Topological Approximation and Compression of Functions and Their Wavelet Expansions

William Lynch, Krasimir Kolarov
lynch@interval.com, kolarov@interval.com
Interval Research Corporation, 1801 Page Mill Road, Bldg.C, Palo Alto, CA 94304

1. INTRODUCTION

We will present a technique to compress functions defined on high dimensional manifolds. Many conventional data compression technologies, unmodified, are not suitable for compression of data defined on more complex geometry such as spheres, general polytopes, etc. In [3] we have described a method for representation and compression for addressing 2-manifold domains using second generation wavelet transforms and zerotree coding. In this paper we will generalize the approximation domain for scalar functions described in [3]. In particular we will describe the transition from 2-manifolds to n-manifolds (2-simplices to n-cells), from mid-point subdivision to dual-intersection subdivision. We will do that while retaining finite stencils of support for wavelet multi-resolution analysis and preserving the compression techniques.

From a mathematical point of view, we will introduce in Section 2 the foundations for a computational approximation theory for CW complexes. In developing those we have used material described in [4] among others. Next, we will address the question: To what extent can approximations of real valued functions defined on curves, surfaces, manifolds, and other Euclidean type spaces, be effectively and computationally carried out by purely topological means?

This question is approached by considering domains that are homeomorphic to finite CW complexes, which tessellate a locally Euclidean space into a finite number of cells of various dimensions. Associated with these cell complexes are the chain complex and boundary homeomorphism that describe how the cells are linked together. We can build *regular* CW complexes where the boundary operator can be easily represented and computed.

We use cells rather simplices (which have been used previously in multiresolution approaches) because cell constructions typically require fewer cells than the equivalent number of simplices. That leads to smaller data structures and quicker computations. We can also build better subdivision methods and there is no need for complicated fix-ups upon partial or adaptive subdivision. Related approaches for cells were use in [1] and [5]

Approximation is accomplished by the iterated application of a subdivision operator which can eventually separate any two points into the interiors of separate cells. Algebraically, a subdivision operator is a 1:1 chain homomorphism from the CW complex to another CW complex generated by the result of the cell partitioning. The regularity of the CW complex also enables the calculation of the boundary operator in the subdivided complex. We can use a finite sequence of finite regular CW complexes to approximate continuous functions, to build and compress wavelet expansions of such functions.

In Section 3 we will describe the data structures and a library of C++ routines that we built intended to represent topological objects. Section 4 illustrates our approach and outlines some further development issues.

2. MATHEMATICAL FOUNDATIONS

2.1 Some relevant definitions from topology

We will start by introducing few key mathematical concepts with their definitions.

Definition 1: Let $\{p_0, p_1, \dots p_m\}$ be an affine independent subset of $\Re^n$. The convex set spanned by this set, denoted $[p_0, p_1, \dots p_m]$, is called the **(affine) m-simplex** with vertices $p_0, p_1, \dots p_m$.

Definition 2: Two sets X and Y are said to be **homeomorphic** if their exists a continuous bijection $f: X \to Y$ with continuous inverse. Such a map f is called a **homeomorphism**.

Definition 3: A subset e of a topological space X is called an (*open*) **n-cell** if it is homeomorphic to the open disk $D^n - S^{n-1} \subset \Re^n$.

Definition 4: An n-dimensional **manifold** M is a Hausdorff set such that every point $m \in M$ has a neighborhood homeomorphic to $D^n - S^{n-1}$.

Definition 5: A continuous map $f:(A,B) \to (X,Y)$ is a **relative homeomorphism** if the map $f|_{A-B}: A-B \to X-Y$ is a homeomorphism.

Definition 6: Let a topological space X be the disjoint union of a collection of cells, $X = \coprod_{e \in E} e$. Then for each $k \geq 0$, the **k-skeleton** of X is defined by $X^{(k)} = \coprod_{e \in E:\dim(e) \leq k} e$, i.e. it consists of all of the cells of dimension *k* or less (see Fig. 1).

Definition 7: A **CW complex** is an ordered triple (X, E, Φ), where X is a Hausdorff space, E is a family of cells in X, and $\Phi = \{\Phi_e : e \in E\}$ is a family of maps such that:

- $X = \coprod_{e \in E} e$;
- for each k-cell $e \in E$, the map $\Phi_e: (D^k, S^{k-1}) \to \left(e \coprod X^{(k-1)}, X^{(k-1)}\right)$ is a relative homeomorphism;
- if $e \in E$ then its closure $\bar{e}$ is contained in a finite union of cells in E;
- X has the weak topology determined by $\{\bar{e} : e \in E\}$

The "CW" stands for ***C***losure–finite in the ***W***eak topology.

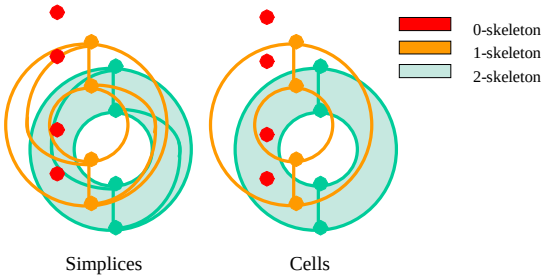


Figure 1. Skeletons for cells and simplices

2.2 Regular CW complexes and manifolds

A finite **regular CW complex** (or RCW for short) K, is a finite collection of elements called cells. Each cell is assigned a non-negative dimension n (making it an n-cell) and is homeomorphic to the unit open ball in Euclidean n-space. An **n-manifold** is a topological space which is locally homeomorphic to an n-cell. A compact manifold may be tessellated as a regular CW complex. The requirement that such a tessellation form a regular CW complex is that the inclusion map of each k-cell into the n-manifold be extendible to a homeomorphism of the closure of the k-cell (i.e., a closed k-ball) into the n-manifold. That is, each k-cell must embed in the manifold in such a way that its boundary embeds as a k-1-sphere.

$$K_n = \{\alpha | \alpha \text{ is an } n-\text{cell}\}$$
$$K = \bigcup_n K_n$$

As a notational matter, if α is a k-cell, then $\overline{\alpha}$ is its topological closure and $\dot{\overline{\alpha}}$ is its topological boundary. Note that the closure is the disjoint union of the cell and its boundary.

We will also be interested in various normed function spaces whose elements are functions from an n-manifold to the reals (or some other Clifford algebra). Additionally, we will be interested in operators from a function space over an RCW^D to a function space over an RCW^R. We will consider such operators to be in a function space over the RCW^{DxR}, the direct product of *D* and *R*.

We define

$$D \times R = \{(\alpha,\beta) | \alpha \in D, \beta \in R\}$$
$$\overline{(\alpha,\beta)} = \overline{\alpha} \times \overline{\beta}$$
$$\dot{\overline{(\alpha,\beta)}} = \overline{(\alpha,\beta)} - (\alpha,\beta)$$

It is clear that if $\alpha \in D_n$ is an n-cell and $\beta \in R_m$ is an m-cell then $(\alpha,\beta) \in (D \times R)_{n+m}$ is an n+m-cell. Less obvious but still true is that $\dot{\overline{(\alpha,\beta)}}$ is homeomorphic to an n+m-1-sphere. This latter fact allows a proof that $D \times R$ is an RCW.

2.3 Faces, Stars and the Boundary Operator

Definition 8: e_1 is a **face** of e_m iff there is a sequence of cells $e_1, e_2, \ldots, e_m$ with $e_i \subseteq \partial(e_{i+1})$

Definition 9: The **star** of a cell *e* is the sum of all cells *a* where *a* is a face of *e*. Stars give the basis for a topology, which in turn "glues" the space together.

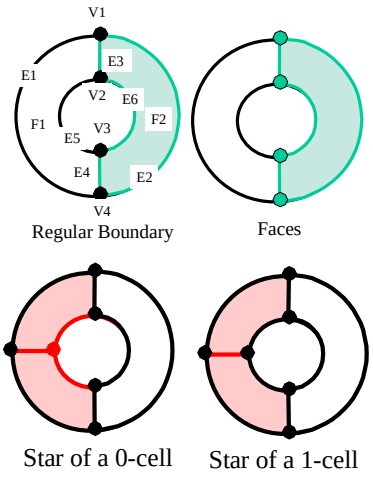


Figure 2. Faces and stars of a cell.

$C_k(\mathbf{X}, \mathbf{Z}_2)$ is a vector space with the *k*-cells as unit vectors and the integers mod 2 as coefficients. The direct product $C_k(\mathbf{X}, \mathbf{Z}_2)$ across k gives the **chain space** of $C(\mathbf{X}, \mathbf{Z}_2)$. The elements of a chain space are called **chains**. We will denote a chain *a* in the subspace generated by a chain *b* by $a \subseteq b$.

A **chain complex** $C(K)$ is a free module over a ring G with a distinguished nil-potent homomorphism ∂ called the **boundary operator**. Nilpotence means that

$$\partial\partial = 0$$

The Abelian group of $C(K)$ is generated by the cells of some RCW *K*. Thus, we can think of the elements of $C(K)$ as linear combinations

$$\sum_{\alpha \in K} c_\alpha \mathbf{i}_\alpha \quad c_\alpha \in G \text{ of the cells } \alpha \text{ of } K.$$

As for direct products we have the obvious

$$C(K_1 \times K_2) = \left\{ \sum_{(\alpha,\beta) \in K_1 \times K_2} c_{(\alpha,\beta)} \mathbf{i}_{(\alpha,\beta)} \right\}$$

so that we have the usual situation that the chain complex of the direct product of the RCWs is the tensor product of the chain complexes. It is easy to verify that

$$\partial\partial((\alpha,\beta)) = 0$$

We can also consider the boundary operator $\partial_k : C_k(\mathbf{X}, \mathbf{Z}_2) \rightarrow C_{k-1}(\mathbf{X}, \mathbf{Z}_2)$ as a distinguished endomorphism. For a cell it can be defined as: $\partial_k (\mathbf{e}_k) = \Sigma[e_k, e_{k-1}]\, \mathbf{e}_{k-1}$ where $[e_k, e_{k-1}]$ is in $\mathbf{Z}_2$ and $[e_k, e_{k-1}] = 1$ if e_{k-1} is on the boundary of e_k and is equal to zero otherwise. In that case $\partial_{k-1} \partial_k = 0$, i.e. the boundary of the boundary is zero.

2.4 Topological and algebraic subdivisions

In the subdivision process, we think of an RCW (or sub-RCW) K as being subdivided into partition pieces. We often write ΣK for the set of cells into which K has been partitioned.

In the case that K is a cell, we expect that the cells of $\dot{\Sigma}\dot{K}$, the boundary cells of ΣK, already are in a subdivided form so that geometrically, as sets of cells

$$\dot{\Sigma}\dot{K} = \Sigma\dot{K}$$

This requires that, topologically, **the boundary of the subdivision is the subdivision of the boundary**. Note that, in effect this rule requires that subdivisions proceed from lower dimensions to higher dimensions, with the boundary of a cell being acceptably subdivided before the cell itself is subdivided. As for direct products we have

$$\Sigma(K_1 \times K_2) = (\Sigma K_1) \times (\Sigma K_2)$$

The topological subdivision process generates a new chain complex $C(\Sigma K)$. It is immediately clear that there is a 1:1 into algebraic homomorphism

$$\Sigma : C(K) \rightarrow C(\Sigma K)$$

$$\Sigma\left(\sum_{\alpha \in K} c_\alpha \mathbf{i}_\alpha\right) = \left(\sum_{\alpha \in K} c_\alpha \sum_{\beta \in \Sigma\alpha} \mathbf{i}_\beta\right)$$

The overloading of the symbol Σ for both topological and algebraic subdivision is unfortunate but of long-standing establishment. In the algebraic domain the rule of "the subdivision of the boundary is the boundary of the subdivision" becomes a lot crisper:

$$\Sigma\partial = \partial\Sigma$$

There are a number of different schemes for performing the actual cell subdivision. One example is the barycentric subdivision (see Fig. 3) which subdivides an m-cell into a (central) 0-cell and a k+1 cell for each k-cell of the subdivided boundary of the m-cell. Each of the k-cells in the subdivision have k+1 0-cells on their boundary and is a k-dimensional generalization of the triangle (called a **simplex** - a 2-simplex is a triangle, a 3-simplex is a pyramid). Note that the barycentric subdivision of a triangle has 13 cells (the central 0-cell, six 1-cells from the six vertices of the subdivided boundary, and six 2-cells from the six edges of the boundary).

Another novel scheme for the actual cell subdivision is **bi-conic** subdivision. Bi-conic subdivision subdivides an m-cell into a (central) 0-cell and a m-k cell for each k-cell of the un-subdivided boundary of the m-cell. Each of the k-cells in the subdivision have at least k+1 0-cells on their boundary. A 2-cell in the subdivision will be a quadrilateral so that the bi-conic scheme is different from the barycentric scheme.

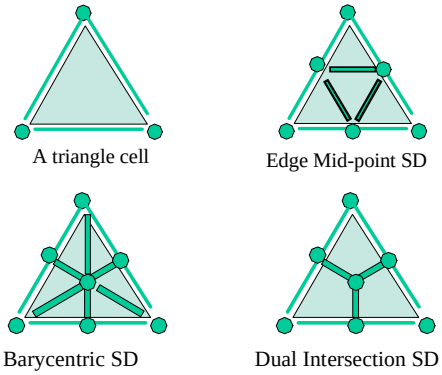


Figure 3. Examples of different subdivision schemes.

In [3] we used the edge mid-point subdivision which is common in computer graphics.

3. IMPLEMENTATION OF THE TOPOLOGY WITH DATA STRUCTURES

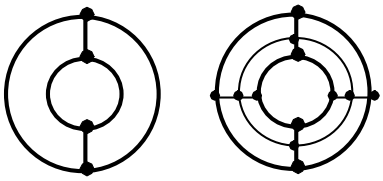
A regular CW complex enables effective computation and is easily achieved with one subdivision. In this section we will describe the data structures we build to model the topological entities described above. We will remember that the boundary of each *k*-cell is a *k*-1 sphere and the regularity of the CW complex allows lifting to the chain group over the integers.

There are a finite number of cells in any approximation. Cells are objects - there is one object for each cell. The dimension of the cell is stored with the cell. The values of any functions defined on the cells of the CW complex are stored with the cell. A chain is an object and is represented by a linked list of pointers to those cells of the chain with non-zero coefficients. (The coefficient may be recorded.) Chain operator methods are implemented in terms of coefficient arithmetic. The boundary operator is representable by one finite matrix for each dimension. The coefficients are 0 or 1. The boundary operator is sparse. The value of the boundary operator (a chain object) at a cell is stored with that cell object. The value (a chain object) of the adjoint of the boundary operator at a cell is stored with that cell object. The adjoint is a chain representing all of the cells for which the given cell is on the boundary. The adjoint, while redundant from an information point of view, allows for rapid calculation of stars.

The value (again a chain object) of the subdivision operator at a cell is stored with that cell object. Each cell not in the original CW complex has a **parent** pointer to the cell whose subdivision created it. A manifold is represented as just another chain. There is an ASCII file format for constructing, internalizing, and externalizing manifolds. External cell names provide an ordering which is retained in the internal representations. The subdivision functor is applied cell-by-cell and may be applied adaptively. When the subdivision functor is applied, new cells in the subdivision are created. The boundary of a cell must be subdivided before a cell is subdivided. It is the responsibility of the subdivision functor to consistently construct all of the data structures. Cells in the subdivision usually correspond to a pair of previous cells, the cell being subdivided and a cell on the

subdivided boundary. Knowing which cells to subdivide, and subdividing according to their ordination, an encoder and decoder can arrive at the same resulting data structure (including ordination). Therefore, the actual information content in a subdivision is minimal. Several different subdivision functors may be adaptively intermixed and the result requires only a small amount of information to describe.

The dual intersection subdivision subdivides by intersection the dimensional dual of a complex with itself. In it most cells are hypercubes. The number of cell increases a 2^{dq}, where d is the dimension and q is the subdivision level. (Barycentric subdivision increases as $(d!)^q$). Edge mid-point subdivision of a tetrahedron results in an octahedron in the center. An k-cell in the subdivision of an n-cell corresponds (1:1) to an n-k-cell face on the subdivided boundary of the n-cell. The boundary of a k-cell in the subdivision corresponds to the adjoint of the corresponding n-k-cell face in the subdivided boundary of the n-cell. Dual intersection subdivision is stationary and only a finite number of compact support wavelet prediction stencils are required.



Original Dual Intersection Subdivision

Figure 4. Dual Intersection Subdivision

The wavelet analysis attaches function values to cells and identifies unique compactly supported wavelet stencil. The stencil topology must match local topology of the cell complex and the stencil coefficients must have at least the symmetries of the local of application. We can use stencil to predict function values at cells introduced by subdivision. We can also record at new cells the difference between the predicted and actual values and "lift" back corrections to values at old cells.

The compression thresholds the wavelet values. It elides subdivisions where wavelet values are uniformly zero and progressively encodes wavelet analysis by successively finer thresholds. For real valued functions we can work in some Besov space. It has been shown ([2]) that the number of non-zero wavelet values will increase much more slowly than the number of cells in the subdivided complex. The 2-D results that we have reported previously [3] indicate that this technique produces practically competitive algorithms. It also shows (for the first time) that compact topological spaces can be approximated in a computationally feasible manner.

4. SIMPLE TOPOLOGICAL EXAMPLE AND FUTURE STUDIES

We are developing a set of C++ classes for working with manifolds. The code is being developed for an SGI using the GNU g++ compiler and a PC using Visual C++. Among other things, the code allows for the construction of arbitrary finite–dimensional manifolds. A manifold can either be constructed on–the–fly during code execution or from a static specification in a file (the file must adhere to a predetermined format, an example of which follows).

4.1 Simple example (topological cube):

; Describes the surface of a cube.

Dimension 2

; Enumerates the 2-cells in the manifold and their orientation.

Manifold { 1 2 3 4 -5 -6 }

; This section describes the boundary of all the 2-Cells by enumerating the bounding 1-Cells of each 2-Cell. Each record in this section maps a 2-cell to its boundary chain. If a cell index in the chain field is negative, the cell will have a coefficient of -1. If the cell index is positive, the cell will have a coefficient of +1.

Cell 2 { (1) 1 2 3 4 (2) -3 5 6 7 (3) -6 8 9 10 (4) -9 11 -1 12 (5) 7 10 12 4 (6) 2 11 8 5 }

; The boundary of each 1-cell.

Cell 1 { (1) 2 -1 (2) 3 -2 (3) 4 -3 (4) 1 -4 (5) 5 -3 (6) 6 -5 (7) 4 -6 (8) 7 -5 (9) 8 -7 (10) 6 -8 (11) 2 -7 (12) 8 -1 }

; 0-cells don't have a boundary. We just specify the coordinates of the 0-cells.

Vertices { (1) -1 -1 -1 (2) 1 -1 -1 (3) 1 1 -1 (4) -1 1 -1 (5) -1 -1 1 (6) 1 -1 1 (7) 1 1 1 (8) -1 1 1 }

The classes in the library are extensible through the standard C++ inheritance mechanisms. They can be used to solve a variety of topological coding problems. For example, function values could be attached to a cell in an application studying multidimensional signal compression.

4.2 Future studies

Multidimensional signal compression. When the underlying signal has some notion of geometry, CW complexes can be used to approximate both the domain and range space of the signal. We would like to be able to build approximations to continuous functions via the cellular approximation theorem. A systematic approach to refining the approximation gives rise to a multiresolution scheme and the possibility of efficiently representing the signal, i.e., wavelets. The approach should preserve the compression techniques in [3]. We would like to prove for that case a theorem of the type described in [2].

Efficient representation of texture maps for computer graphics applications. In some computer graphics problems, thousands of simplices need to be texture mapped to properly display a scene. Efficient storage and rapid usability of texture maps can be studied using the library.

REFERENCES

[1] Brisson, E. "Representation of d-Dimensional Geometric Objects", PhD Thesis in CS, U. of Washington, 1990.

[2] DeVore, R. A., Jawerth, B. and Popov, V. A. "Compression of wavelet decompositions" *American J. of Mathematics* 114, 1992

[3] Kolarov, K. and Lynch, W. "Wavelet Compression for 3D and Higher-Dimensional Objects", Invited Paper, Proc. of SPIE Conference on Applications of Digital Image Processing, Volume 3164, San Diego, California, July 1997.

[4] Massey, W. S., *A Basic Course in Algebraic Topology*, Springer-Verlag, 1991

[5] MacCracken, R. and Joy, K. "Free-Form Deformations with Lattices of Arbitrary Topology", Proc. of Siggraph, August '96.