \tilde{h}^0_{j,k,l} \tilde{\phi}_{j+1,l}, \;\; k \in K(j) \\
\tilde{\psi}_{j,m} &= \sum_{l \in K(j+1)} \tilde{h}^1_{j,m,l} \tilde{\phi}_{j+1,l}, \;\; m \in M(j).
\end{aligned}
$$

The subspaces $V_j, W_j, \tilde{V}_j$ and $\tilde{W}_j$ are all interrelated and satisfy the following biorthogonal conditions

$$V_j \perp \tilde{W}_j, W_j \perp \tilde{V}_j. \tag{4}$$

In addition, the basis functions $\phi_{j,k}, \tilde{\phi}_{j,k}, \psi_{j,m}$ and $\tilde{\psi}_{j,m}$ satisfy the orthogonality conditions:

$$\langle \phi_{j,k}, \tilde{\phi}_{j,l} \rangle = \delta_{k-l}, \;\; \langle \psi_{j,m}, \tilde{\psi}_{j,l} \rangle = \delta_{m-l}, \tag{5}$$

where

$$\delta_{k-l} = \begin{cases} 1 & k = l \\ 0 & k \neq l \end{cases}.$$

We can now represent an arbitrary function $f(\cdot) \in V_j$ as a linear combination of the single scale basis functions

$$f = \sum_{k \in K(j)} \lambda_{j,k} \phi_{j,k} + \varepsilon,$$

where $\lambda_{j,k} = \langle f, \tilde{\phi}_{j,k} \rangle$ and some small error ε, or in terms of its equivalent *multi resolution* representation:

$$V_j = V_0 \oplus \sum_{i=0}^{j-1} W_i, \Rightarrow f = \sum_{k \in K(0)} \lambda_{0,k} \phi_{0,k} + \sum_{i=0}^{j-1} \sum_{m \in M(i)} \gamma_{i,m} \psi_{i,m} + \varepsilon,$$

where $\gamma_{j,m} = \langle f, \tilde{\psi}_{j,m} \rangle$. In general, the multiresolution form leads to a more compact representation of the function f.

In Figure 2(a) an example of a simple multiresolution representation over 1D domain is shown. The primary scaling and wavelet equations are well suited for the representation of smooth functions in f with linear accuracy. Notice, that the duals are Dirac's and in general will not be well suited for functional representation. Other smoother duals can be constructed, (see Chapter 3 in Castrillón-Candás (2001)). However, the duals will be used implicitly for showing that the multi resolution decomposition of the primary functions exists and for estimating the decay of the wavelet coefficients.

This is one of the most simple wavelet representations and has been successfully employed for adaptive meshing and preconditioning of finite element matrices (Yserentant (1992)). Moreover, in a departure to traditional wavelet theory biorthogonality and the wavelet structure is preserved even if the nodes are irregularly placed. This class of wavelet is an example of an interpolating wavelet basis. In the following section we shall concentrate further into this class of wavelets and show how to construct linear interpolating wavelets in R^3.

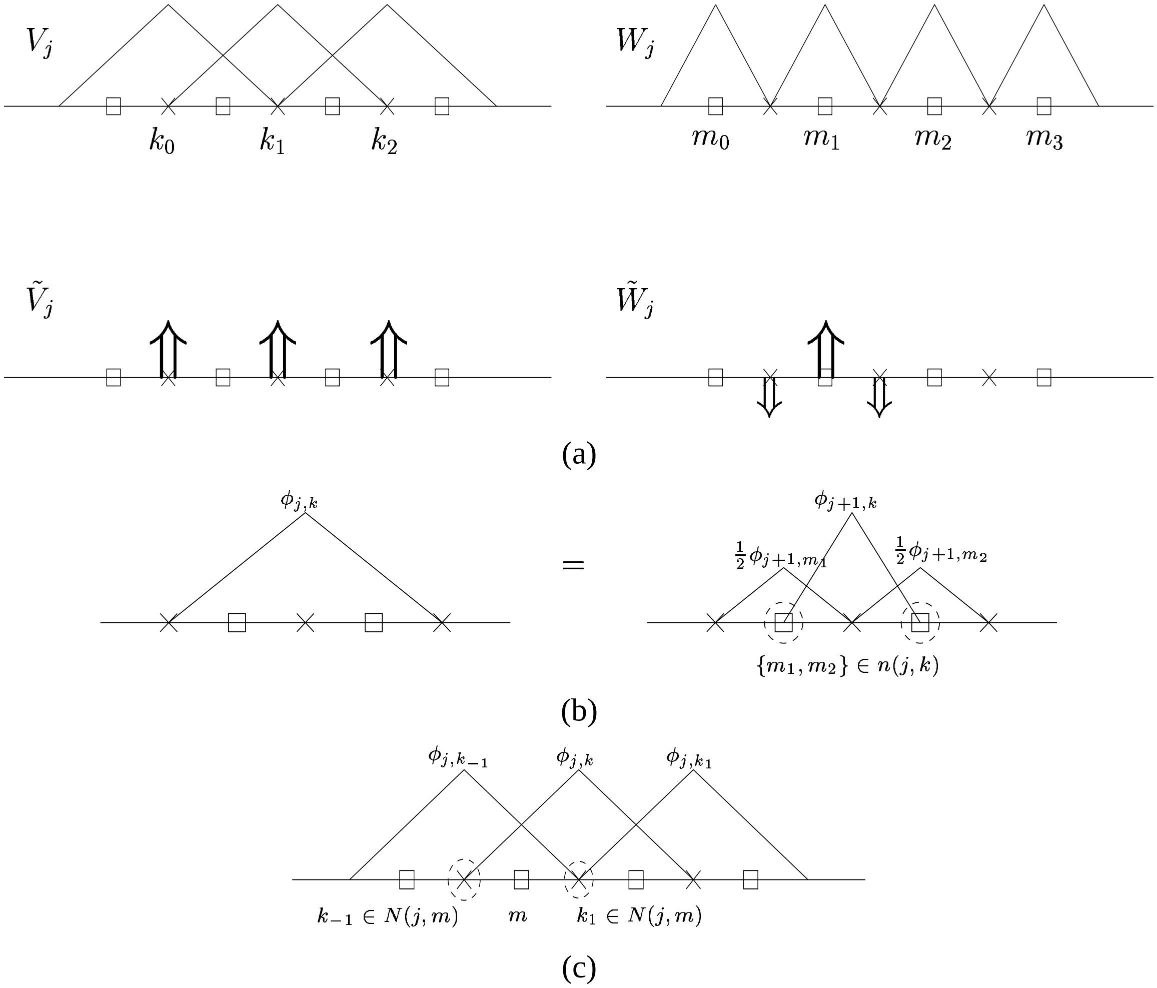


Fig. 2. Construction of Second Generation Wavelets. (a) Simple example of second generation wavelets and scaling functions. The primary scaling functions $\phi_{j,k}$ are constructed as linear hats on $K(j)$ and the wavelets are chosen such that $\psi_{j,m} = \phi_{j+1,m}$, we let $h_{j,m,l} = 0$ for all $l \neq m$. The dual scaling functions and wavelets are easily constructed as delta functions, i.e. $\tilde{\phi}_{j,k_1} = \delta(x - x_{j,k_1})$ and $\tilde{\psi}_{j,m_1} = \delta(x - x_{j,m_1}) - \frac{1}{2}\delta(x - x_{j,k_0}) - \frac{1}{2}\delta(x - x_{j,k_1})$. It is easy to see that these four spaces of function satisfy biorthogonality. Other smoother duals can be constructed (Castrill´on-Cand´as (2001)). (b) In this linear “hat” example, the scaling function $\phi_{j,k}$ is constructed from three scaling functions at the next level of resolution $j+1$. The neighborhood $n(j,k)$ is formed from nodes in $M(j)$ that are needed to construct $\phi_{j,k}$. (c) This figure illustrates the neighborhood $N(j,m)$ for the index point $m \in M(j)$. Both scaling functions $\phi_{j,k_{-1}}$ and $\phi_{j,k}$ use the node m for their respective refinement equation, thus m belongs to $n(j,k_{-1})$ and $n(j,k)$. We say then that $\{k_{-1}, k\}$ belong to $N(j,m)$.

2.1.2 *Interpolating wavelets basis*

An interpolating function $\phi_{j,k}$, satisfies the following condition

$$\phi_{j,k}(x_{j,k'}) = \delta_{k-k'}, \forall k, k' \in K(j).$$

Since all the scaling functions are interpolating the refinement equation (2) may be written in a more compact form as

$$\phi_{j,k} = \phi_{j+1,k} + \sum_{m \in n(j,k)} h^0_{j,k,m} \phi_{j+1,m}, \ \forall j \text{ and } \forall k \in K(j), \tag{6}$$

where the local neighborhood $n(j,k)$ is defined as follows:

- $n(j,k) \subset M(j)$ corresponds to the set of indicies in $M(j)$ needed to construct $\phi_{j,k}$.

Similarly we define the counterpart to $n(j,k)$ as

- $N(j,m) \subset K(j)$ is equal to $\{k \in K(j) \mid m \in n(j,k)\}$.

$N(j,m)$ will be useful to apply the wavelet transforms.

For the linear interpolating basis shown in Figure 2 only two coefficients are needed to satisfy the refinement equation (6), thus for each j and k, $n(j,k) = \{m_1, m_2\}$ and the coefficients $h^o_{j,k,m_1} = h^o_{j,k,m_2} = \frac{1}{2}$, see Figure 2(b). In figure 2(b) we show how to construct the neighborhood $N(j,m)$.

Assuming that we can find a series of scaling functions and coefficients $h^0_{j,k,m}$ that satisfy the refinement equation (6) a simple *interpolating* wavelet can be constructed. At every single node $m \in M(j)$ choose $h^1_{j,m,l} = \delta_{l,m}$; this implies the wavelet $\psi_{j,m} = \phi_{j+1,m}$. Although we can build more complex wavelets by applying the *lifting scheme* (Sweldens (1996)), in this paper we restrict ourselves to this simple wavelet.

Having set up the framework for interpolating wavelets we can move from single scale at level j to a multi scale representation for an arbitrary function f by applying the wavelet transform

$$\begin{aligned} \forall m \in M(j), \ \ \gamma_{j,m} &= \lambda_{j+1,m} - \sum_{k \in N(j,m)} h^0_{j,k,m} \lambda_{j+1,k}. \\ \forall k \in K(j), \ \ \lambda_{j,k} &= \lambda_{j+1,k}. \end{aligned}$$

or vice versa by applying the inverse wavelet transform

$$\forall k \in K(j), \ \ \lambda_{j+1,k} = \lambda_{j,k},$$

$$\forall m \in M(j), \;\; \lambda_{j+1,m} = \gamma_{j,m} + \sum_{k \in N(j,m)} h^0_{j,k,m} \lambda_{j+1,k}.$$

(See Sweldens (1998) for a proof). Both transforms require only $\mathcal{O}(N)$ computations, where N is the total number of vertices at level $K(j)$. Note that if $f \in V_n$ then $\lambda_{n,k} = \langle f, \tilde{\phi}_{j,k} \rangle = f(x_{j,k})$. The last step allows us to to approximate f by using samples of the function f at the highest resolution level n and avoid explicitly constructing the dual scaling functions (see Castrillón-Candás & Amaratunga (2003) on how to construct them), this leads to a more simple implementation. In the following section we describe a large class of coefficients $h_{j,k,m}$ that satisfy equation 6.

2.1.3 Spatially Adapted MultiWavelets

We now flesh out the framework introduced in the previous section, by showing how to construct interpolating piecewise scaling functions of arbitrary degree. Such constructions are introduced in (Castrillón-Candás & Amaratunga (2003)).

Let $\{S_{j,v}\}$ be a multi resolution partitioning of the domain $S \subset R^3$ and choose a set of linearly independent functions $\{P_i(x)\}$, $i = 0, 1, \ldots, M-1$. Moreover, for all the nodes $x_{z,k}$ in $S_{j,v}$, $z \geq j$, all the neighboring nodes $x_{z+1,m}$, such that $m \in n(z,k)$, are contained in $S_{j,v}$. Furthermore, assume that the number of elements of $N(z,m)$ is equal to some positive integer M. If the collection of nodes $\{x_{z,k}; z \geq j\}$ in $S_{j,v}$ are distinct and form a dense subset in $S_{j,v}$ then there exists an interpolating scaling function (see Castrillón-Candás & Amaratunga (2003) for a proof)

$$\phi_{j,k}(x) = \sum_{i=0}^{M-1} a_i P_i(x) \quad\;\; a.e.$$

over every $S_{j,v}$ for some set of unique coefficients $\{a_0, a_1, \ldots, a_{M-1}\}$, such that

$$\sum_{i=0}^{M-1} a_i P_i(x_{j,k'}) = \begin{cases} 1 & k' = k \\ 0 & k' \neq k \end{cases} \quad k' \in K(j).$$

and

$$\int_S P_i(x) \tilde{\psi}_{j,k}(x) \; dS(x) = 0.$$

The support of $\phi_{j,k}$ is equal to union of all the neighboring partitioning of the node k and the coefficients of the refinement equation are given by

$$h^0_{j,k,l} = \begin{cases} \delta_{k-l} & l \in K(j) \\ \sum_{i=0}^{M-1} a_i P_i(x_{j,l}) & l \in n(j,k), x_{j,l} \in S_{j,v} \end{cases}. \tag{7}$$

The wavelets may now be constructed by letting $\psi^{old}_{j,m} = \phi_{j+1,m}$. In this paper, for simplicity, we construct a linear interpolating wavelet basis.

2.1.4 *Construction of linear Volumetric Rectangular Wavelets on 3D domains*

We now have the framework to construct a multiresolution decomposition of a volumetric mesh of the form

$$V_j = V_0 \oplus \sum_{i=0}^{j-1} W_i,$$

over trilinear functions. Such constructions provide high degree of accuracy while simple to implement.

2.1.4.1 Linear Scaling function construction

(1) Choose the order of approximation of the scaling functions. For a linear approximation basis we choose $\{P_0, \ldots, P_7\} = \{1,\ x_1,\ x_2,\ x_3,\ x_1x_2,\ x_1x_3,\ x_2x_3,\ x_1x_2x_3\}$, thus M = 8.

(2) Denote the initial level of resolution $j = 0$ and define an initial coarse cube mesh C_0 scaled to one.

(3) Subdivide the mesh equally into eight new cubes and form the set C_1 as all the cubes at resolution level 1.

(4) Place all 27 nodes at level $K(1)$ on the cubes C_1. The eight nodes of $k \in K(0)$ are put on on each corner of C_0 and the rest of the 19 nodes $m \in M(1)$ are placed according to the pattern in Figure 3(a) i.e. mid points. All, 19 nodes $m \in M(1)$ form $n(0,k)$ and $h^0_{0,k,l} = \sum_{i=0}^{M-1} a_i P_i(x_{0,l}), \forall l \in n(j,k)$. Since $\{a_0, \ldots a_{M-1}\}$ are chosen to make the trilinear polynomial over C_0 interpolating, only six nodes are non-zeros, which correspond to:
- $h^0_{0,k,l} = \frac{1}{2}$ if $l = a$ (edge node)
- $h^0_{0,k,l} = \frac{1}{4}$ if $l = b$ (face node)
- $h^0_{0,k,l} = \frac{1}{8}$ if $l = c$ (interior node)
- $h^0_{0,k,l} = 0$ otherwise.

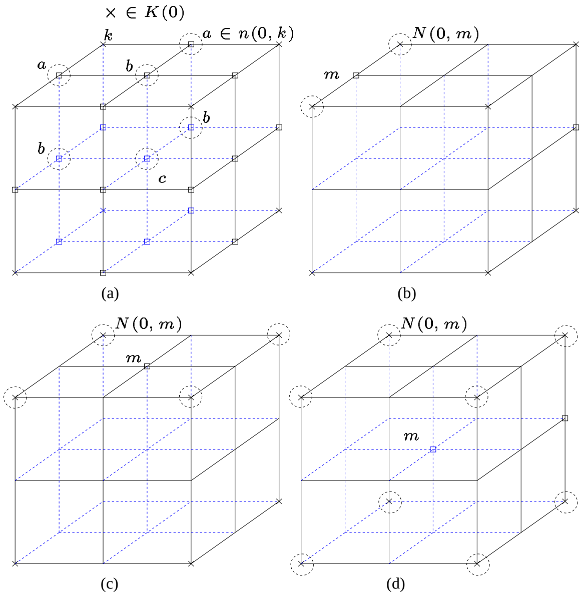
(See Figure 3(a)). The neighborhood $N(0,m)$ cardinality is also M, however, in practice we also need less since the polynomial $\sum_{i=0}^{M-1} a_i P_i(x)$ is zero at many of the corners. Thus, depending if the node is an edge, the interior of a face of the neighborhood $N(0,m)$ is chosen accordingly. See Figures 3(b), (c) and (d).

(5) For each of the subdivided cube at level C_1 we repeat the above steps and place the new nodes $m \in M(1)$. The neighborhoods $N(1,m)$, $n(1,k)$ and the coefficient $h^0_{1,k,l}$ are are also calculated on the subdivided cubes.

(6) The process is repeated ad infinitum such that the set of $\{x_{j,k}\}$ forms a dense subset of the volumetric domain. Notice that each V_j on C_j has a dimension of $(2^j + 1)^3$.

(7) Apply the spatially adapted multiwavelet existence theorem from section 2.1.3 (or Castrillón-Candás & Amaratunga (2003)). This implies that every cube $v \in C_j$ consists of a trilinear polynomial. The scaling functions $\phi_{j,k}$ at any vertex $x_{j,k'}$ is equal to the union of all neighboring cubes in C_j. Denote this union of cubes as $C^{\phi}_{j,k}$.

2.1.4.2 Linear Wavelet Construction With all the scaling functions and refinement equation constructed a simple wavelet can be formed at every single node $m \in M(j)$, by letting $h^1_{j,m,l} = \delta_{l,m}$. This implies the wavelet has the form $\psi_{j,m} = \phi_{j+1,m}$ and support $C^{\psi}_{j,m} = C^{\phi}_{j+1.m}$.



(a) (b) (c) (d)

Fig. 3. Trilinear HB wavelet construction. (a) The distribution of nodes for the construction of trilinear scaling functions on a rectangular mesh. For these functions we pick $\{P_0, \ldots, P_7\}$ to be equal to all the terms in the expansion of the shape function $(1 + x_1)(1 + x_2)(1 + x_3)$. The 8 starting nodes in $K(j)$ are marked as ×. The mesh is refined into eight elements and the new nodes belonging to $M(0)$ are placed in the middle (although this is not a hard restriction). As we observe, the cube is subdivided into 8 elements with identical starting node patterns. The new nodes that belong to $M(0)$ form the set $n(j, k_i)$ for every single node $k_i \in K(j)$. However, many of the coefficients $h_{j,k,m}$ are equal to zero, thus for any node $\times \in K(0)$ only six nodes in $m \in M(0)$ are needed (a,b,c). (b) Example of the nodes belonging to $N(0, m)$. If m is an edge node then only the two immediate neighbor nodes are needed to completely determine the polynomial in equation 2.1.3.(c) For a interior face node 4 nodes are needed for $N(0, m)$.(c) For an interior cubic nodes $N(0, m)$ has eight nodes.

2.1.4.3 Wavelet transform We can now interchange between the single scale and multi scale representations $V_n \leftrightarrow V_0 \oplus \sum_{i=0}^{n-1} W_i$ by applying the wavelet transform. Since $\lambda_{n,k} = f(x_{n,k})$ are the coefficients for the trilinear approximation at the highest resolution n and $\lambda_{j,k} = \lambda_{j+1,k}$ for all $j \leq n$ then

$$\forall m \in M(j), \;\; \gamma_{j,m} = f(x_{j+1,m}) - \sum_{k \in N(j,m)} h^0_{j,k,m} f(x_{j+1,k})$$

and the inverse

$$\forall m \in M(j), \;\; f(x_{j+1,m}) = \gamma_{j,m} + \sum_{k \in N(j,m)} h^0_{j,k,m} f(x_{j+1,k})$$

since $\lambda_{n,k} = f(x_{n,k})$ are the coefficients for the trilinear approximation at the highest resolution n. Notice, that since $\gamma_{j,m}$ only depends on samples of the density function the wavelet transform can be done in any order. In fact, the wavelet transform at level j is exactly a trilinear interpolation of the data at level j minus the values of data at the node of the wavelets, which is exactly the interpolation error for a trilinear grid.

2.1.5 Adapted Compression Strategy for General Volumetric Maps

The compression scheme can be broken in two steps. First, the data has to be sparsified using a suitable functional representation as described in this section. The second step is produce an *encoding* which tells where the non-zero terms are located. In this paper we adapt the zero-bit encoding scheme that we have developed (Bajaj, Ihm & Park (2001*b*)) to HB.

The linear HB is well suited for the representation of smooth data with localized discontinuities. We shall first derive some bounds on the decay of the wavelet. These bounds indicate that the wavelets we have constructed are very well suited to compress data on a local and global scope. We also show some strategies for compression of molecular data.

Let $f : V_n(C) \subset \mathbb{R}^3 \to \mathbb{R}$ be a density function twice differentiable defined on a unit cube C. Let n be the maximum multiresolution level and $V_1 \subset V_2 \subset \ldots V_n$ the scaling function spaces defined on C, then the density function can be expanded in the single scale formulation in V_n as

$$f = \sum_{k \in K(n)} \lambda_{n,k} \phi_{j,k} + \varepsilon.$$

For the interpolating projection of f onto V_n the coefficients are equal to $\lambda_{n,k} = f(x_{n,k})$ for all $k \in K(n)$. For the previously derived trilinear function space V_n the global error ε is bounded as

$$|\varepsilon| \leq 2^{-2n-3} \left(\| \frac{\partial^2 f}{\partial x_1^2} \|_{\mathcal{L}^\infty(C)} + \| \frac{\partial^2 f}{\partial x_2^2} \|_{\mathcal{L}^\infty(C)} + \| \frac{\partial^2 f}{\partial x_3^2} \|_{\mathcal{L}^\infty(C)} \right),$$

where $\|f\|_{\mathcal{L}^\infty(C)} = \sup_C |f|$. However, the coefficients $\lambda_{n,k}$ strictly depend on the value of the function f at the vertex $x_{n,k}$ and in general will not have fast decay, thus not suitable for compression. An alternative representation is the multiresolution form

$$f = \sum_{k \in K(0)} \lambda_{0,k} \phi_{0,k} + \sum_{i=1}^{n-1} \sum_{m \in M(i)} \gamma_{i,m} \psi_{i,m} + \varepsilon.$$

Since both the single scale and multi scale representation span the same space, the error ε will be the same. However, the advantage of the multiresolution representation is that the coefficients $\gamma_{j,m}$ vanish quickly.

Recall that the wavelet coefficients $\gamma_{j,m}$ can be obtained from a few samples of the original function:

$$\gamma_{j,m} = f(x_{j,m}) - \sum_{k \in N(j,m)} h^0_{j,k,m} f(x_{j+1,k}). \tag{8}$$

A more careful examination of Equation (8) and the neighborhood $N(j,m)$ shows that the term $\sum_{k \in N(j,m)}$

$h^0_{j,k,m} f(x_{j+1,k})$ is the interpolating estimate of the $f(x_{j,m})$. For the trilinear wavelets derived previously, the value of edge node m is estimated from the two adjacent nodes, for a face node the estimate is from all four corners, and for the interior node all eight corners of the cube are used (See Figure 3(b),(c) and (d)). This implies that

$$|\gamma_{j,m}| \leq 2^{-2j-3} \left(\|\frac{\partial^2 f}{\partial x_1^2}\|_{\mathcal{L}^\infty(C^\psi_{j,m})} + \|\frac{\partial^2 f}{\partial x_2^2}\|_{\mathcal{L}^\infty(C^\psi_{j,m})} + \|\frac{\partial^2 f}{\partial x_3^2}\|_{\mathcal{L}^\infty(C^\psi_{j,m})} \right).$$

The wavelet coefficient values depend on the level of resolution j and the maximum of the local second derivative and are exactly the trilinear error. This estimate can be used to derive a strategy for the construction of hierarchical adaptive representation of the density function f. If f contains a discontinuity the neighborhood wavelet coefficients are in general large, but away from the discontinuity the decay is very fast. This makes wavelets very well suited for representing local and global variations. In contrast, Fourier and spherical harmonics have very slow decays due to the bias toward global representations.

It should be pointed out that this linear HB representation is essentially an adaptive mesh representation. However, most adaptive meshes are discontinuous unless a computational expensive step is used apriori to avoid hanging nodes. These discontinuities lead to artifacts for visualization of molecular volumes and surfaces, in particular the isosurface extraction would be very poor.

One strategy for compression is to compute $\gamma_{j,m}$ for every single m and level of resolution j and store only the values above a predetermined level dependent value ϵ_j. Thus depending on the level of resolution and the derivative $\|\partial^2 f\|_{\mathcal{L}^\infty(C^\psi_{j,m})}$ we sparsify the representation.

(1) Start at level $j = 0$
(2) $\forall m \in M(j)$ compute $\gamma_{j,m}$
(3) Store wavelet coefficients only if $\gamma_{j,m} < \epsilon_j$
(4) $j := j + 1$
(5) Go to step 2 until $j = n$

The downside of this interpolating representation is that the linear interpolating basis is not *Reisz stable* (Daubechies (1998)), thus the traditional choice of removing the wavelet coefficients with the minimum energy does not lead to an optimal L^2 representation of the density function f.

Since the wavelets are the error between one level of resolution and the next one choice is to pick $\epsilon_j = \epsilon * \sup |f|$, where $\epsilon \in [0, 1]$. Thus eliminating the wavelets with low contribution to the error. However, low level wavelets have a higher weight in the total error. Instead we weight the threshold according to the level of resolution, i.e. $\epsilon_j = \frac{\epsilon * \sup |f|}{2^{n-j-1}}$.

Although this scheme does produce highly compressed maps with good accuracy, it does not lead to an error bounded scheme. In the Appendix A we present a fully error bounded compression wavelet scheme. Moreover, in Appendix B error bounded mesh sizes and local wavelet updates for Radial Basis Function molecular maps are derived.

2.2 Encoding

As with all compression schemes, a suitable encoding of the non zero wavelets is a necessary step. Many such schemes have been developed for efficient transmission of compressed data streams . However, for visualization purposes such as volume rendering and iso-contouring, very few encoding schemes have been developed. Such encoding schemes should aim for high compression ratios and fast random access. The zero-bit encoding scheme, introduced by Bajaj, Ihm & Park (2001*a*), uses a two-stage significance map to achieve a good compromise between these two aspects. Whether wavelet coefficients are null or not is stored in zero-bits while the significance information of the detail coefficients of each non-null detail node is stored in additional bits. Nevertheless, since the encoding is limited to Haar wavelets, displeasing and biologically incorrect visual artifacts appear at high compression. To solve this problem, a novel encoding scheme is created on the basis of zero-bit encoding for HB (Hierarchical Basis) linear wavelets.

The wavelet coefficients are assumed to be mostly zeros, with a few localized non-zero coefficients, as shall be the case for a large number of Cryo-EM and molecular synthetic maps. Under this assumption, a multiresolution encoding scheme can be built to optimally access the concentrated information. In this encoding scheme the rectangular 3D volume data is subdivided into blocks of equal size (which can be predetermined by the user). Each block is further segmented into cells which contain a fixed number of voxels. Now, we run the wavelet transform on each cell separately. Since the transformed data usually has large swaths of zeros, a large number of cells or blocks can be ignored. By constructing an appropriate multi resolution indexing scheme, an efficient encoding of the non-zero coefficients is built.

A 3D volume of N^3 voxels is divided into blocks of equal size n^3 where N is a multiple of n. The blocks are subdivided into cells of size m^3, where n is a multiple of m. The next step is to index the blocks and cells appropriately. Provided that there are $p = \frac{N^3}{n^3}$ blocks in the 3D volume, a p-bit length word is defined. A non-empty block is tagged as a ’1’ bit, while empty block as a ’0’ bit. The ’0’ indicates that further searches in that block are not required. Similarly, for each non empty block, a word is constructed such that a ’1’ bit indicates a non empty cell. (See Figure 4(a)).

Now, each block contains a byte to represent the Number of non-null cells (NNNC) in it. When NNNC is zero, the block is a null block, which means that all coefficients in the block are zero. Otherwise, Cell Bit Flag Table(CBFT), in which one bit is zero if and only if its corresponding cell is null, is used to indicate whether each cell in the block is null or not.

When a cell is not null, a one or two tier scheme is exploited to represent it. To encode the non-zero wavelet coefficients in each cell for wavelet level 0 and 1, we use one bit to index a voxel. For level 2 we employ a two tier “zero-bit” tree. The coefficients are separated in groups of two bytes (16 bits). If all 16 coefficients are empty, we indicate it with a ’0’ at the first level of the tree, otherwise a ’1’ label is used. For each non-empty group, a 16 bit word is used to indicate the non zero coefficients. (See Figure 4(b)).

The non-zero values are stored in a data stream, which can be retrieved by a combination of Zero-bit, Significance Map Stream (ZSMS) table and the Cell Information Array (CIA). The two tier encoding scheme is stored in the ZSMS table. For each cell, the CIA contains the Data Offset that gives the starting position in the data stream, in addition to the Zero-bit Offset, which indicates the beginning position in the ZSMS table for a cell.

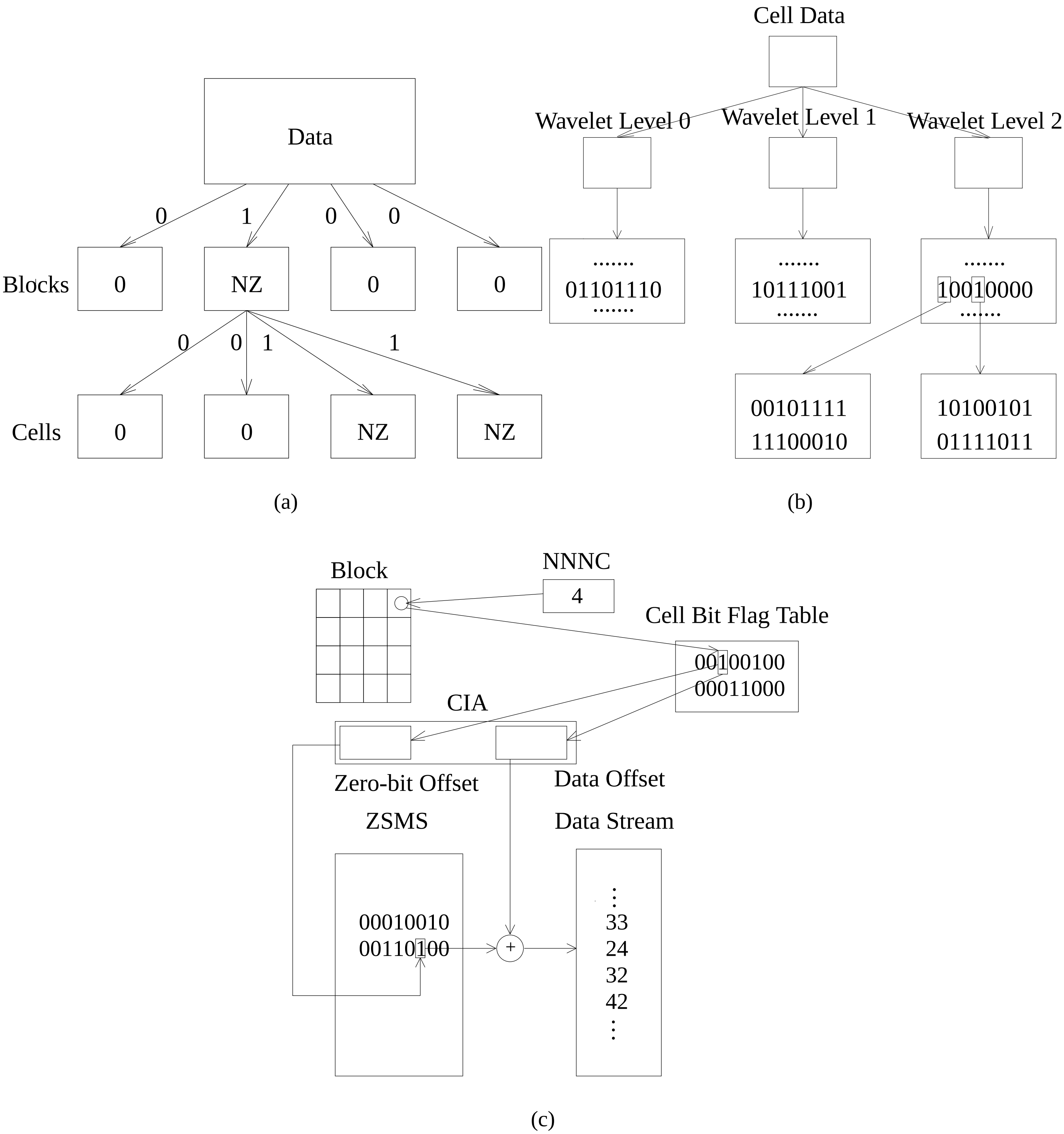


Fig. 4. Scheme for Hierarchical Encoding and Decoding. (a)The data is decoded into hierarchically into blocks and cells. Bit vectors are used to show if a block or cell is empty or not. (b) Encoding of wavelets in each cell of variable size, although in our implementation we used cells of 9^3. (c) Procedures and structures used to decode wavelet coefficients.

2.3 Decoding

With this encoding scheme we can efficiently recover any original data coefficient. The decoding is obtained by the following steps (see Figure 4(c)):

(1) Identify the wavelet coefficients contributing to that data point. For each coefficient, follow the next three steps.
(2) Find out the block that contains the coefficient. Check the NNNC of the block. If it is zero, discontinue and return a zero. Otherwise, go to the next step.
(3) Check the bit in CBFT associated with the cell containing the coefficient. If it is '0', the cell is a zero cell and discontinue. Otherwise, go to the next step.
(4) Get the bit associated with the coefficient from ZSMS table by Zero-bit Offset. If the bit is zero, the coefficient has the value of zero. Otherwise, add the position with the Data Offset of the cell to get the position of the wavelet coefficient in the data stream and fetch its value accordingly.
(5) Once all the significant coefficients are retrieved, the original data point is calculated with the wavelet basis.

2.4 Random Access

An important feature of our encoding scheme is fast random access with high compression rates. In this section, the worst case reconstruction and lookup computational costs for decoding a voxel are analyzed. In addition, the average number of sparsified wavelets needed to represent the entire domain is presented.

Let γ_i represent the ratio of the number of non zero wavelet coefficients to the total number of vertices's at level i for all non zero cells. Similarly, let α be the ratio of the number of non zero blocks to total number of blocks and β be the ratio of the number of non zero cells to total number of cells located in non zero blocks. In section 2.5, we use 9 as the cell size, so i is no more than 3. To decode the value of any voxel in the cell, we first identify the number of wavelets that contribute to the reconstruction. For every wavelet, the ZSMS table is searched to provide the coefficient value. The total decoding cost is the sum of the lookup cost in the ZSMS tree and the computational reconstruction cost. To decode a voxel, the following procedure is executed:

(1) If the voxel to be reconstructed is located at scaling function level 0, then it is the value of the cell and no reconstruction is needed. But we still need to perform one lookup. Otherwise suppose that the voxel is located at a level j wavelet vertex.
(2) Subdivide the cube into eight new subcubes and let $i = 0$.
(3) Locate the subcube that contains the voxel to be reconstructed at level i and retrieve the other seven coefficients which belong to that cube.
(4) Add the wavelet contributions to the original voxel location's value.
(5) Go to the next level, i.e. $i := i + 1$.
(6) Subdivide each subcube into eight again.
(7) Repeat Steps 3-6 until $i = j$.

To reconstruct a voxel in V_0, at most 15 arithmetic operations for a interior node are needed, where an arithmetic operation denotes addition, subtraction, multiplication or division. To reconstruct a voxel in $W_0 = V_1 - V_0$, 3*3+7*3+15=45 more arithmetic operations are needed in the worst case. In the previous formula, 3 arithmetic operations for an edge node, 7 arithmetic operations for a face node and 15 arithmetic operations for a interior node are needed. So 15+45=60 arithmetic operations are used to reconstruct a voxel in W_0 in the worst case. The worst case analysis can also be extended to W_1 and W_2 in a similar way and shown in Table 3(a).

In general, we are more interested in the average cost to decode a single voxel, since this would determine the speedup due to our encoding scheme. To reconstruct the voxel at level j, then $7j + 1$ wavelets are needed at most. Since γ_i is the ratio of non zero wavelets to the total number of vertices at level i for non zero cells, on average only $8 + 7\gamma_0 + 7\gamma_1 + 7\gamma_2$ wavelets and scaling functions are needed to decode the voxel in any non-empty cell. This leads to an upper bound of the average arithmetic cost, which is

$$\alpha\beta(15 + 60\gamma_0 + 105\gamma_1 + 150\gamma_2).$$

In addition to the computational cost, memory lookup costs can also be significant. In the following analysis we estimate the average memory lookup cost.

To retrieve any wavelet picked at random, we must first check that blocks and cells are non-zero. Since $(1 - \alpha)$ is the probability that a block is zero and $(1 - \beta)$ is the probability that the cell is zero in the non zero blocks, we must at least have an average lookup cost of $(1 - \alpha) + 2\alpha(1 - \beta)$. After identifying the block and cells, any random wavelet we choose can land in level 0, 1, and 2 with probability 19/721, 98/721 and 604/721. Since Level 2 has a two-tier zero-bit encoding scheme, at most 5 lookups are needed, which implies that the average cost for landing in level 2 is $3020\alpha\beta(1 - \gamma_2)/721$. A similar analysis is done for level 1 and 0, however only a one level zero-bit encoding, and hence only a maximum of 4 lookups are needed. The total lookup memory cost is

$$(1 - \alpha) + 2\alpha(1 - \beta) + \frac{3020\alpha\beta\gamma_2 + 392\alpha\beta\gamma_1 + 76\alpha\beta\gamma_0}{721}.$$

2.5 Timings, errors and visualizations of molecular maps

To gauge the effectiveness of the HB representation (zerobit and non zerobit) to molecular maps, we apply it to different types of synthetic and natural data. In particular, electron density, electrostatic and Cryo-EM maps that have been described in section 1. Moreover, theoretical and actual random access timings for the zero-bit decoder are presented.

We first test the HB representation with zero-bit encoding on the electron density map of the NMR Structure of Ectodomain of Siv Gp41 (2EZP in the PDB). The molecular map is created by Equation 1 with a rate of decay $\mathcal{B} = -2.3$, which corresponds to 1-3 angstroms resolution. In Table 1(a) file compression ratio (CR) results for 2EZP are shown. As the threshold level ε is increased we observe that CR increased but the PSNR of the reconstructed image degrades. This can be observed in Figure 5(a)-(d), however, notice that even for high compression rates, the molecular surface is retained very well. We obtain similar results for compression of the 50s Large Ribosomal Subunit electron density and electrostatic map, as shown in Figure 2.5(a)-(h) and Table 1(b).

A Cryo-EM map of the Rice Dwarf Virus was used as an example of a natural map to test our compression scheme on. The original dataset dimension was 241×481^2. In Figure 5(e), we show a volume rendering of the original map. We again used three different thresholds for wavelet compression as shown in Figure 5(f)-(h). High visual accuracy of the structure of the virus can be observed even at a compression ratio of 37:1. In Table 1(c), we present details of our compression algorithm for more threshold levels.

Another interesting feature to observe is the compression capabilities of our scheme with respect to maps of varying resolutions. In table 2(a)-(c) compression results for the 50s Large Ribosomal Subunit electron density at rates of decay $\mathcal{B} = -1, -2, -5$. For a fixed threshold level, we observe that both the PSNR and file Compression Ratio increases with increase in $\mathcal{B}$ (i.e. decrease in resolution).

We now compare our zero-bit continuous HB scheme with zero-bit Haar compression, which to our knowledge, is the only other interactive compression format. In table 2(d) we directly compare our scheme with Haar for the same file compression ratios. It is clear from the data, that our scheme is superior. This is supported visually in Figure 7. Notice that the visual quality for HB is much superior.

2.5.1 Random Access Times

To give some meaning to the costs that we have derived in section 2.4 we study the 513^3 electron density map of the 50s ribosomal large unit with decay parameter $\mathcal{B} = -1$ (see Table 2(a)-(c)). For each cell we obtain the parameters $\alpha, \beta, \gamma_0, \gamma_2$ and γ_3 and average them for all the cells. The following values are obtained: $\alpha = 0.421$, $\beta = 0.784$, $\gamma_0 = 0.384$, $\gamma_1 = 1.651 \times 10^{-3}$ and $\gamma_2 = 0.0$. Simple calculations give the average arithmetic cost of 12.61 and the average lookup memory cost of 1.145 for accessing a single voxel. Considering that the optimal lookup cost is 1, we obtain very good access speeds. Moreover, arithmetic operations in many computer architectures are much faster than memory lookups.

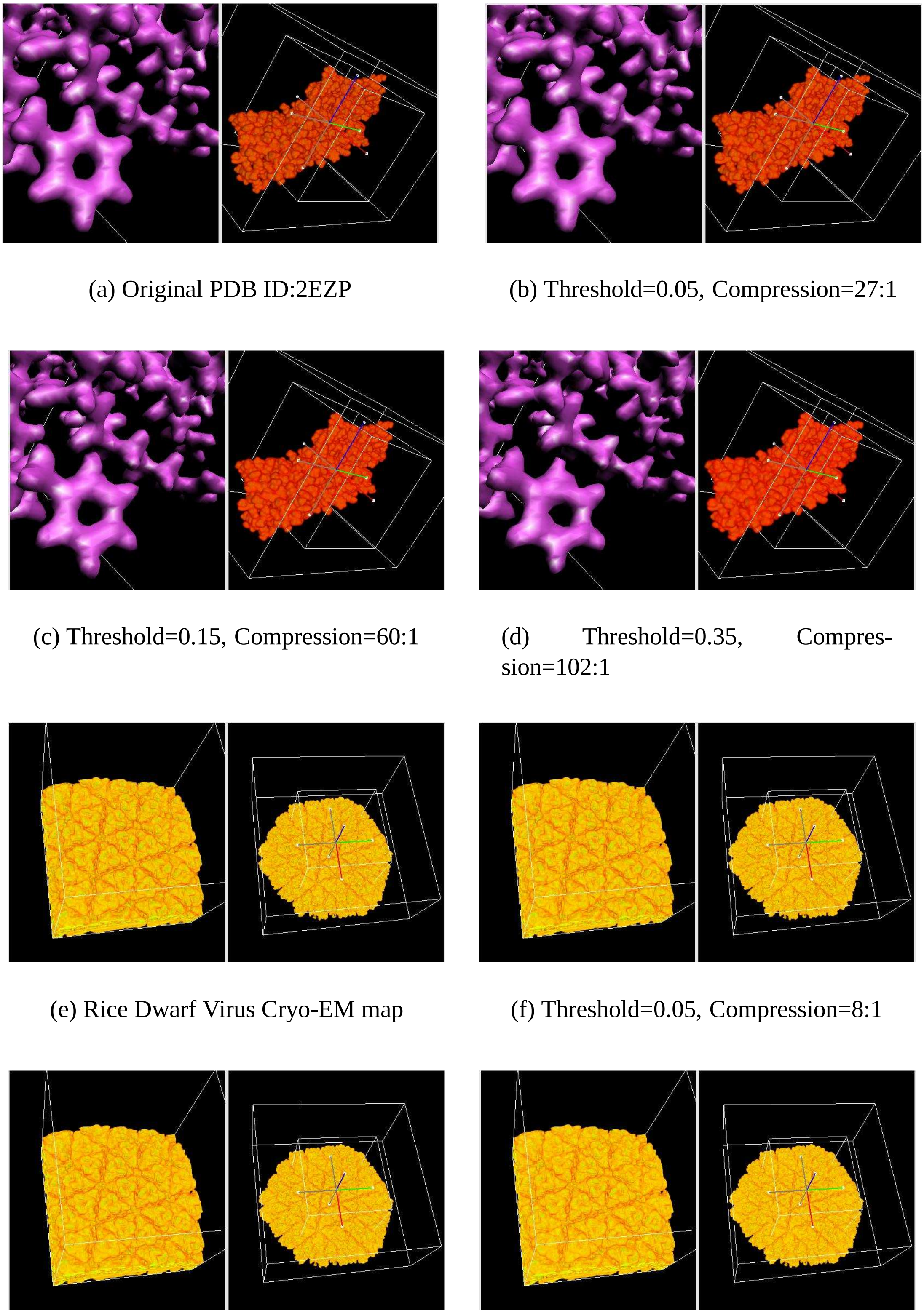
In Table 3(b) we show random access timing test cases for our method tested on the 50s Ribosomal large unit with varying rates of decay. The four test cases are

- t_1: The average time of 1000 random accesses of a voxel from the compressed format.
- t_2: The average time of 1000 random accesses of a full cell from the compressed format.
- t_3: The average time of 1000 random accesses of a voxel inside a cell, where the cell was chosen at random from the uncompressed format.
- t_4: The average time to decode the entire compressed volume for 10 runs.

The code was run on a SGI ONYX2 system with R12000 processors. The timing results are for one processor. As observed, to decode a single voxel the slowest case is t_1. However, for most volume rending and isocontouring applications it is more efficient to decode an entire cell since many neighboring points are used.

3 Conclusions

We have demonstrated the value of HB wavelet compressed volumetric LBC and AP representations, especially in the age of rapidly increasing structures and medium resolution maps in both the PDB and EMBI. The added value



(a) Original PDB ID:2EZP

(b) Threshold=0.05, Compression=27:1

(c) Threshold=0.15, Compression=60:1

(d) Threshold=0.35, Compression=102:1

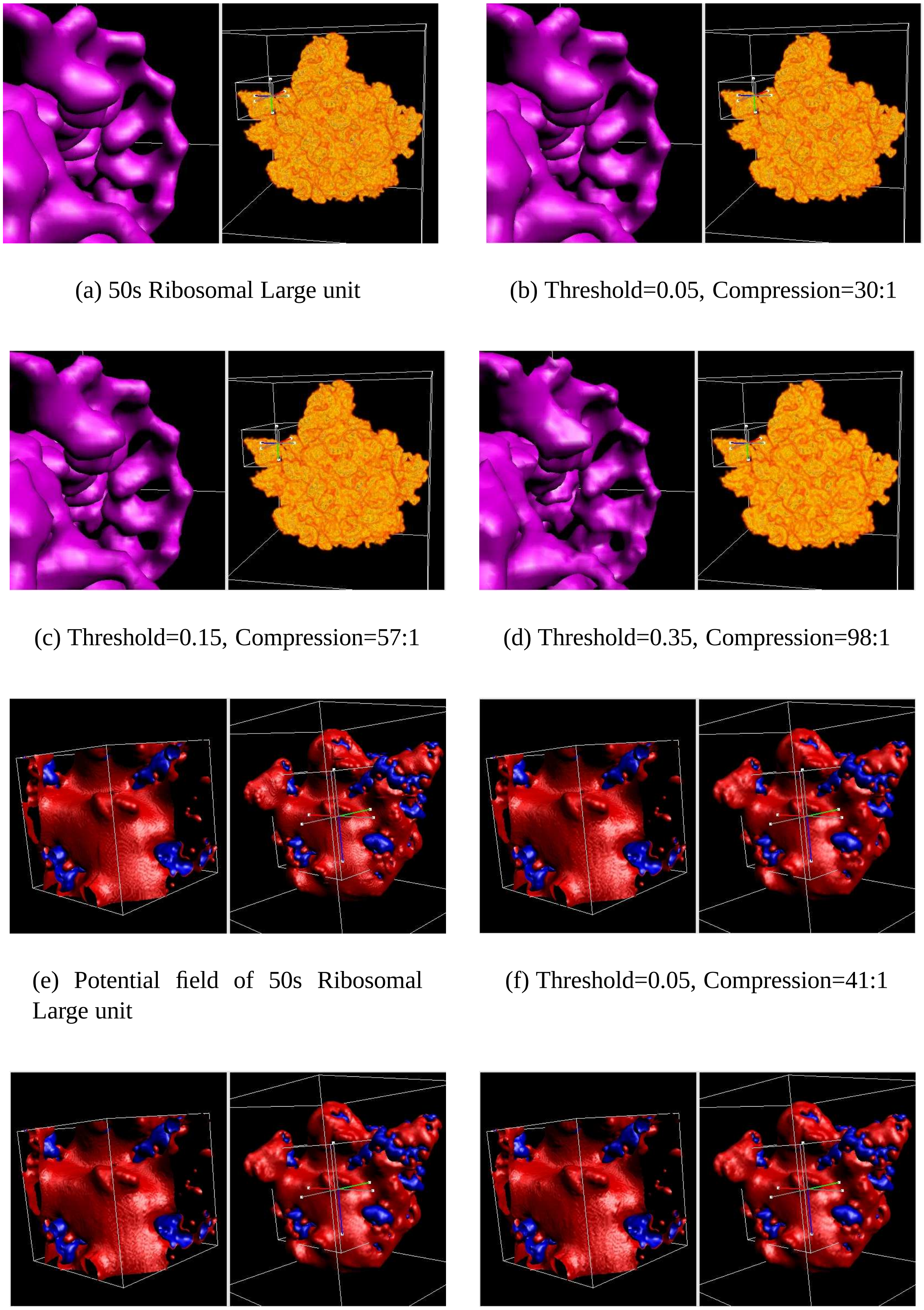
(e) Rice Dwarf Virus Cryo-EM map

(f) Threshold=0.05, Compression=8:1

(g) Threshold=0.15, Compression=22:1

(h) Threshold=0.35, Compression=37:1

Fig. 5. Compression examples of synthetic and natural molecular maps. (a)-(d) Electron density map of PDB 2EZP (Siv Gp41) with 3057 atoms in a 257^3 grid. (e)-(h) Rice Dwarf Virus Cryo-EM compressed maps in $241 * 481^2$ voxels.



(a) 50s Ribosomal Large unit

(b) Threshold=0.05, Compression=30:1

(c) Threshold=0.15, Compression=57:1

(d) Threshold=0.35, Compression=98:1

(e) Potential field of 50s Ribosomal Large unit

(f) Threshold=0.05, Compression=41:1

(g) Threshold=0.15, Compression=55:1

(h) Threshold=0.35, Compression=62:1

Fig. 6. Compression examples of synthetic and natural molecular maps.(a)-(d) Electron density map of 50s Ribosomal subunit (PDB:1JJ2) with 98543 atoms in 512^3 grid. (e)-(h) Potential field of 50s Ribosomal subunit in 513^3 grid created from the Adaptive Poisson Boltzmann Solver (*https://gridport.npaci.edu/apbs/*)

ϵ	PSNR	% 0's after TX	% 0's original	TCC	NZC	e_∞	e_2	Size(Bytes)	CR
0.50	32.73	99.50	86.23	201.65	27.76	51.56	23.47	539321	125.89
0.35	34.53	99.34	86.23	153.54	21.14	41.04	19.08	666472	101.87
0.15	38.65	98.80	86.23	83.56	11.50	18.93	11.87	1139150	59.60
0.05	45.19	97.20	86.23	35.73	4.92	6.53	5.59	2512429	27.02
0.01	60.65	93.19	86.23	14.70	2.02	1.26	0.94	5764216	11.77
0.00								67898440	

(a) 2EZP Electron density map

ϵ	PSNR	% 0's after TX	% 0's original	TCC	NZC	e_∞	e_2	Size(Bytes)	CR
0.500	38.50	98.84	0.58	85.91	85.40	63.01	3.48	8263922	64.96
0.35	40.24	98.78	0.58	82.39	81.91	45.67	2.85	8572631	62.62
0.15	44.18	98.60	0.58	71.85	71.43	19.75	1.81	9675507	55.48
0.05	50.29	98.06	0.58	51.56	51.26	6.28	0.90	12972064	41.38
0.01	60.36	96.13	0.58	25.85	25.70	1.29	0.28	24344104	22.05
0.00								536870980	

(b) Ribosome 50s Large Subunit Potential Map 513^3

ϵ	PSNR	% 0's after TX	% 0's original	TCC	NZC	e_∞	e_2	Size(Bytes)	CR
0.50	31.10	99.15	72.61	117.83	32.27	41.11	25.61	4771349	46.74
0.35	32.97	98.88	72.61	89.49	24.51	36.45	20.64	6042206	36.91
0.150	36.03	98.08	72.61	52.29	14.32	18.95	14.53	10031643	22.23
0.05	42.69	94.73	72.61	19.01	5.20	6.68	6.74	26590966	8.38
0.01	55.69	86.26	72.61	7.27	1.99	1.31	1.48	67075200	3.32
0.00								223032072	

(c) Rice Dwarf Virus Cryo-EM map (481*481*241)

Table 1
Zero Bit encoded HB Compression results for 2EZP electron density map, Large Ribosomal subunit 50s and a Cryo-EM map of the Rice Dwarf Virus. Legend: ϵ as defined in 2.1.5, PSNR=Peak Signal to noise ratio, TX: wavelet transform, TCC: Total Coefficient Compression ratio, NZC: Non Zero Compression ratio, e_∞:maximum l^∞ percent error between reconstructed compressed version and the original, e_2= l^2 percent error between reconstructed compressed version and the original. CR:File Compression Ratio.

of fast decompression time yields the necessary ingredient for enabling interactive visualization performance on current desktop computers. Our future work is predominantly in the development and analysis of visualization algorithms that work directly from the HB compressed volumes of bio-molecules and associated properties, with fast decompression. A related future direction of our research is in the development of visualization techniques which optimally utilize the faster bandwidth of programmable graphics hardware, commonplace on most desktop computers.

Another research avenue is to obtain a fully optimized error bounded scheme with HB wavelets. Moreover, a new basis is sought which combines spatially multi-resolution refinement and polynomial refinement to further optimize the compression vs accuracy ratio.

ϵ	PSNR	% 0's after TX	% 0's original	TCC	NZC	e_∞	e_2	Size(Bytes)	CR
0.50	30.15	99.10	77.67	111.73	24.94	54.13	27.84	6376816	84.19
0.35	31.87	98.81	77.67	84.07	18.77	43.37	22.82	8025803	66.89
0.15	36.13	97.73	77.67	44.13	9.85	19.09	13.97	14473038	37.09
0.05	42.52	94.71	77.67	18.91	4.22	6.63	6.63	32247352	16.64
0.01	58.49	87.39	77.67	7.93	1.77	1.30	1.04	72738554	7.38
0.00								536870980	

(a) $\mathcal{B}$=-5

ϵ	PSNR	% 0's after TX	% 0's original	TCC	NZC	e_∞	e_2	Size(Bytes)	CR
0.50	30.82	99.19	74.77	123.72	31.20	29.80	19.76	5807819	92.43
0.35	32.76	98.92	74.77	93.06	23.47	25.71	15.73	7315547	73.38
0.15	37.35	97.99	74.77	49.80	12.56	17.18	9.31	12469751	43.05
0.05	42.15	95.76	74.77	23.63	5.96	6.53	5.31	25731023	20.86
0.01	56.88	86.17	74.77	7.23	1.82	1.31	0.96	79575816	6.74
0.00								536870980	

(b) $\mathcal{B}$ = -2

ϵ	PSNR	% 0's after TX	% 0's original	TCC	NZC	e_∞	e_2	Size(Bytes)	CR
0.50	33.16	99.35	71.58	153.85	43.72	22.24	11.93	4586097	117.06
0.35	34.52	99.22	71.58	128.86	36.62	16.91	10.23	5460302	98.32
0.15	39.18	98.54	71.58	68.56	19.48	9.27	5.92	9432914	56.91
0.05	44.58	97.06	71.58	34.07	9.68	5.77	3.18	17706006	30.32
0.01	54.79	89.12	71.58	9.19	2.61	1.31	0.96	63375575	8.47
0.00								536870980	

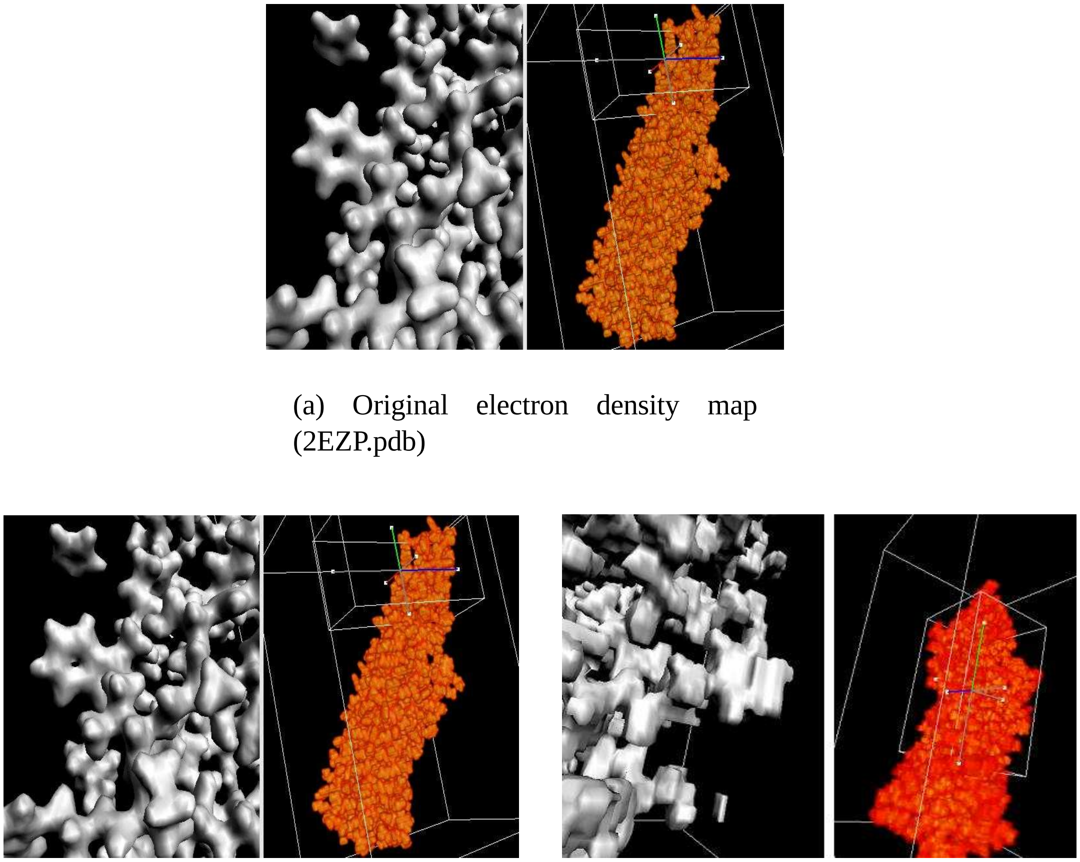
(c) $\mathcal{B}$ = -1

Zero-bit HB				Zero-bit Haar			
CR	PSNR	e_∞	e_2	CR	PSNR	e_∞	e_2
98.32	34.53	16.91	10.23	98.32	29.76	65.60	28.88
56.91	39.19	9.27	5.92	56.91	32.94	52.13	20.03
30.32	44.59	5.78	3.18	30.22	40.93	25.33	7.96

(d) Zero-bit HB and zero-bit encoded Haar compression

Table 2
Compression and comparison results for 50s Ribosomal Large unit with different Rate of Decay parameter $\mathcal{B}$.(a)-(c) Compression results with HB basis. (d) Compression results for 50s Ribosomal large unit in 513^3 grid with $\mathcal{B} = -1$ produced by zero-bit HB and zero-bit encoded Haar compression. It is clear from the results that for the same file compression ratios the zero-bit HB is about 5db higher that zero-bit HB, which is very significant at high compression ratios. Legend: ϵ as defined in 2.1.5, PSNR=Peak Signal to noise ratio, TX: wavelet transform, TCC: Total Coefficient Compression ratio, NZC: Non Zero Compression ratio, e_∞:maximum l^∞ percent error between reconstructed compressed version and the original, e_2= l^2 percent error between reconstructed compressed version and the original. CR:File Compression Ratio.

Appendix



(a) Original electron density map (2EZP.pdb)

(b) non zero-bit 207:1 coefficient compression using HB

(c) non zero-bit 207:1 coefficient compression using Haar

Fig. 7. Compression comparison between HB and Haar schemes. In (a) we have a view of the iso surface corresponding to the molecular surface of the original electron density map. By setting the compression ratio for both hierarchical methods to 207:1 we can readily see that HB leads to better feature preserving (as shown in (b)), in contrast to the poor performance of Haar (as shown in (c)).

A Error bounded compression

By eliminating only zero and *low valued* wavelets, we can balance compression and accuracy. We would like to ensure that the error due to the lossy compression is within a user defined threshold $\varepsilon > 0$. The following algorithm describes a top down approach to providing error bounded wavelet compression.

Algorithm 1: Top down error-guided wavelet compression

The maximum error $\varepsilon_T(C)$ between a function f and the trilinear approximation on C is given as.

level	$numbas$	$numrec$	$arith$
0 (V_0)	8	8	15
0 (W_0)	19	15	60
1 (W_1)	98	22	105
2 (W_2)	604	29	150
total	729		

(a)

Rate of Decay	t_1	t_2	t_3	t_4
$\mathcal{B} = -1$	5.049×10^{-5}	1.456×10^{-4}	2.58×10^{-6}	26.679
$\mathcal{B} = -2$	5.379×10^{-5}	1.427×10^{-4}	2.58×10^{-6}	25.835
$\mathcal{B} = -5$	5.404×10^{-5}	1.375×10^{-4}	2.58×10^{-6}	24.829

(b)

Table 3
(a) Worst case reconstruction costs for a Cell with three wavelet levels. Let $numbas$ be the number of basis functions at level j. In addition, let $numrec$ be the worst case decoding cost and $arith$ be the worst case arithmetic cost to decode a single voxel. (b) Reconstruction random access timings for the 50s Ribosomal large unit. The timings are in seconds. Legend: t_1= The average time of 1000 random accesses of a voxel from the compressed format. t_2= The average time of 1000 random accesses of a full cell from the compressed format. t_3= The average time of 1000 random accesses of a voxel inside a cell, where the cell was chosen at random from the uncompressed format. t_4= The average time to decode the entire compressed volume for 10 runs.

$$|\varepsilon_T(C)| \leq 2^{-2n-3} \left(\| \frac{\partial^2 f}{\partial x_1^2} \|_{\mathcal{L}^\infty(C)} + \| \frac{\partial^2 f}{\partial x_2^2} \|_{\mathcal{L}^\infty(C)} + \| \frac{\partial^2 f}{\partial x_3^2} \|_{\mathcal{L}^\infty(C)} \right) \tag{A.1}$$

Given a predefined maximum number of levels for the wavelets in a block, we perform an adaptive top down decomposition of the domain as follows:

(1) Let $i = 0$ correspond to the initial level where we have a single cube $C_0^1 \in C_0$. If $|\varepsilon_T(C_0^1)| \leq \varepsilon$, we are done. Else, we decompose and continue with the next steps.
(2) Decompose the current cube $C_i^k \in C_i$ into 8 new subcubes C_{i+1}^k, where $k \in \{1..8\}$.
(3) Obtain the wavelets associated with the nodes $M(i+1)$ that correspond to the cube C_{i+1}^k.
(4) For each subcube C_{i+1}^k, if the corresponding trilinear error $\varepsilon_T(C_{i+1}^k)$ is within the user defined error ε, we do not decompose C_{i+1}^k any further. Otherwise, we need to decompose the subcube, and continue in a hierarchical fashion with step 2.
(5) This error driven adaptive decomposition is continued till we reach the maximum level or the error is satisfied everywhere.

A subcube C_{i+1}^k with error $|\varepsilon_T(C_{i+1}^k)| \leq \varepsilon$ is not decomposed in the above algorithm. But if a neighboring subcube C_{i+1}^{j+k} is decomposed, then the shared node $m \in M(i+1)$ adds a new function, the compactly supported trilinear associated with a node at level $i+1$ centered at m, to the subcube C_{i+1}^k. This changes the function in C_{i+1}^k, which can potentially violate the requirement $|\varepsilon_T(C_{i+1}^k)| \leq \varepsilon$. Hence the above algorithm in fact does not satisfy the error bound required by the user in a global sense and is modified as shown below.

Let us consider a subcube C_l^k which is surrounded by cubes at level $i \geq 2$ on all its 6 faces. For each level, each face can be subdivided into 4 new faces. Each time, 5 new nodes are added per old face, 4 of which are shared by 2 faces and 1 is on the center (It should be noted that any cube can be surrounded by all sides only if it is defined after level 1. Hence a cube fully surrounded by fully subdivided cubes cannot have 8 new nodes at level 1). The total number of nodes that could surround a cube, given i levels is bounded by

$$\frac{4(6 \times 4^{i-2})}{2} + 6 \times 4^{i-2} = \frac{9}{8}4^i$$

Let $\varepsilon'_T(node, C_l^k)$ be the trilinear error in the cube containing $node$ (if $node$ is a corner of C_l^k). If, on the other hand, $node$ was a hanging face node for the cube C_l^k, let $\varepsilon'_T(node, C_l^k)$ represent the bilinear error associated with that face. This error $\varepsilon'_T(node, C_l^k)$, from equation A.1 is bounded as follows:

- Node is a corner vertex of C_l^k:

$$\varepsilon'_T(node, C_l^k) = \varepsilon_T(C_l^k)$$

- Node is a hanging node on a face C_l^k: Let the node be created due to the recursive subdivision of a face of C_l^k at some level $\hat{l} > l$. (The node itself is at level $\hat{l}+1$ and is created from a face subdividing at level $\hat{l}$.)

$$\varepsilon'_T(node, C_l^k) \leq |2^{-2*(\hat{l}-l)-3}\varepsilon_T(C_l^k)|$$

The error due to the new nodes on the faces of C_l^k is hence bounded by

$$\begin{aligned} \varepsilon_{nodes}(C_l^k) &= \sum_{k=l}^{L} \sum_{node=1}^{\frac{9}{8}4^k} \varepsilon'_T(node, C_l^k) \\ &\leq \left| 2^{-2l-3} \left(\| \frac{\partial^2 f}{\partial x_1^2} \|_{\mathcal{L}^\infty(C_l^k)} + \| \frac{\partial^2 f}{\partial x_2^2} \|_{\mathcal{L}^\infty(C_l^k)} + \| \frac{\partial^2 f}{\partial x_3^2} \|_{\mathcal{L}^\infty(C_l^k)} \right) \right. \\ &\quad \left. + \sum_{k=l}^{L} \frac{9}{8} \left(\| \frac{\partial^2 f}{\partial x_1^2} \|_{\mathcal{L}^\infty(face(C_l^k))} + \| \frac{\partial^2 f}{\partial x_2^2} \|_{\mathcal{L}^\infty(face(C_l^k))} + \| \frac{\partial^2 f}{\partial x_3^2} \|_{\mathcal{L}^\infty(face(C_l^k))} \right) \right| \end{aligned} \tag{A.2}$$

where $face(C_l^k))$ corresponds to the faces of C_l^k and L is the maximum number of levels.

Hence the algorithm is modified to continue to subdivide the subcube C_l^k to satisfy the error due to the possible addition of face nodes due to subdivision of its neighbors. For each new subdivision, we see a reduction in at least 4 in the error $\varepsilon_{nodes}(C_l^k)$ as given by equation A.1. Hence we need to subdivide a cube at the most C_l^k: $log_4(L-l)$ more levels after $|\varepsilon_T(C_l^k)| \leq \varepsilon$ is satisfied.

Algorithm 2: Top down error-bounded wavelet compression

Here we summarize the modified algorithm, which guarantees that the error in the wavelet compression is within the user defined error.

(1) Compute the second derivatives in the volume to help us calculate equation A.2.
(2) Let $i = 0$ correspond to the initial level where we have a single cube C_0. If $|\varepsilon_T(C_0)| \leq \varepsilon$ flag the cube C_0 and proceed to the last step. Else, we decompose and continue with the next 3 steps.
(3) Decompose the current cube C_i^k into 8 new subcubes C_{i+1}^k.
(4) Obtain the wavelets associated with the nodes $M(i+1)$ that correspond to the cube C_{i+1}^k.
(5) For each subcube C_{i+1}^k, if $|\varepsilon_T(C_{i+1}^k)| \leq \varepsilon$ flag the cube C_{i+1} and proceed to the last step. Otherwise, we need to decompose the subcube, and continue in a hierarchical fashion with step 3.
(6) This error driven adaptive decomposition is continued till we reach the maximum level or the error is satisfied everywhere.
(7) For each node not at the last level and hence flagged in the above steps, we subdivide further to satisfy the theoretical error due to addition of new functions from possible subdivision of neighbors. Hence for each flagged node C_i, we continue to subdivide till

$$|\varepsilon_{nodes}(C_i^k)| + |\varepsilon_T(C_i^k)| < \varepsilon$$

From the previous discussion, we also know that the above inequality is always satisfied if we subdivide $log_4(L - i)$ more levels. This step is placed separately from the previous steps just for clarity. It could be performed in step 2 and 5 itself.

Note that the cost of this algorithm is $\mathcal{O}(Nlog(N))$, where N is the total number of voxels in C.

Although we have a fully error bounded scheme, it is not yet optimal. In many cases we might over shoot the required accuracy. This algorithm will be further optimized to obtain a tighter bounded.

B Apriori Mesh Width Estimation and local Wavelet Updates for RBF Gaussian Maps

Most compression schemes require the computation of all the wavelets and then subsequently threshold them. Due to the size of the problems involved, with grid sizes well into the 256^3 voxel range, this can be a costly step. However, for a class of synthetic molecular maps such as RBF Gaussian summations, a-priori mesh size estimates and significant wavelet coefficients can be computed. In particular, we are interested in compression of data maps of the form

$$f = \sum_{i=1}^{M} c_i e^{-||\bar{x}_i||^2/B_i},$$

where $||\bar{x}_i||^2 = (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2$, M is the number of atoms and $B_i = (\mathcal{B}/R_i^2)^{-1}$, where $\mathcal{B}$ is the rate of decay and R_i is the radius of the i^{th} atom. Moreover, fast updates of the wavelets coefficients with respect to the rate of decay is desirable.

The source of error in the compression scheme is broken up into trilinear interpolation plus error associated with the wavelet. The trilinear error estimates provides a suitable mesh size to accurately represent the RBF map. The wavelet error estimates lead to the fast computation of the projection coefficients.

B.1 Mesh size estimation.

First, we assume that $B_{min} \leq B_i \leq B_{max}$ for all $i = 1 \ldots M$, and and let the ratio $Q = B_{max}/B_{min}$. Note that Q is $1 \ldots 9$, since the Van der Waals radius of the atoms in a protein vary between 1 to 3 angstroms. The second partial derivative of each summation term of $f(\bar{x}) = f(x, y, z)$ takes the form

$$\frac{\partial^2 e^{-||\bar{x}_i||^2/B_i}}{\partial x \partial x} = \frac{2}{B_i} e^{-||\bar{x}_i||^2/B_i} \left(\frac{2(x - x_i)^2}{B_i} - 1 \right),$$

where $\bar{x}_i = (x - x_i, y - y_i, z - z_i)$ for each Gaussian centered around (x_i, y_i, z_i). The minimum critical point is located at $\bar{x} = (x_i, y_i, z_i)$ with a value of $-\frac{2}{B_i}$. The maximum critical point is located at $\bar{x} = (x_i \pm \sqrt{\frac{3B_i}{2}}, y_i, z_i)$ with a value of $\frac{4}{B_i} e^{-3/2}$, then

$$\begin{aligned}
|\varepsilon| &\leq \frac{h^2}{8} \left(\|\frac{\partial^2 f}{\partial x^2}\|_\infty + \|\frac{\partial^2 f}{\partial y^2}\|_\infty + \|\frac{\partial^2 f}{\partial z^2}\|_\infty \right) \\
&\leq \frac{h^2}{4B_{min}} \sum_{i=1}^{M} \left(\|c_i e^{-||\bar{x}_i||^2/B_i} (\frac{2}{B_i}(x - x_i)^2 - 1))\|_\infty \right. \\
&+ \|c_i e^{-||\bar{x}_i||^2/B_i} (\frac{2}{B_i}(y - y_i)^2 - 1))\|_\infty \\
&+ \left. \|c_i e^{-||\bar{x}_i||^2/B_i} (\frac{2}{B_i}(z - z_i)^2 - 1))\|_\infty \right).
\end{aligned}$$

Now, let M_r be the maximum number of expansion points in a sphere $\mathbb{S} \subset \mathbb{R}^3$ of radius $x_r = 9\sqrt{B_{max}}$ around a desired point (x, y, z), then

$$\begin{aligned}
\| \sum_{i=1}^{M} c_i e^{-||\bar{x}_i||^2/B_i} (\frac{2}{B_i}(x - x_i)^2 - 1))\|_\infty &\leq \| \sum_{i=1}^{M_r} |c_i| \frac{2}{B_{min}} + \sum_{i=M_r+1}^{M} |c_i| \|e^{-||\bar{x}_i||/B_{min}} (\frac{2(x - x_i)^2}{B_{min}} - 1)\|_{l^\infty(\mathbb{R}^3/\mathbb{S})} \\
&\leq \| \sum_{i=1}^{M_r} |c_i| \frac{2}{B_{min}} + \sum_{i=M_r+1}^{M} |c_i| e^{-x_r^2/B_{min}} (\frac{2x_r^2}{B_{min}} - 1)
\end{aligned}$$

$$\leq \|\bar{c}_r\|_1 \frac{2}{B_{min}} + \|\bar{c}\|_1 e^{-x_r^2/B_{min}} (\frac{2x_r^2}{B_{min}} - 1),$$

where $\|\bar{c}\|_1 = \sum_{i=1}^{M} |c_i|$ and $\|\bar{c}_r\|_1 = \sum_{i=1}^{M_r} |c_i|$. The second term on the right hand side becomes less than $10^{-33}\|\bar{c}\|$. This is below machine precision for molecules up to 10^{16} atoms, thus it can be safely ignored. This implies then that

$$|\varepsilon| \leq h^2 \left(\|\bar{c}_r\|_1 \frac{3}{2B_{min}^2} \right) \leq \epsilon.$$

where the desired accuracy is ϵ. By setting the mesh size

$$h = B_{min} \sqrt{\frac{2\epsilon}{3\|\bar{c}_r\|_1}},$$

then the absolute error is lower than ϵ.

B.2 *Fast Updates of Mesh size and Wavelet coefficients with Respect to Changes in the Rate of Decay.*

As we increase the rate of decay B_i we can easily update the grid size to maintain the same accuracy. Suppose that B_{min} increases to $B_{min}^{'}$, then

$$h^{'} = h \frac{B_{min}^{'}}{B_{min}} \sqrt{\frac{\|\bar{c}_r\|_1}{\|\bar{c}_r^{'}\|_1}}.$$

A fast update of the wavelet coefficients can also be achieved with increasing (or decreasing) rate of decay. In particular we provide bounds on the wavelet coefficients for which the error is below a predetermined threshold. For each wavelet $\psi_{j,m}$ with support bounded by 2^{-j} along each dimension

$$|\psi_{j,m}| \leq \| \frac{2^{-2j-3}}{B_{min}} \sum_{i=1}^{M} c_i e^{-\|\bar{x}_i\|^2/B_{min}} \left(\frac{2}{B_{min}} ((x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2) - 3 \right) \|_{\infty}.$$

We are now interested in bounding the magnitude of the the wavelet coefficient. For each vertex $\bar{x}_i$ any wavelet support that intersects with a sphere of radius $\sqrt{\frac{3B_{max}}{2}}$ is computed if

$$|\psi_{j,m}| \leq \frac{2^{-2j-3}}{B_{min}^2} \sum_{i \in M_{j,m}} |c_i| \leq \epsilon_j,$$

where $M_{j,m}$ is an index set for all the vertices $\bar{x}_i$ such that $d(\psi_{j,m}, \bar{x}_i) \leq \sqrt{\frac{3B_{max}}{2}}$, where

$$d(\psi_{j,m}, \bar{x}_i) = \min_{\bar{x} \in suppt(\psi_{j,m})} \|x - \bar{x}_i\|_2.$$

Now, for wavelets such that $d(\psi_{j,m}, \bar{x}_i) > \sqrt{\frac{3B_{max}}{2}}$, pick the minimum distance r_{min} from the support of the wavelet $\psi_{j,m}$ to the closest Gaussian, then

$$|\psi_{j,m}| \leq \frac{2^{-2j-3}}{B_{min}} \|\bar{c}\|_1 e^{-r_{min}^2/B_{min}} \left(\frac{2}{B_{min}} r_{min}^2 - 3 \right).$$

We now choose the wavelets which are bounded by ϵ_j, then

$$|\psi_{j,m}| \leq \frac{2^{-2j-3}}{B_{min}} \|\bar{c}\|_1 e^{-r_{min}^2/B_{min}} \left(\frac{2}{B_{min}} r_{min}^2 - 3 \right) \leq \epsilon_j.$$

This condition is satisfied if

$$r_{min} \geq \max \left(\sqrt{-B_{min} W \left(-\frac{\epsilon_j B_{min}}{2^{-2j-2} \|\bar{c}\|_1} \right)}, -\sqrt{-B_{min} W \left(-\frac{\epsilon_j B_{min}}{2^{-2j-2} \|\bar{c}\|_1} \right)} \right)$$

where W is the LambertW function. Thus only the T wavelets that are contained in the radius r_{min} from any Gaussian center with vertex x_i is computed. Note that they can be computed in $\mathcal{O}(M+T)$ time, by using the *Fast Gauss Transform*. (Potts et al. (2004), Beatson & Greengard (1997), Baxter & Roussos (2002)).

Acknowledgements

We are very grateful to the National Science Foundation and National Institutes of Health for supporting this research under NSF grants INT-9987409, ACI-022003, EIA-0325550 and a grant from the NIH 0P20 RR020647. Moreover, we appreciate very much the help from Dr. Wah Chiu's group, Baylor School of Medicine. He was kind enough to provide us with with many Cryo-EM maps, in particular the Rice Dwarf Virus. Finally, we mention that to our knowledge we do not have any financial conflict that can influence the results or interpretation of this manuscript.

References

Akkiraju, N. & Edelsbrunner, H. (1996), 'Triangulating the surface of a molecule', *Discrete Applied Mathematics* **71**, 5–22.

Amaratunga, K. & Castrillón-Candás, J. E. (2001), 'Surface wavelets: A multiresolution signal processing tool for 3d computational modeling', *International Journal of Numerical Methods in Engineering* **52**, 239–271.

Bajaj, C., Ihm, I. & Park, S. (2000), 'Compression-based 3d texture mapping for real-time rendering', *Graphical Models* **62**(6), 391–410.

Bajaj, C., Ihm, I. & Park, S. (2001*a*), '3d rgb image compression for interactive applications', *ACM Transactions on Graphics* **20**(1), 10–38.

Bajaj, C., Ihm, I. & Park, S. (2001*b*), Visualization-specific compression of large volume data, *in* 'Proc. of Pacific Graphics', Tokyo, Japan, pp. 212–222.

Bajaj, C., Ihm, I., Park, S. & Song, D. (2000), Compression-based ray casting of very large volume data in distributed environments, *in* 'HPC-Asia 2000', Beijing, China, pp. 720–725.

Bajaj, C. L., Lee, H. Y., Merkert, R. & Pascucci, V. (1997), NURBS based B-rep models for macromolecules and their properties, *in* 'Proceedings of the 4th Symposium on Solid Modeling and Applications', ACM Press, New York, pp. 217–228.

Bajaj, C. L., Pascucci, V. & Schikore, D. (1996), Fast isocontouring for improved interactivity, *in* 'Preceedings of the 1996 Symposium for Volume Visualization'.

Bajaj, C. L., Sohn, B. & Shamir, A. (2001), *Isocontour Tracking in Time-Varying Data*, Technical Report, Department of Computer Sciences.

Bajaj, C., Pascucci, V., Shamir, A., Holt, R. & Netravali, A. (2001), 'Dynamic maintenance and visualization of molecular surfaces', *Discrete Applied Mathematics, Fourth issue in the special series of on Computational Molecular Biology* **104**(4).

Bajaj, C., Pascucci, V. & Zhuang, G. (1999*a*), Progressive compression and transmission of arbitrary triangular meshes, *in* 'Proceedings of IEEE Visualization Conference', San Francisco, CA, pp. 307–316.

Bajaj, C., Pascucci, V. & Zhuang, G. (1999*b*), Single resolution compression of arbitrary triangular meshes with properties, *in* 'IEEE Data Compression Conference', pp. 247–256.

Baker, N., Spet, D., Joseph, S., Holst, M. & McCammon, J. (2001), 'Electrostatics of nanosystems: application to microtubules and the ribosome', *Proc. Natl. Acad. Sci. USA* **98**, 10037–10041.

Baxter, B. J. C. & Roussos, G. (2002), A new error estimate of the fast gauss transform, *in* 'SIAM Journal on Scientific Computing', Vol. 24, pp. 257–259.

Beatson, R. K. & Greengard, L. (1997), A short course on fast multipole methods, *in* 'Wavelets, Multilevel Methods and Elliptic PDEs (M. Ainsworth, J. Levesley, W. Light, and M. Marletta, eds.)', Oxford University Press, pp. 1–37.

Blinn, J. F. (1982), 'A generalization of algebraic surface drawing', *ACM Transactions on Graphics* **1**(3), 235–256.

Boys, S. (1950), 'Electronic wave functions. i, a general method of calculation for the stationary states of any molecular system', *Proceedings of the Royal Society of London, Series A, Mathematical and Physical Sciences* **200**(1063), 542–554.

Castrillón-Candás, J. (2001), *Spatially Adaptive Multiwavelet Representations on Unstructured Grids with Applications to Multidimensional Computational Modeling*, Ph.D. thesis, MIT.

Castrillón-Candás, J. & Amaratunga, K. (2003), 'Spatially adapted multiwavelets and sparse representation of integral operators on general geometries', *SIAM Journal on Scientific Computing* **24**(5), 1530–1566.

Connolly, M. L. (1983*a*), 'Analytical molecular surface calculation', *Journal of Applied Crystallography* **16**, 548–558.

Connolly, M. L. (1983*b*), ‘Solvent-accessible surfaces of proteins and nucleic acids’, *Science* **221**, 709–713.

Daubechies, I. (1998), ‘Orthonormal bases of compactly supported wavelets’, *Comm. Pure Appl. Math.* **41**, 909–996.

Duncan, B. S. & Olson, A. J. (1993*a*), ‘Approximation and characterization of molecular surfaces’, *Biopolymers* **33**, 219–229.

Duncan, B. S. & Olson, A. J. (1993*b*), ‘Shape analysis of molecular surfaces’, *Biopolymers* **33**, 231–238.

Grant, J. & Pickup, B. (1995), ‘A gaussian description of molecular shapes’, *J. Phys. Chem.* **99**, 3503–3510.

Koide, A., Doi, A. & Kajioka, K. (1986), ‘Polyhedral approximation approach to molecular orbital graphics’, *Journal of Molecular Graphics* **4**, 149–155.

Leach, A. R. (1996), *Molecular Modelling: Principles and Applications*, Addison Wesley.

Lee, B. & Richards, F. M. (1971), ‘The interpretation of protein structures: Estimation of static accessibility’, *J. Mol. Biol* **55**, 379–400.

Lee, M., Geig, M., Salsbury, F. & Brooks, C. (2003), ‘New analytic approximation to the standard molecular volume definition and its application to generalized born calculations’, *J. of Computational Chemistry* **24**(11), 1348–1356.

Leicester, S. E., Finney, J. L. & Bywater, R. P. (1988), ‘Description of molecular surface shape using fourier descriptors’, *J. Mol. Graphics* **6**, 104–108.

Livnat, Y., Shen, H.-W. & Johnson, C. R. (1996), ‘A Near Optimal Isosurface Extraction Algorithm Using the Span Space’, *IEEE Transactions on Visualization and Computer Graphics* **2**(1), 73–84.
URL: *citeseer.nj.nec.com/livnat96near.html*

Lorensen, W. E. & Cline, H. E. (1987), Marching cubes: A high resolution 3D surface construction algorithm, Vol. 21, pp. 163–169.

Max, N. L. (1983), ‘Computer representation of molecular surfaces’, *IEEE Computer Graphics and Applications* **3**(5), 21–29.

Max, N. L. & Getzoff, E. D. (1988), ‘Spherical harmonic molecular surfaces’, *IEEE Computer Graphics & Applications* **8**, 42–50.

Mestres, J., Rohrer, D. & Maggiora, G. (1997), ‘Mimic: A molecular -field matching program. exploiting applicability if molecular similarity approaches’, *Journal of Computational Chemistry* **18**(7), 934–.

Potts, D., Steidl, G. & Nieslony, A. (2004), ‘Fast convolution with radial kernels at nonequispaced knots’, *Numer. Math.* **98**(2), 329–351.

Ritchie, D. W. & Kemp, G. J. (1999*a*), *Protein Docking Using Spherical Polar Fourier Correlations*, Proteins: Structure, Function & Genetics, John Wiley & Sons.

Ritchie, D. W. & Kemp, G. J. L. (1999*b*), ‘Fast computation, rotation, and comparison of low resolution spherical harmonic molecular surfaces’, *Journal of Computational Chemistry* **20**(4), 383–395.

Sanner, M. F. & Olson, A. J. (1996), ‘Reduced surface: An efficient way to compute molecule surfaces’, *Biopolymers* **38**, 305–320.

Sanner, M. F., Olson, A. J. & Spehner, J. C. (1995), Fast and robust computation of molecular surfaces, *in* ‘Proc. 11th Annual ACM Symposium on Computational Geometry’, pp. C6–C7.

Shen, H.-W. (1998), Isosurface extraction in time-varying fields using a temporal hierarchical index tree, *in* ‘IEEE Proceedings of Visualization ’98’, pp. 159–166.

Strang, G. & Nguyen, T. (1996), *Wavelets and Filter Banks*, Wellesley, MA: Wellesley-Cambridge Press.

Sutton, P. M. & Hansen, C. D. (2000), ‘Accelerated isosurface extraction in time-varying fields’, *IEEE Transactions on Visualization and Computer Graphics* **6**(2), 98–107.

Sweldens, W. (1996), ‘The lifting scheme: A custom-design construction of biorthogonal wavelets’, *Appl. Comput. Harmon. Anal.* **3**(2), 186–200.

Sweldens, W. (1998), ‘The lifting scheme: A construction of second generation wavelets’, *SIAM J. Math. Anal.* **29**(2), 511–546.

Taubman, D. S. & Marcellin, M. W. (2001), *JPEG 2000: Image Compression Fundamentals, Standards and P*, Norwell, MA: Kluwer Academic Publishers.

Varshney, A. (1994), Hierarchical Geometric Approximations, PhD thesis, Dept. of CS, U. of North Carolina, Chapel Hill. TR-050.

Varshney, A. & Brooks, Jr., F. P. (1993), Fast analytical computation of Richard’s smooth molecular surface, *in* G. M. Nielson & D. Bergeron, eds, ‘Proceedings of the Visualization ’93 Conference’, IEEE Computer Society Press, San Jose, CA, pp. 300–307.

Voorintholt, R., Kosters, M. T. & Vegter, G. (1989), ‘A very fast program for visualizing protein surfaces, channels and cavities’, *Journal on Molecular Graphics* **7**, 243–245.

Wilhelms, J. & Gelder, A. V. (1992), ‘Octrees for faster isosurface generation’, *ACM Transactions on Graphics* **11**(3), 201–227.

Yserentant, H. (1992), ‘Hierarchical bases’, *ICIAM 91, R.E. Ed., SIAM Philadelphia* **49**, 329–412.