ALGEBRAIC MINING OF SOLID MODELS FROM IMAGES

Francesco Furiani[†], *Claudio Paoluzzi*[‡*] *and Alberto Paoluzzi*[†]

[†]Mathematics & Physics Dept, University "Roma Tre", Rome, Italy
[‡]Next S.p.A., Rome, Italy

ABSTRACT

We introduce a map algebra based on a cochain extension of the Linear Algebraic Representation (LAR), used to efficiently represent and query geometric and physical information through sparse matrix algebra. LAR, based on standard algebraic topology methods, supports all incidence structures, including enumerative (images), decompositive (meshes) and boundary (CAD) representations, is dimension-independent and not restricted to regular complexes. This algebraic representation enjoys a neat mathematical format—being based on *chains*, the domains of discrete integration, and *cochains*, the discrete prototype of differential forms, so naturally integrating the geometric shape with the supported physical properties, and provides a mechanism for strongly typed representation of all physical quantities associated with images. It is easy to show that k-cochains form a linear vector space over k-cells, which means that they can used as basic objects in a rich and virtually unlimited calculus of physical properties.

Index Terms— Map algebra, image information mining, solid modelling, cochain complex, algebraic topology

1. INTRODUCTION

Computational problems in science and technology must deal with *increasingly complex* geometric information and applications. The complexity of geometric information stems from dramatic increase in *size, diversity, and complexity* of geometric data, including digital images, point clouds, boundary schemes, NURBs representations, finite element meshes, 3D medical images. This increasing complexity of geometric information and applications, and the goals of *unification*, *scalability*, and support of *massively parallel* distributed computing, strongly push for rethinking the foundations of geometric and topological computing. In particular the emerging applications from space, nano & bio technology, and medical 3D, require a novel *convergence of shape synthesis and analysis* methods from computer imaging, computer graphics, computer-aided geometric design, with meshing of computational domains and physical simulations.

Objects and relations distributed in time and space are all *cell complexes*. Examples: digital images, finite element meshes, B-reps of solids, assemblies, networks, and so on, are all cell complexes of various dimension [1]. Such decompositions of space into k-cells ($0 \leq k \leq d$) may generate *k-chain spaces*, linear spaces constituted by any combination of k-cells, where (a) singletons of k-cells give a *basis*, i.e. a minimal set of generators, and (b) linear *boundary operators* compute the boundary chain of *any* given chain, by mean of a single SpMV (sparse matrix-vector) multiplication.

Basically, k-cochains are discrete densities of quantities contained in the k-cells of a cell complex (such as a digital image or a finite element mesh), k being the dimension. The LAR (Linear Algebraic Representation) scheme [1] is a simple, general and effective representation of (co)chain complexes [2], based on a CSR (Compressed Sparse Row) representation [3] for characteristic matrices of linear spaces of (co)chains. LAR supports all topological incidence structures, is dimension-independent and not restricted to regular, i.e., dimensionally uniform complexes. It allows for fast validity checks of the topology of geometric models, possibly generated from 3D scanner data or extracted from 3D images, using only elementary linear algebra, namely, sparse matrix-vector multiplication. Any query about incidence relations between chains or cochains of same or different dimensions are answered by a single SpMV product.

2. LINEAR ALGEBRAIC REPRESENTATION (LAR)

2.1. Cellular Complexes

All *geometrical objects* considered in this paper are *chains* in a *cellular complex* $\Lambda(X)$ partitioning a (topological) space $X \subset \mathbb{E}^d$. Informally, a cellular complex is made of basic building blocks called *cells*, suitably glued together [4].

More formally, a cellular complex is a Hausdorff space X, i.e. a topological space in which distinct points have disjoint neighbourhoods, together with a partition $\Lambda = \Lambda_0 \cup \cdots \cup \Lambda_d$ of X into open cells (of varying dimension) that satisfies some additional properties. Some definitions useful in the remainder follow.

A compact topological subspace is a *convex cell* if it is the set of solutions of affine equalities and inequalities. A *face* of a cell is the convex cell obtained by replacing some of the inequalities by equalities. A *facet* of a cell is a face defined

C.P. performed the work while on sabbatical from consulting at Thales Alenia Space S.p.A, Rome.

The research of A.P and F.F. was supported by a POC grant 2012/13 by SOGEI, the ICT company of the Italian Ministry of Economy and Finance.

by just one equality. The *dimension* n of a n-cell is that of its *affine hull*, the smallest affine subspace that contains it. A c*onvex-cell complex* or *polytopal complex* P is a finite union of convex cells such that: (i) if A is a cell of P, so are the faces of A; (ii) the intersection of two cells of P is a common face of each of them. A simplicial (respectively, cuboidal) complex is a polytopal complex where all cells are simplices (respectively, cuboids). The dimension of P is the maximal cell dimension of P. The r-skeleton P_r is the subcomplex formed by the cells of dimension $\leq r$. The 0-skeleton coincides with the set $V(P)$ of *vertices* of P.

2.2. LAR definition

The *Linear Algebraic Representation* (LAR) scheme supports all topological constructions and queries that arise in typical cellular decomposition of space (image, mesh, boundary, etc.). Formally, LAR relies on standard definitions [4, 5]: in the mod 2 cellular complexes, d-chains are *sets* of d-cells; the standard basis of the $\mathbb{Z}_2$-linear space C_d of d-chains is provided by *singletons* of d-cells; each d-cell is represented by a map $C_d \rightarrow \mathbb{Z}_2\, C_0$, i.e. by a *row* of a binary characteristic matrix M_d. Of course, every d-chain in C_d may be generated by a $(\mathbb{Z}_2)$-linear combination of M_d rows. The formulation may be extended to d-cochains that represent any possible field over the chains. In this context, the boundary is a linear operator and the coboundary is its dual. Boundary and coboundary operators provide the computational tools needed by the discrete version of the generalized Stokes theorem for integration of fields over d-dimensional domains [6].

The characteristic matrices of a chain complex, i.e. the binary matrices that encode the incidence of d-cells with 0-cells, provide a convenient tool for computing boundary (and coboundary) operators and answering queries concerning the topological relations between cells [1]. Characteristic matrices are *very* sparse for actual chain complexes used in applications, and can be represented in a standard CSR (Compressed Sparse Row) format. The product and transposition of CSR matrices [7, 3], needed to compute the boundary, adjacency and incidence operators between such linear spaces, are intrinsically efficient, since the sparse matrix-vector (SpMV) multiplication is linear in the size of the (sparse) output.

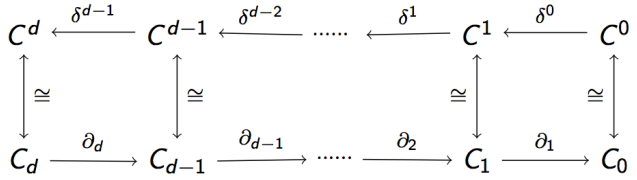

Fig. 1. Chain and cochain complex as a sequence of linear spaces of chains (C_d), i.e. domain subsets, and cochains (C^d), i.e. discrete fields, with linear boundary (∂) and coboundary (δ) operators, under constraints $\partial_{d-1} \circ \partial_d = \delta_d \circ \delta_{d-1} = 0$.

2.3. Chains and cochains

Given a space partition into k_d connected d-cells, we are mainly interested to d-chains, *sets* of d-cells, members of the linear space C_d with scalars in $\mathbb{Z}_2 = \{0, 1\}$. Accordingly, the coordinate representation of a cell $\lambda \in C_d$ is a binary vector of length k_d with only one unit element.

d-Cochains are the elements of the dual space $C^d := C_d^* = \{f : C_d \rightarrow \mathbb{R}\}$ of real-valued linear functions of chains. By definition, a d-cochain is (a functional representation of) a discrete field over a cellular d-complex partitioning a given space. The standard basis of C^2 is given by the characteristic functions $\chi^i : C_2 \rightarrow \mathbb{R}$ such that $\chi^i(\lambda_j) = \delta_{ij}$, with $\delta_{ij} = 1$ $(i = j)$, and $\delta_{ij} = 0$ $(i \neq j)$. A 2-cochain is represented as $f = \sum_i f^i \chi^i$, where $f^i(\lambda_i) \neq 0$ gives the value of f on λ_i, and $f^i(\mu) = 0$ elsewhere. The evaluation of a cochain f over a chain λ is given by $f(\lambda) := \langle \lambda, f \rangle = \sum \lambda_i f^i$, and is the discrete prototype of domain integration.

3. EXTRACTION OF (CO)CHAINS FROM IMAGES

LAR takes advantage of the data-parallel many-core computational model supported by the graphic processing units, designed to rapidly manipulate memory to accelerate the creation of images for output display. GPUs are present in embedded systems, mobile phones, personal computers, workstations, and game consoles. Of course, modern GPUs are extremely efficient for producing computer graphics, and their highly parallel structure is more effective than general-purpose CPUs for algorithms where processing of large data blocks may be performed in parallel.

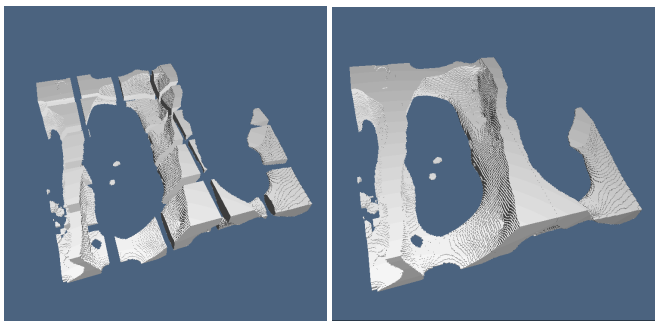

Fig. 2. Since the bottleneck of GPGPU is moving data from global to local memory, the extraction of model portions is done in parallel using the *same* $[\partial_3]$ matrix for a n^3 subimage.

The extraction of a solid model from an image was implemented using a *divide-et-impera* approach (see Figure 2). The cell complex taken into account is a n^3 set of image voxel, with $n \in \{64, 128\}$ depending on the size of the GPU storage. The matrix $[\partial_3] = [\delta_2]^t$ of the (co)boundary operator defined on the (co)chain complex supported by such subimage is computed once and for all, and stored in GPU's constant

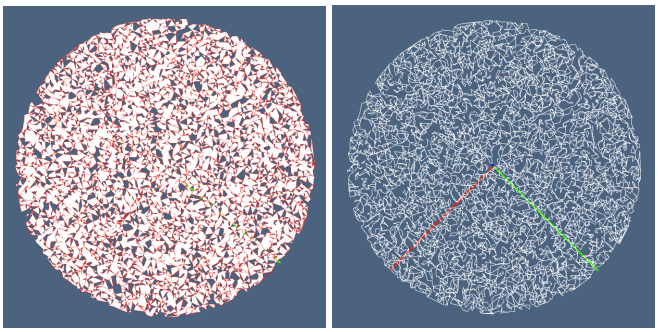

Fig. 3. (a) A random polygon in the unit circle; (b) the boundary of the total chain, computed by a single SpMV multiplication. The same $[\partial_2]$ matrix is used to extract the boundary of *any* polygon subset.

memory, so that the boundary models of several image portions can be extracted in parallel, by sending to device's work groups a set of coordinate representations (proper strings of bits) of the chains to be extracted. The coordinate representations of their boundaries are computed in parallel using local memory to perform several parallel SpMV operations.

LAR is being developed [8] as a service in a web-based computational environment, using some Khronos's APIs for industry standard heterogeneous computing. The power of GPU to perform accelerated topological computation has been capitalised via OpenCL, the open general-purpose GPU computing language [9]. By accelerating the topological operators we show the LAR native data-parallel nature and versatility, further endorsed by the recent WebCL specification (a way to use OpenCL through browser-based technologies) exposed as a computational service to be accessed by clients that do not have access to silicon accelerated solutions.

In particular, LAR is being used in biomedical applications which require fast performances with big geometric data for topological tasks such as model extraction from 3D images (see Figure 4). Density values in medical images represent scalar fields (cochains) over cubical cellular complexes, and LAR is used to guarantee topologically correct 3D image segmentation as well as to extract (enumerative) solid model from the image[1], which is subsequently smoothed out (see Figure 4). A nice feature of this approach is that the whole image is partitioned into a set of cochains associated to field intervals, including the interstitial space, so providing a well-defined meshing of both the features and their outer space.

The main operation we have implemented [8] in OpenCL is of course the SpMV multiplication, where three kernels are run on a set of chain vectors to build the result of SpMV product in CSR format: the first kernel computes the number of non-zero elements per row; the second computes a prefix scan on such elements. It is implemented via the prefix sum proposed in [10], with local cache and blocks to divide the work shares among work-items, in a divide-et-impera fashion; then the partial results are merged and propagated globally. The third kernel builds and returns the output of the operation.

Although being the core part of the LAR library, the OpenCL kernels for SpMV and SpMM (sparse matrix-vector and sparse matrix-matrix multiplication) are not the only software components; to make them work smoothly, other computations are needed on the host side, that must be able to receive and manipulate the input, send it to the GPU, verify the execution state and return the output. Furthermore, the library must be able to empower calculations either on accelerated hosts or not. To achieve this goal, besides the kernel modules, two different architectures have been devised: a REST computational service and a pipelined single host system. These architectures were implemented in different languages (javascript, java and C++) and using (various combination of) different technologies [8].

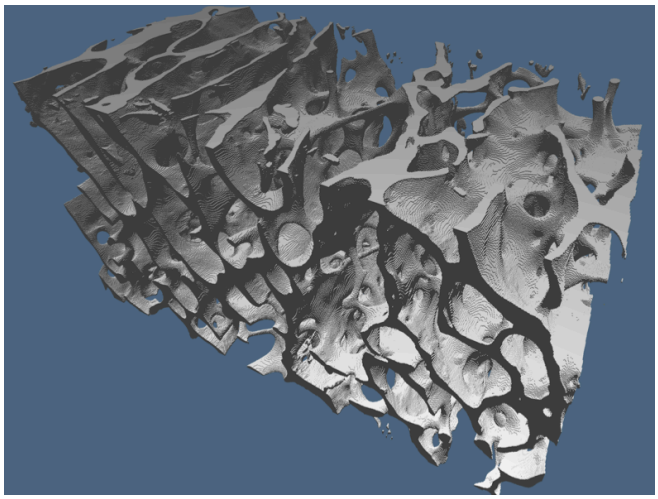

Fig. 4. The solid model of a sample of spongy bone extracted from a 3D image (μNMR).

4. MAPPING IMAGES TO GEOMETRIC MODELS

We are currently working to the smoothing and transformation of the models extracted from images into discrete meshes of quotient cochains, i.e. into a discrete set decomposition of the mined models. The resulting output is a cell complex with a strongly reduced number of cells, characterised by a constant density of some physical quantity in each cell. The goal of this mapping is to transform the extracted model from a big collection of small simplices to a greatly reduced number of discrete cells with simple topology and possibly curved boundaries. Notice that the output of the mining step, described in the previous section, is either a big set of line intersections between adjacent pixels for 2D images, or a big set of

[1]This work is carried out within the framework of the IEEE-SA Project P3333.2 - Standard for "Three-Dimensional Model Creation Using Unprocessed 3D Medical Data".

triangles defined by the intersection of adjacent voxels in the 3D case. In particular, we are making experiments with a set of linear sparse matrix algorithms derived from the Taubin filter [11] and from the Morse-Smale topology decompositions methods derived from [12].

5. APPLICATIONS

Space products like satellite imagery and global positioning systems (GPS) are entered into common public everyday use. Another space technology, GPR (ground penetrating radar or *undersurface* radar), is now getting on the stage. In particular, GPR was already used to map large archaeological sites and ruins, and to study underground geologic formations, like water reservoirs under the surface of Mars planet [13, 14]. The depth penetration of micro and radio waves is greater, and their use is easier and more economic, than conventional geological investigation methods. Some new technology, like atomic dielectric resonance (ADS), already provides geophysical services for the location, identification, mapping, and exploration of undersurface natural resources, without relying upon interpretation or probability.

Even more promising than the previous generation of satellite imagery of the Earth surface, scanning via remote sensors the top layers of the planet crust is going to produce very-large amounts of digital data, that can be employed for a range of useful products and services to the public, including integration of outdoor and indoor mapping, reconstruction of 3D building models, discovery of archeological sites, underground utility mapping, road and bridge inspection, monitoring and early-alarming of high-risk geological areas, disaster prevention and recovery, geodynamical models, control of unauthorized construction, security operations in densely built environments, and military operations.

In our view such a computational environment, enriched with proper spatial indexing, will be able to efficiently manage huge cell decompositions of Earth crust parts of variable depths, by integrating grids, maps and images with 3D static and time-varying FEM meshes. E.g., geologists well understand that fully 3D geodynamic models, incorporating heterogeneity in boundary conditions and lithospheric properties, are required to describe more accurately the development of collisional orogens [15].

6. REFERENCES

[1] A. DiCarlo, A. Paoluzzi, and V. Shapiro, "Linear algebraic representation for topological structures," in *2013 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, New York, NY, USA, 2013, SPM '13, Acm, To appear on *Computer-Aided Design*, Volume 46, January 2014, Pages 269–274.

[2] A. DiCarlo, F. Milicchio, A. Paoluzzi, and V. Shapiro, "Chain-based representations for solid and physical modeling," *Automation Science and Engineering, IEEE Transactions on*, vol. 6, no. 3, pp. 454–467, 2009.

[3] Aydın Buluç and John R. Gilbert, "Parallel sparse matrix-matrix multiplication and indexing: Implementation and experiments," *SIAM Journal of Scientific Computing (SISC)*, vol. 34, no. 4, pp. 170 – 191, 2012.

[4] Allen Hatcher, *Algebraic topology*, Cambridge University Press, 2002.

[5] Jean-Claude Hausmann, *Mod Two Homology and Cohomology*, 2012, Book Project.

[6] Hassler Whitney, *Geometric integration Theory*, Dover, 1957.

[7] Anton Lokhmotov, "Implementing sparse matrix-vector product in OpenCL," in *OpenCL tutorial, 6th Int. Conference on High Performance and Embedded Architectures and Compilers (HiPEAC'11)*, 2012.

[8] Francesco Furiani, "HPC acceleration of topological operators in the linear algebric representation," M.S. thesis, Dept of Computer Science and Automation, University Roma Tre, Rome, 2013.

[9] Janusz Kowalik and Tadeusz Puźniakowsk, *Using OpenCL: Programming Massively Parallel Computers*, IOS Press, 2012.

[10] Shubhabrata Sengupta, Mark Harris, Michael Garland, and John D. Owens, "Efficient parallel scan algorithms for many-core GPUs," in *Scientific Computing with Multicore and Accelerators*, Jakub Kurzak, David A. Bader, and Jack Dongarra, Eds., Chapman and Hall/CRC Computational Science, chapter 19, pp. 413–442. Taylor and Francis, 2011.

[11] Gabriel Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, New York, NY, USA, 1995, SIGGRAPH '95, pp. 351–358, ACM.

[12] Attila Gyulassy, Peer-Timo Bremer, Bernd Hamann, and Valerio Pascucci, "A practical approach to morse-smale complex computation: Scalability and generality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1619–1626, Nov. 2008.

[13] G. Picardi, J. Plaut, O. Bombaci, D. Calabrese, M. Cartacci, A. Cicchetti, S. Clifford, P. Edenhofer, W. Farrell, C. Federico, A. Frigeri, D. Gurnett, T. Hagfors, E. Heggy, R. Huff, A. Ivanov, W.T.K. Johnson, R. Jordan, D. Kirchner, W. Kofman, C. Leuschen, E. Nielsen, R. Orosei, E. Pettinelli, R. Phillips, D. Plettemeier, A. Safaeinili, R. Seu, E. Stofan, G. Vannaroni, T. Watters, and E. Zampolini, "Radar soundings of the subsurface of Mars," *Science*, vol. 310, pp. 1925–1929, December 2005.

[14] Claudio Paoluzzi, "Dual frequency algorithms for Sharad's SAR signal processing," M.S. thesis, InfoCom Dept, University La Sapienza, Rome, 2008, In Italian.

[15] S. Sol, A. Meltzer, R. Bürgmann, R. D. Van Der Hilst, R. King, Z. Chen, P. O. Koons, E. Lev, Y.P. Liu, P. K. Zeitler, X. Zhang, J. Zhang, and B. Zurek, "Geodynamics of the southeastern Tibetan Plateau from seismic anisotropy and geodesy," *Geology*, vol. 35, no. 6, pp. 563–566, June 2007.