

CAD models from medical images using LAR

Alberto Paoluzzi^a , Antonio DiCarlo^a , Francesco Furiani^a  and Miroslav Jiřík^b 

^aRoma Tre University, Italy; ^bWest Bohemia University, Czech Republic

ABSTRACT

This paper points out the main design goals of a novel representation scheme of geometric-topological data, named Linear Algebraic Representation (LAR), characterized by a wide domain, encompassing 2D and 3D meshes, manifold and non-manifold geometric and solid models, and high-resolution 3D images. To demonstrate its simplicity and effectiveness for dealing with huge amounts of geometric data, we apply LAR to the extraction of a clean solid model of the hepatic portal vein subsystem from micro-CT scans of a pig liver.

KEYWORDS

Solid modeling; representation scheme; linear algebraic representation; LAR; 3D medical images

1. Introduction

Technological advances made it possible to acquire large sets of biomedical data at a fast rate and affordable costs. In turn, the easiness of producing and collecting data in digital form has triggered a progressive paradigm shift from experiments on model organisms to simulation based on virtual prototypes and mathematical modeling [21, 22, 26, 45, 47, 3].

The capability to extract geometrical models from medical images fosters the development of quantitative, evidence-based medicine, where laboratory and clinical observations are cumulated and made accessible to integrative research [22]. In the near future, the collected knowledge of multifarious physiological subsystems on a hierarchy of dimensional scales and of a variety of biological functions will be formalized, catalogued, organized, shared and combined in many ways, providing integration across subsystems, temporal and spatial scales, biomedical and bioengineering disciplines, to give rise to personalized healthcare [3, 22, 47, 21].

Consistently with the availability of quantitative data, the interest in physically-based simulations, customary in engineering CAD, is now growing also in medicine, with the clinical aim of getting a better understanding of physiology and pathologies on a single-patient basis, using personalized models extracted from patient's body scans. A meaningful example of this trend, akin to the application we focus on in this paper (the extraction of the liver portal vein system), is provided by the current developments in techniques aiming at providing surgeons with accurate, patient-specific guidelines when designing partial

hepatic resections for the treatment of liver tumors [37, 24, 23].

The rising applications of 3D medical modeling [25], computer-based training of medical doctors, computer-assisted surgery, etc. call for the convergence of methods and know-hows from computer imaging, computer graphics, geometric/solid modeling, and physical modeling and simulation. Similar challenges are posed also by more established endeavors, such as materials science—think of soft matter, engineered surfaces, nano-materials and meta-materials—and biophysics, where modeling issues range from the molecular/protein level to multi-scale modeling of subcellular organelles, cellular structures, tissues and organs. Serious progress in these directions demand major innovations, from cooperative collaboration to multi-physics support, where different field equations imply different geometric structures at the level of basic descriptive data, to enhanced robustness toward scale mismatch in coupled problems, complexity of the simulation environment, terascale number of elementary entities or agents.

As a contribution to these efforts, we present here a novel representation scheme [14] which unifies the treatment of images, meshes and polyhedral data, and requires the minimum amount of storage for a complete representation of their topology and geometry.

1.1. Previous work

The foundational concept of representation scheme in solid modeling as a mapping between mathematical models and computer representations, was defined at

Rochester by the Production Automation Project, led by Herbert Voelcker and Aristides Requicha at the end of the seventies, and sparked an enormous research work in the two following decades.

This work on representation schemes includes, among others, [36, 35, 39, 8, 6, 10, 15, 19, 50, 2, 38, 48, 42, 40, 28, 32, 9, 18, 33, 41, 20, 34, 13, 4, 31, 29, 30, 1, 7]. It is apparent that the totality of representation schemes for solid modeling, in particular the so-called non-manifold representations, provide very intricate and super-linked data structures, most of the times paired with very specialized software functions to perform localized changes on these data structures, in order to preserve topological consistency.

B-splines and NURBS emerged concurrently from research developments in numerical analysis, as the ubiquitous and most useful mathematical tool to support boundary representations of solids, and the first geometry kernels were created in some universities in US and Europe. These kernels, transferred into commercial software and supported by huge investments, later became the foundational framework for all commercial solid modelers and the then emerging business of PLM systems for aerospace, automotive, naval, and manufacturing industries.

Nowadays, while the information and communication technologies are changing at a furious pace, the most widely used software tools in the PLM industry are still based on the old approach established about twenty years ago, centered around non-manifold topology, boundary representation, NURBS curves and surfaces, despite the tremendous amount of research done and the continuing technical advances in computer-aided design, geometric computing and scientific visualization.

1.2. Our contribution

We believe that the time is ripe to radically rethink the fundamentals of solid modeling. Web- and cloud-based systems of the next generations urge that complex algorithms and data structures be unified and simplified, making them document-based processing compatible and strongly distributed, according to the *MapReduce* paradigm.

In particular, with LAR we aim at providing the simplest and most general representation of topological data, equally suitable for use in solid modeling, basic computer graphics and rendering, 3D printing, image processing, geometric modeling, and meshing for physical simulations. Algebraic topology provides the proper language, where collections of d -cells are described as d -chains, elements of a linear space, parameterized by sequences of signed integers.

In this paper we show that most common geometric and imaging operations reduce in LAR to simple compositions of linear operators, implemented by sparse matrix multiplication and transposition, possibly supported by advanced graphics hardware. We expect this approach to be beneficial for producing the CAD tools of the next generation, capable to face the challenge posed by the treatment of big geometric data, when solid models are to be derived from 3D and 4D high-resolution images. A sample application of this sort is presented and discussed in Section 4.

1.3. Paper organization

The present paper is organized as follows. Section 2 gives an elementary introduction to the main definitions of Linear Algebraic Representation, illustrated with some simple examples of cellular spaces. Section 3 discusses the principal topological tools in LAR and presents several related algorithms. Section 4 is dedicated to the LAR of images, showing how to map the structure of a d -dimensional image to a cellular complex of d -cuboids, and discussing how to reduce the standard morphology operators on images to the composition of topological operators on chain complexes. This section also contains the main application of LAR presented in this paper, i.e. the extraction of a well-defined solid model of the liver portal vein system from 3D images. Section 5 summarizes the current state of our prototype implementation and points to future developments.

2. Linear algebraic representation

A *representation scheme* is a mapping between the mathematical spaces to be represented by a computer system and their symbolic representation in computer memory [36]. The *Linear Algebraic Representation* (LAR) scheme [14], uses *Combinatorial Cellular Complexes* (CCC) as its mathematical domain [5], and various compressed representations of *sparse matrices* [11] as its codomain.

Since LAR provides a *complete* representation of the topology of the represented space, the matrix $[\partial_d]$ of the *boundary operator* may be used to compute the coordinate representation $[\partial_d c]$ of the boundary chain of any collection c of cells, through a single operation of SpMV multiplication [11] between the CSR (Compressed Sparse Row) representation of $[\partial_d]$ and the CSC (Compressed Sparse Column) representation of the $[c]$ chain.

Importantly, the matrices of *coboundary operators* $[\delta_0]$, $[\delta_1]$, and $[\delta_2]$, computable in the LAR scheme by means of multiplications between sparse matrices, provide respectively the discrete *gradient*, *curl*, and *divergence* on the given space decomposition. The *Laplacian*

operator Δ is computed as a combination of boundary and coboundary operators. Last but not least, the standard operators of *mathematical morphology* on images (*dilation, erosion, opening and closing*) [16] are obtained by product of sparse matrices of topological incidences times sparse matrices of boundary and/or coboundary.

For the sake of concreteness, in the following section we use some *python* notations and expressions, often directly computable in (<https://github.com/cvdlab/lar-cc>), our open-source prototype software system.

2.1. Some basic definitions and examples

The first important concept introduced by LAR is the definition of the *model* of a cell complex, as composed of a list of vertices, each of which is given as a list of coordinates, and by one or two topological relations. The *list* structures we refer to, here and in the sequel, are

modern data structures, typical of languages like *python* or *javascript*, where the access to a list element, given its ordinal value, requires constant time.

Definition 1 (LAR model): A LAR model is either a pair V, FV , or a triple V, FV, EV , where:

- (1) V is the list of vertices, given as lists of coordinates;
- (2) FV is a cell-vertex relation, given as a list of cells, where each cell is given as a list of vertex indices;
- (3) EV is a facet-vertex relation, given as a list of cells, where each cell is given as a list of d vertex indices and facet stands for $(d - 1)$ -face of a d -cell.

2.1.1. Examples

Some very simple examples of 0D, 1D, and 2D models are displayed in Figure 1. It is to be remarked that the LAR representation scheme is *dimension-independent*. Hence,

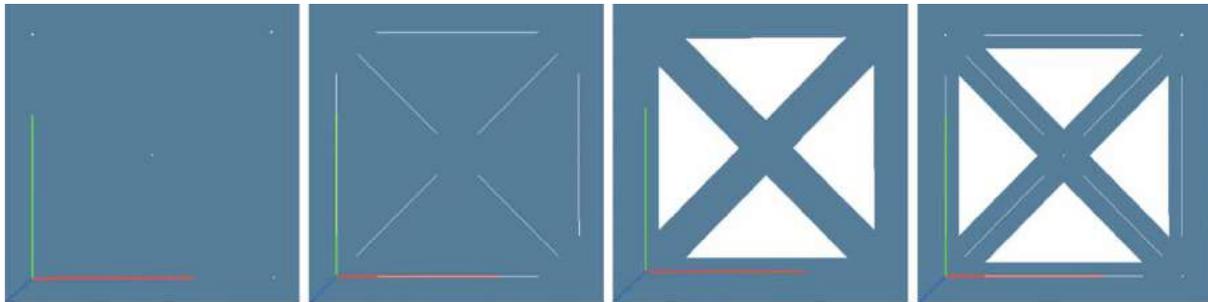


Figure 1. Images of model10d, model11d, model12d, complex2d, drawn exploded.

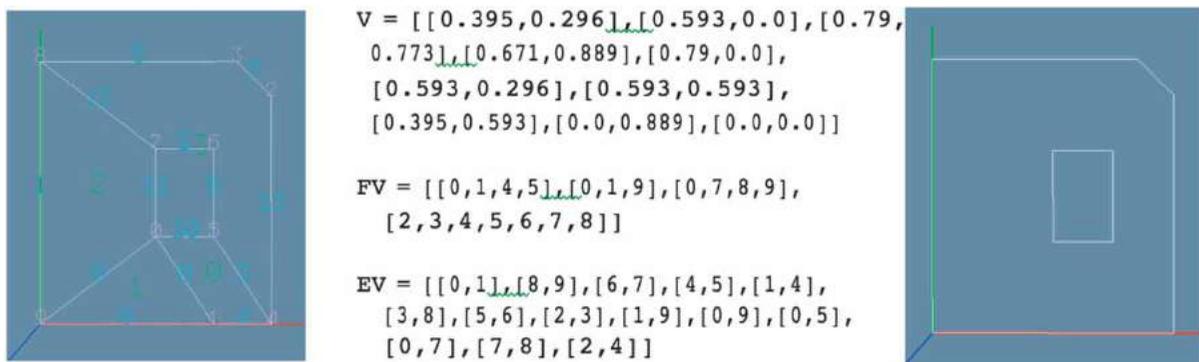


Figure 2. (a) LAR model with 0-, 1-, and 2-cells; (b) the triple V, FV, EV of vertices, faces and edges (indexed on vertices); (c) the extracted boundary. Note that 2-cells have different numbers of vertices, and may be non-convex.

$$FV = \begin{bmatrix} [0, 1, 4, 5], \\ [0, 1, 9], \\ [0, 7, 8, 9], \\ [2, 3, 4, 5, 6, 7, 8] \end{bmatrix} \mapsto M_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \mapsto \begin{matrix} <4x10 sparse \\ matrix of type \\ 'numpy.uint8' \\ with 18 stored \\ elements in CSR \\ format> \end{matrix}$$

Figure 3. The binary characteristic matrix M_2 (center) of the cellular complex in Figure 2 and its BRC (left) and CSR (right) representations.

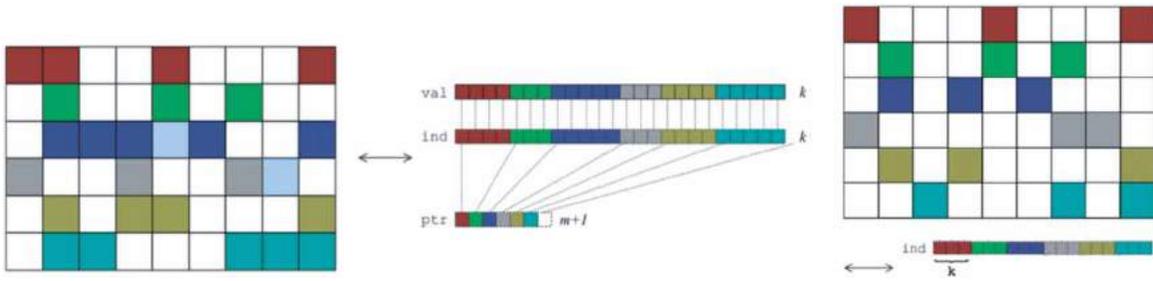


Figure 4. Compressed Sparse Row (CSR) matrix: (a) general case with storage in 3 arrays (image from [49]); (b) special case: LAR of d -meshes, with binary values and same number k of non-zeros in each row, with $k = d + 1$ (regular simplicial d -complexes) or $k = 2d$ (regular cuboidal d -complexes).

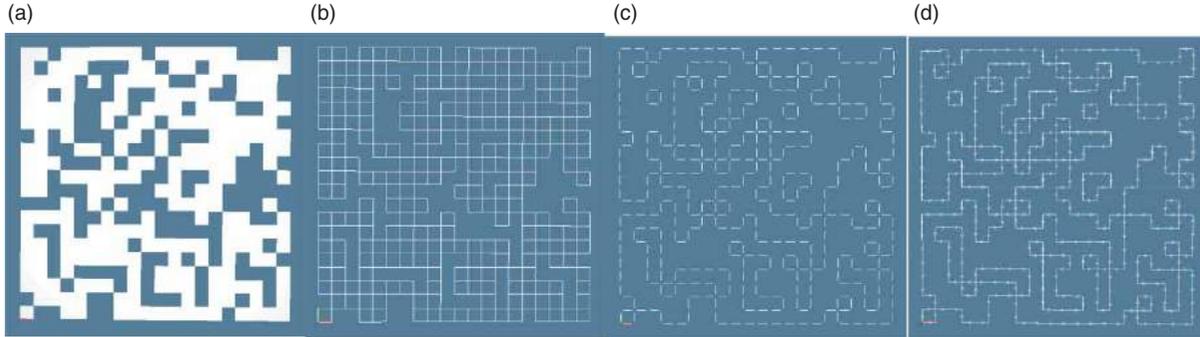


Figure 5. Orientation of the boundary of a randomly generated cuboidal 2-complex; (a) 2-cells; (b) 1-cells; (c) exploded boundary 1-chain; (d) coherently oriented boundary 1-chain.

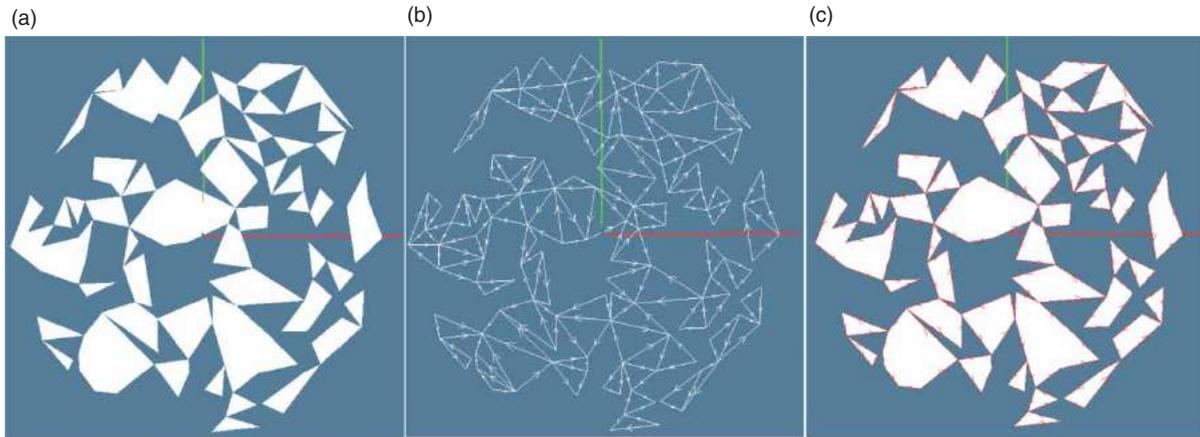


Figure 6. Orientation of the boundary of a randomly generated simplicial 2-complex; (a) 2-cells; (b) 1-cells; (c) coherently oriented boundary 1-chain (red).

the same data structures and algorithms can be used in every Euclidean space.

Example 1 (0D, 1D, 2D model examples): The LAR definition of models in Figure 1 follows:

$$\begin{aligned}
 V &= [[0, 0], [1, 0], [0, 1], [1, 1], [0.5, 0.5]] \\
 VV &= [[0], [1], [2], [3], [4]] \\
 EV &= [[0, 1], [0, 2], [0, 4], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4]] \\
 FV &= [0, 1, 4], [1, 3, 4], [2, 3, 4], [0, 2, 4]
 \end{aligned}$$

$$\begin{aligned}
 &model0d, model1d, model2d, complex2d \\
 &= (V, VV), (V, EV), (V, FV), (V, VV + EV + FV)
 \end{aligned}$$

Definition 2 (Complete LAR model.): A complete d -model is a triple V, FV, EV , where the relations FV and EV define the d -cells and $(d-1)$ -cells, respectively

Remark 1 (Complete LAR model.): In order to compute the boundary and coboundary operators ∂_d and $\delta_{d-1} = \partial^T$, in general the lists V, FV and EV are all

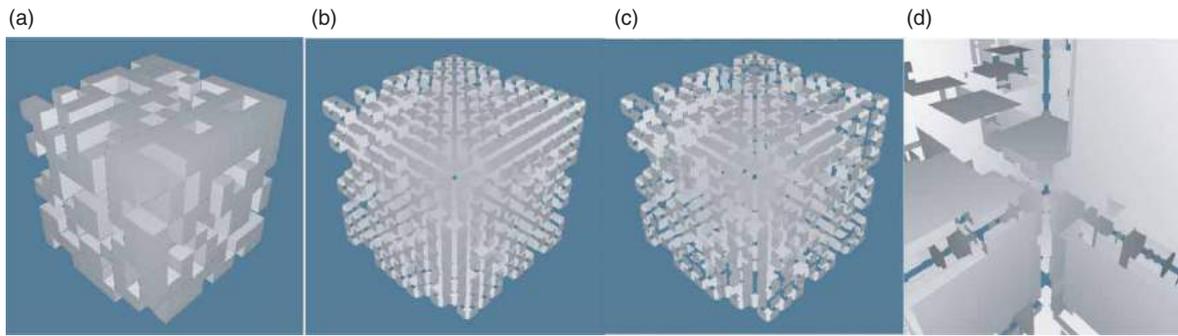


Figure 7. Extraction of the oriented boundary of a cuboidal 3-complex: (a) cellular 3-complex; (b) complex of 2-cells; (c) (exploded) oriented boundary; (d) exploded view from inside.

needed, since both d - and $(d-1)$ -cells need to be explicitly described. However, for meshes and grids where all cells have the same number of vertices the pair V , FV suffices, since EV can be computed in linear time from FV . These complexes (such as triangular, quadrilateral, hexahedral, and regular simplicial complexes), while special, are widely used in important applications.

3. Basic representations

Several basic representations of topology are used in the LARCC library, including some common *sparse matrix* representation: CSR (Compressed Sparse Row), CSC (Compressed Sparse Column), COO (Coordinate Representation), and BRC (Binary Row Compressed).

3.1. (BRC) Binary Row Compressed

We denote as BRC (Binary Row Compressed) the standard input representation of the LAR-CC computational framework. A BRC representation is an array of arrays of integers, with no requirement of equal length for the component arrays. The BRC format is used to represent a (typically sparse) binary matrix. Each array component corresponds to a matrix row, and contains the indices of columns that store a 1 value. Zero values are not stored. Consider the LAR model in Figure 2. The relationships between the BRC of FV relation, the binary matrix M_2 and its CSR representation are shown in Figure 3. An example of a general CSR matrix is given in Figure 4 (left). In Figure 4 (right) the special case of the CSR associated to a simplicial mesh is discussed.

3.2. Compressed Sparse Row (CSR) matrix storage

General purpose representations combining a description of the boundary with a description of the interior of the model are mostly used for physical simulation, whereas computer graphics applications usually prefer a boundary representation, like the set of triangles or

quads on the boundary surfaces, efficiently supported by LAR. For example, the triangle-mesh geometry representation used in the STL format for stereolithography, the open standard AMF (Additive Manufacturing File) and the ISO/ASTM standard for describing objects for additive manufacturing processes (3D printing) can be considered, as far as geometry is concerned, as special cases of LAR.

3.2.1. 3D triangulations

Unstructured representations of the interior of an object often consist of 3D tetrahedra, and store the four references to vertices and the four references to the adjacent tetrahedra for each tetrahedron [28, 27]. This adjacency information is efficiently reconstructed by LAR, whose long-term storage and transmission consists of only 4 vertex indices per tetrahedral cell.

3.2.2. Solid boundary representations

The representation scheme of topology most frequently used by solid modelers is a decompositive representation of the boundary, to be coupled with a meshing of the interior just in case of need. The boundary is usually decomposed into *faces*, with face boundaries represented in turn by a decomposition into *edges*, given as pairs of *vertices*. In the case of *manifold* representations, storing only a subset of the binary incidence between such boundary elements is sufficient. Usual non-manifold representation relations include by necessity some set of pointers between incident pairs of boundary elements, usually circularly ordered to discriminate locally between interior and exterior, so doubling (at least) the storage size of the representation. Contrariwise, LAR includes only lists of cells as unordered lists of vertex indices, and manages equally well both manifold and non manifold models. Figures 5 and 6 display the oriented boundaries of LAR models of 2D cuboidal and simplicial meshes, respectively. The extraction of the oriented boundary of a cuboidal 3-chain is presented in Figure 7.

3.2.3. Comparison with Baumgart's scheme

The common reference term for comparing the memory requirement of solid boundary representations in 3D is the *Winged-Edge* scheme by Baumgart [6], which makes use of relation tables with a storage occupancy $8|E| + |V| + |F|$, where F , E , V stand for the sets of boundary faces, edges and vertices, respectively. An equivalent LAR representation of topology of the boundary of a 3D solid (B-rep) needs only the storage of the $CSR(M_2)$ sparse matrix, corresponding to the FV incidence relation, and the computation of the $CSR(M_1)$ sparse matrix, to obtain the EV relation, for a total memory size of $2|E| + 2|F|$, according to [50].

3.3. Some LAR-based operations and algorithms

Several LAR-based algorithms have already been implemented in our prototype *python* code `lar-cc`, mostly in the *MapReduce* style:

- generation of 0- and 1-dimensional cellular complexes;
- generation of *simplicial* and *cuboidal* d -complexes;
- computation of (sparse) matrices of *hk-incidence relations* ($0 \leq h, k \leq d$);
- computation of $(d - 1)$ -faces (*facets*) of d -cells;
- computation of the all hierarchy of *k-skeletons* ($0 \leq k \leq d$), based on the previous algorithm combined with efficient sorting and removal of duplicates;
- computation of *boundary* and *coboundary* matrices for both *oriented* and *non-oriented* complexes;
- *Cartesian product* of cellular complexes;
- *extrusion* of simplicial complexes;
- computation of *integrals* of polynomials over polyhedral domains (2D, 3D);
- generation of *hierarchical chains* (i.e., *structures*) and *cochains* over a cellular complex;
- computation of the cellular complex generated as *assembly* (i.e., hierarchical structure) of LAR models and affine transformations.

4. LAR of images

In this section we mainly discuss how to map a d -image, with normally $d \in \{2, 3\}$, to the coordinate representation of *chains* (collections of voxels) within the linear space C_d generated by the *cellular complex* corresponding in (generalized) row-major order to the image voxels, using LAR.

4.1. From d -images to chains and cochains

In order to generate the coordinate representation of a chain in a multidimensional image (or d -image), we

choose a basis of image elements—i.e., of d -cells—and a total ordering of image voxels, then map the multi-index identifying each d -cell to a single integer, so labeling the cell with its ordinal position within the chosen basis ordering.

Grid of hyper-cubes: Let $N_h := (0, 1, \dots, n_h - 1)$ be an ordered set of integers. Then $S := N_0 \times N_1 \times \dots \times N_{d-1}$ is the set of *multi-indices* of elements of a d -image.

Definition 3 (d-image shape.): The shape of a d -image, with $N = n_0 \times n_1 \times \dots \times n_{d-1}$ elements, called voxels or d -cells, is the ordered set $(n_0, n_1, \dots, n_{d-1})$.

Definition 4 (d-dimensional row-major order): Given a d -image of shape $S = (n_h)$, ($0 \leq h \leq d-1$) and number of d -cells $N = \prod_h n_h$, the mapping $\mu : S \rightarrow \{0, 1, \dots, N - 1\}$ is a combination of multi-indices with integer weights $(w_0, w_1, \dots, w_{d-2}, 1)$, such that:

$$(i_0, i_1, \dots, i_{d-1}) \mapsto i_0 w_0 + i_1 w_1 + \dots + i_{d-1} w_{d-1},$$

with $w_k = n_{k+1} n_{k+2} \dots n_{d-1}$ for $(0 \leq k \leq d - 2)$.

Example 2 (LAR voxels): The general hexahedral 3-cell (with 8 vertices), depending on three indices h, i, j (page, row, column) is obtained as the convex combination of the vertices indexed as integers via the mapping:

$$\begin{aligned} \mu : N_0 \times N_1 \times N_2 &\rightarrow M \\ &= \{m | m \in Z, 0 \leq m \leq n_0 n_1 n_2 - 1\}, \\ (h, i, j) &\mapsto h n_1 n_2 + i n_2 + j, \\ 0 \leq h &\leq n_0, 0 \leq i \leq n_1, \\ 0 \leq j &\leq n_2, \end{aligned}$$

where (n_0, n_1, n_2) correspond respectively to the number of pages, rows, and columns of a 3-dimensional array, called the array shape, and $N_h = \{0, 1, \dots, n_h - 1\}$, ($0 \leq h \leq 2$). Therefore we have, as LAR representation of a 3-cell (voxel):

$$\begin{aligned} \text{cell}[\mu(i, j, k)] &= [\mu(i, j, k), \mu(i + 1, j, k), \mu(i, j + 1, k), \\ &\quad \mu(i, j, k + 1), \mu(i + 1, j + 1, k), \\ &\quad \mu(i + 1, j, k + 1), \mu(i, j + 1, k + 1), \\ &\quad \mu(i + 1, j + 1, k + 1)] \end{aligned}$$

Assuming that vertices are located on a 3D lattice of points with integer coordinates, it is easily seen that an explicit storage of coordinates is not required, thanks to the explicit bijective mapping μ between the ordinal index of cells and the tuples of coordinates of their vertices.

Figure 8a shows our model of a d -image as a *cuboidal grid* with integer coordinates. Every d -cell is identified by a d -tuple of integer coordinates, mapped to a single integer, in order to compute the *basis vector* corresponding to the cell in the linear space C_3 of chains of 3-cells (voxels).

Let us remark that the matrix $[\partial_3]$ of the boundary operator $C_3 \rightarrow C_2$, used to compute the boundary of any possible subset of voxels, i.e., of any vector in the linear space C_3 of 3-chains associated with the image, depends only on the image shape (n_0, n_1, n_2) , and may be computed once for all (choosing a set of standard image shapes), and stored or transmitted accordingly.

Since the bottleneck of GPGPU implementations lies in the moving of data from global to local memory,

our solution is to store the (sparse) matrix operator $[\partial_3]$ of n^3 voxels, with $n \in \{64, 128, 256\}$, in *Constant Memory*, and move just the (binary) *coordinate vectors* of chains in *Private Memory*. The boundary computation is therefore done by partitioning the image, according to the paradigm *divide et impera*, as shown in Figures 8b and 8c.

4.1.1. From multi-index tuples to chain coordinates

A functional implementation of a *tuple* \rightarrow *integer* mapping may use a second-order function, that accepts the *shape* of the image (in order to compute the tuple space of indices of d -cells) in a first application, and then takes a *multi-index tuple* as parameter in a second application.

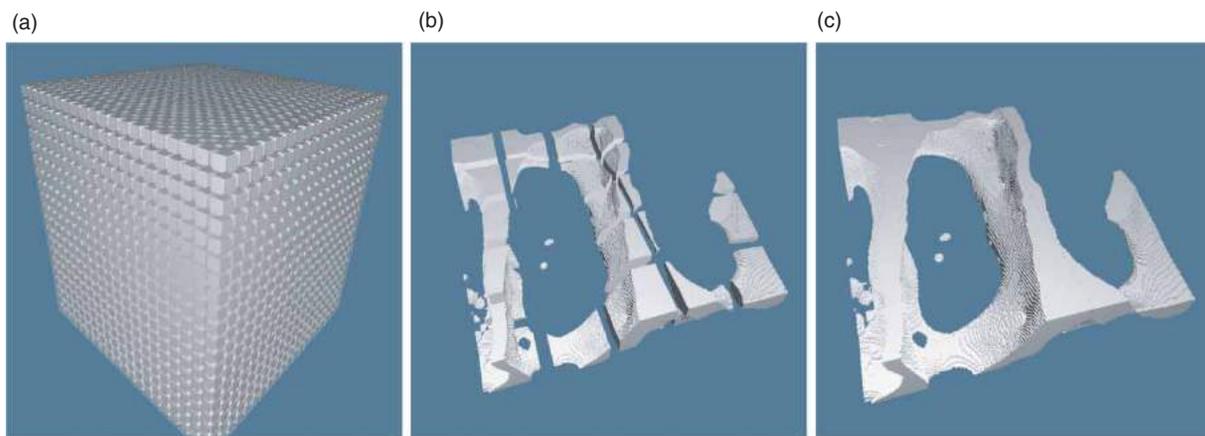


Figure 8. 3D image portions as 3-cell complexes: (a) image portion seen exploded; (b) *divide et impera* paradigm; (c) reconstruction by removal of double cells, via a sort-based *MapReduce* algorithm.

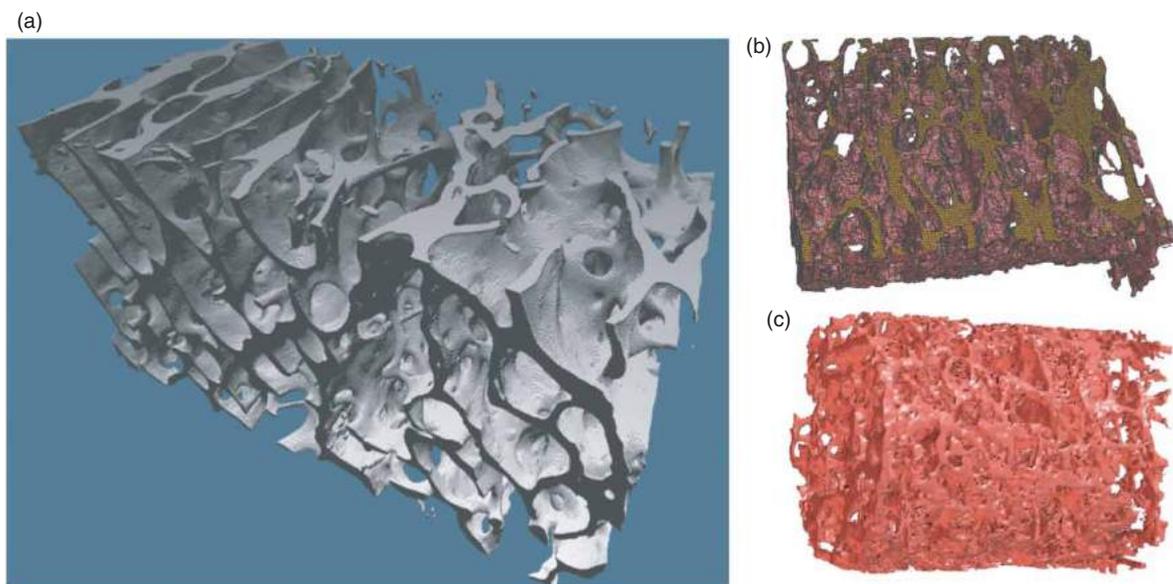


Figure 9. (a) Solid model, closed and topologically correct at the resolution of the image, of a sample of spongy bone, as LAR of the boundary of the chain of solid voxels, computed with the GPGPU support provided by *OpenCL*. Compare it with the open surfaces generated by marching-cubes, or similar algorithms, over the 3D field defined by image intensity (b), (c). Nontrivial difficulties are encountered for closing these imperfect boundaries without topological violations.

This function returns the cell index in the linear address space associated with the given shape.

The set of address tuples of d -cells (i.e. of d -dimensional image elements) within a given *image mask* is mapped to the corresponding set of (single) integers associated with the low-level image elements (pixels or voxels, depending on the image dimension and shape). This *total chain* of the mask window is then filtered to coordinates of image elements of the given *colour* (intensity values) within the considered *image window*, and returned as a list of integer cell indices.

In summary, when using LAR, an image is regarded as a general 3D mesh, with full and straightforward control of 0-,1-,2-, and 3-cells, each identified by a single *signed integer* (the sign denoting the cell orientation in a chain). The boundary of any chain (collection of cells) is computable via a *single SpMV multiplication*. The boundary operator (i.e., its sparse matrix) is stored *once and for all*, as all allied operators (coboundary, gradient, curl, divergence and Laplacian). An image of the boundary model of a sample of human spongy bone, extracted using LAR, is given in Figure 9a. Figures 9b and 9c are conversely produced through a standard iso-surface extraction algorithm.

4.2. Imaging morphology with LAR

In this section we show how to implement the four operators of mathematical morphology, i.e., *dilation*, *erosion*, *opening* and *closing*, by way of matrix operations representing a composition of the linear topological operators of *boundary* and *coboundary* with other incidence relations. We give here just a few hints of these computations. Thanks to its multidimensional nature, the LAR implementation of morphological operators is *dimension-independent*.

Notice that all incidence relations between chains of different dimensions, e.g., FV and EV, can be readily transformed in LAR to *linear operators* with sparse matrices $CSR(M_2)$ and $CSR(M_1)$, respectively (see Section 3.2.3). The linear operators $U: C_d \rightarrow C_{d+1}$ and $D: C_d \rightarrow C_{d-1}$, that we denote respectively as *Up* and *Down*, correspond to the incidence relations FE, EV (Up) and VE, EF (Down) in case of 2D images, and to the incidence relations CF, FE, EV (Up) and VE, EF, FC (Down) in 3D images. Note that, if chains are represented as (binary) column vectors, the relation symbols should be read from right to left: FE stands for “from Edges to Faces”, and so on.

The *morphological gradient* operator is a composition of three basic operators: a dilation, an erosion of the input image and a subtraction of these two results. The (coordinate representation of a) d -chain γ being given as input, we

- (i) compute its boundary $\partial_d(\gamma)$;
- (ii) extract the $(d-2)$ -chain $\varepsilon = (D \circ \partial_d)(\gamma)$;
- (iii) single-out the $(d-1)$ -chain returned from its coboundary $\delta^{d-2}(\varepsilon)$;
- (iv) finally compute the d -chain $\eta := (U \circ \delta^{d-2})(\varepsilon) \subset C_d$.

It is possible to show that $\eta = (\oplus\gamma) - (\ominus\gamma)$, where \oplus , \ominus respectively denote the *dilation* and *erosion* operators, as illustrated in Figures 10, 11 and 10.

Hence, we obtain $\oplus\gamma = \gamma \cup \eta$, and $\ominus\gamma = \gamma - \eta$. The other operators, i.e., *opening* $\oplus \circ \ominus$ and *closing* $\ominus \circ \oplus$, are algebraically computable in the standard way [43]. The component steps of dilatation and erosion operators on a small 2D image portion are displayed and discussed in Figure 12.

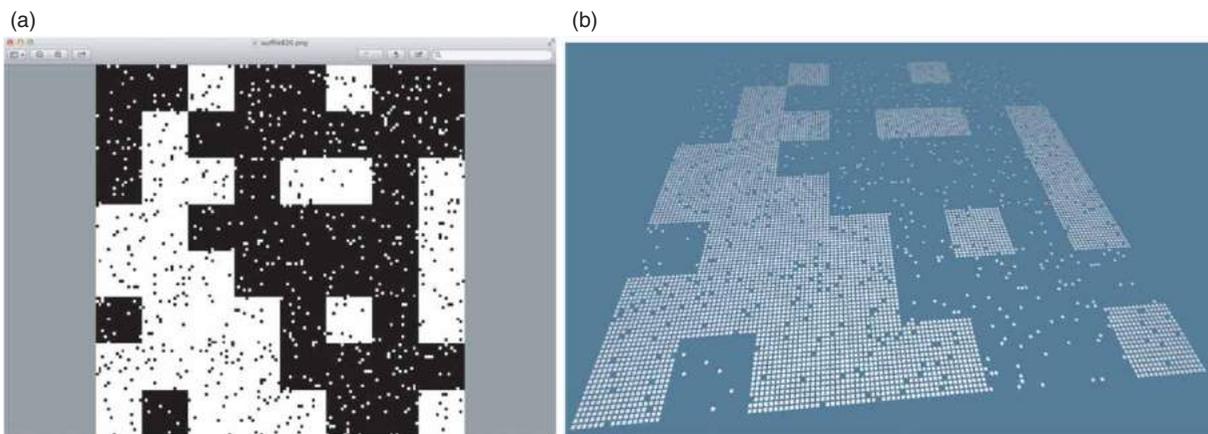


Figure 10. Consider the random chain $\gamma \in C_2$ of white pixels of an image: (a) original PNG image; (b) exploded model of $|\gamma| \subset E^3$

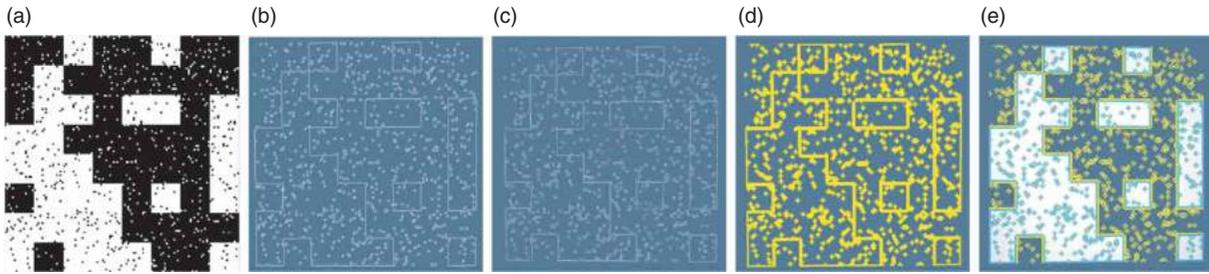


Figure 11. Subimage 128×128 example: (a) input chain $\gamma \in C_2$ (white); (b) extraction of the boundary chain $\beta = \partial_2(\gamma) \in C_1$; (c) chain $\eta = \mathbb{V}\mathbb{E}(\beta) \in C_0$; (d) chain $\beta_2 = \mathbb{F}\mathbb{V}(\eta) \in C_2 \equiv (\mathbb{F}\mathbb{V} \circ \mathbb{V}\mathbb{E} \circ \partial_2)(\gamma)$; (e) from the chain β_2 the dilation component $\beta_2 - \gamma$ (yellow), and the erosion component $\beta_2 \cap \gamma$ (cyan) are obtained.

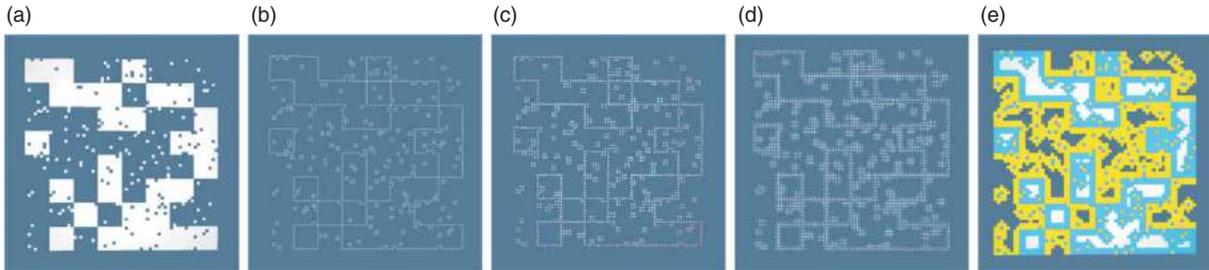


Figure 12. Subimage 64×64 : (a) input chain $\gamma \in C_2$; (b) chain $\beta = \partial_2(\gamma) \in C_1$; (c) chain $(\mathbb{V}\mathbb{E} \circ \partial_2)(\gamma) \in C_0$; (d) $(\mathbb{E}\mathbb{V} \circ \mathbb{V}\mathbb{E} \circ \partial_2)(\gamma) \in C_1$; (e) chain $\beta_2 = (\mathbb{F}\mathbb{E} \circ \mathbb{E}\mathbb{V} \circ \mathbb{V}\mathbb{E} \circ \partial_2)(\gamma) \in C_2$, exhibiting the *dilation* component chain $\text{DIL}(\beta_2)(\gamma) = \beta_2 - \gamma$ (yellow), and the *erosion* component chain $\text{ERO}(\beta_2)(\gamma) = \beta_2 \cap \gamma$ (cyan).

4.3. Extraction of models from images

Biomedical applications require fast performances with big geometric data, for topological tasks such as model extraction from 3D images. In medical images density values 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 models subsequently smoothed out via Laplacian smoothing. This approach has the nice feature that the entire image is partitioned into a set of cochains associated with field values, including the interstitial space, thus providing a well-defined mesh both of the relevant features and of their outer space.

In particular, any portion of a d -image ($2 \leq d \leq 4$) can be seen as a d -chain in the linear space of chains induced by a regular d -cubical decomposition of the bounded d -cuboidal image space, delimited by two extreme picture elements of minimum and maximum multi-indices.

Because of the isomorphism between a d -complex S and its dual S^* , any d -image subset $c \subset S(d)$ can be represented as a 0-chain $c^* \subset S^*(0)$, and stored as a column vector of binary coordinate representation $\text{CSC}(c)$. Since the structure of the space decomposition does not depend on the image content, but only on the shape of the image, the boundary/coboundary matrices of images are computed once for all, and stored/transmitted in CSR format for the most used image shapes.

The stored content of any image chain (subset of image elements—either pixels or voxels) shall be seen as a cochain associated to the given chain, and its discrete integrals (e.g. the volume, or surface area, or inertia moments) or other chains to be computed by means of discrete differential operators, shall be computed accordingly, by the proper SpMspV multiplication, taking appropriate benefits by advanced computational hardware, e.g., by GPGPU methods.

In conclusion, we would like to remark that any model mesh, either of the internal or the external surface, using either unstructured (triangle, tetrahedra) or structured (quadrilaterals, hexaedra) or more general convex cells, can be stored on computer media, or transmitted on communication networks, using LAR as efficient representation of topology and as support for curved geometry.

4.4. Extraction of the liver portal vein system

Venous systems are called *portal* when a capillary bed pools into another capillary bed through veins, without first going through the heart. The hepatic portal system is the system of veins comprising the liver portal vein and its tributaries. The liver is a vital organ of all vertebrates. In turn, hepatic vasculature is essential to the liver function. A good assessment of individual liver vasculature is preliminary to hepatic surgery. While the macroscopic

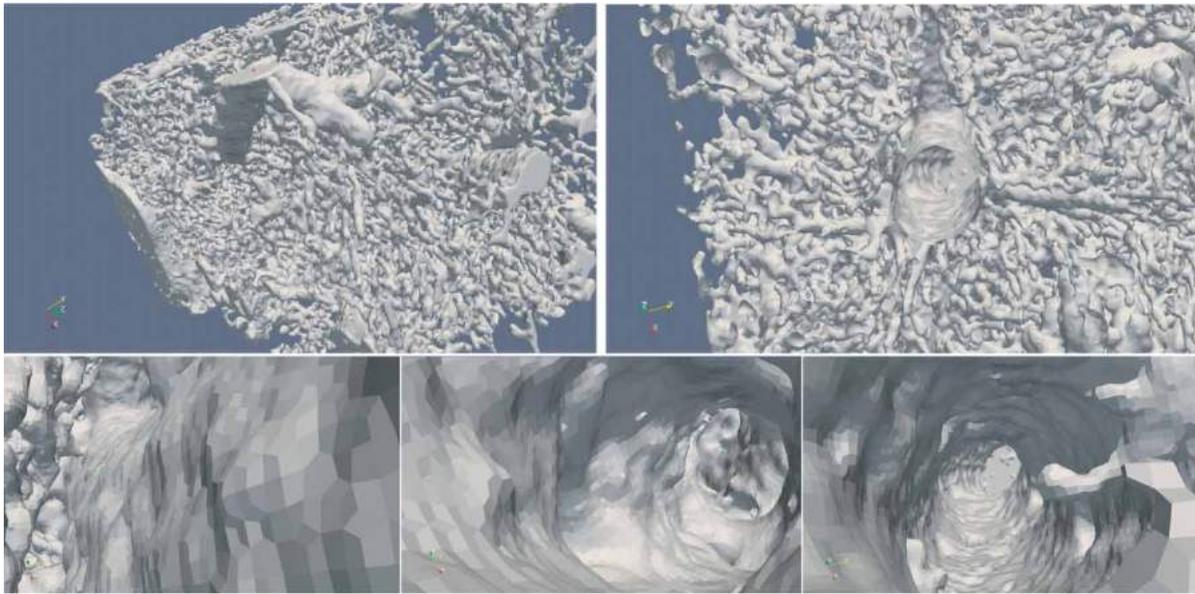


Figure 13. The exterior and a cross-section of the portal vein system through the liver sample (top). Images of a vein interior (bottom). Note the mesh of quadrilaterals.

structure of the hepatic vasculature is well studied, the microvasculature is not yet fully understood.

Several techniques have been applied to investigate models of the liver unit [12], but several issues are still open. Current knowledge is limited, in particular, by the resolution of imaging techniques. From sequential cryosections of autopsy specimens of human liver, [46] reconstructed polyhedral modules of the microvasculature, having 7 to 9 facets, whose size was in the range 0.3 to 0.9 mm. [17] used scanning electron microscopy of vascular corrosion casts to study the liver lobule anatomy. These and other authors have examined only a small piece of liver tissue, comprising just a few lobules.

The present work is part of a collaborative effort with a Czech research team based at the University of West Bohemia and the Charles University, integrating specialists in biomechanics, biophysics, informatics, liver surgery, radiology, and histology. We aim at increasing both the scope and the resolution of 3D liver imaging—an arduous goal, but crucial to enhance our understanding of liver *lobule* anatomy and function.

Data from scanning electron microscopy of corrosive casts of pig liver were used to prepare our input images. Their resolution is $(0.004682 \text{ mm})^3$. The size of the image subset selected for visualization is $370 \times 228 \times 237$. Figure 13 illustrates the extracted 3D model of the portal vein system. Its topology is correct (the boundary representation is closed and valid) at the resolution of the input image, and its geometry is accurate. This solid model will be used to perform computational fluid dynamics (CFD) simulations of blood flow within the *portal system*.

5. Conclusion

This paper demonstrate that LAR—a general-purpose framework for solid and geometric modeling—has the capability of generating topologically valid and geometrically accurate boundary models of complex biological systems extracted from 3D high-resolution imaging. Our prototype implementation of LAR is an integral part of a permanent effort to rethink the foundations of solid modeling, aiming at simplifying and generalizing its data representation and disentangling its main algorithms, in order to produce a computational framework well adapted to the new world of big geometric data over cloud- and web-based infrastructures. This long-term project has already achieved some tangible results in applications to the extraction of solid models from 3D medical images (as documented in this paper) and to the simplified generation of building models for *indoor mapping* and the *Internet of Things* [44].

The prototype implementation (done in *Python*) is currently in progress. We are presently working on a novel approach to Boolean operations on cellular complexes, designed to be multidimensional, variadic and well-suited to distributed implementation over big geometric datasets. The next step will be to consolidate LAR in the form of a *Haskell* library, characterized by a strongly-typed reference implementation with well-defined API and function signatures, suitable for being subsequently compiled to *C* and to *JavaScript*, the language of modern web.

We see great opportunities in this project: (i) LAR uses just arrays of signed integers, instead of complicated data structures, to describe 1D/2D/3D/4D meshes/images and topologies of any sort and size; (ii) whenever necessary, LAR uses distributed algorithms of *MapReduce* kind; (iii) it is based on the well-established conceptual infrastructure of algebraic and combinatorial topology. On the downside, the main threat to this project is a dearth of funding. The prototype implementation underway is adequate for demonstration purposes. However, for LAR to be really successful, it would need a dedicated company and investments enabling the development of an optimized and portable library for general use.

Acknowledgements

A.D.C. and A.P. are grateful to the members [25] of the 3D-Based Medical Application Working Group sponsored by the IEEE Computer Society/Standards Activities Board for their interest and encouragement. They also acknowledge the inspiring cooperation from Vadim Shapiro. Federico Spini, Enrico Marino, and Vittorio Cecchetto contributed to the early stages of the LAR project. A.P. thanks SOGEL, the ICT company of the Italian Ministry of Economy and Finance, for the support provided through several grants.

ORCID

Alberto Paoluzzi  <http://orcid.org/0000-0002-3958-8089>
 Antonio DiCarlo  <http://orcid.org/0000-0002-7736-684X>
 Francesco Furiani  <http://orcid.org/0000-0001-7997-3777>
 Miroslav Jirík  <http://orcid.org/0000-0002-8002-2079>

References

- [1] Ala, S. R.: Performance anomalies in boundary data structures, *IEEE Comput. Graph. Appl.* 12(2), 1992, 49–58. <http://dx.doi.org/10.1109/38.124288>
- [2] Ansaldo, S.; De Floriani, L.; Falcidieno, B.: Geometric modeling of solid objects by using a face adjacency graph representation, In *ACM SIGGRAPH Computer Graphics*, 19, 1985, 131–139. <http://dx.doi.org/10.1145/325165.325218>
- [3] Bajaj, C.; DiCarlo, A.; Paoluzzi, A.: Proto-plasm: A parallel language for scalable modeling of Biosystems, *Philosophical Transactions of the Royal Society A* 366, 1878 (September 2008), 3045–3065. Issue “The virtual physiological human: building a framework for computational biomedicine I”, compiled by Marco Viceconti, Gordon Clapworthy, Peter Coveney and Peter Kohl. <http://dx.doi.org/10.1098/rsta.2008.0076>
- [4] Bajaj, C.; Paoluzzi, A.; Scorzelli, G.: Progressive conversion from B-rep to BSP for streaming geometric modeling, *Computer-Aided Design and Applications*, 3(5), 2006, 577–586.
- [5] Basak, T.: Combinatorial cell complexes and Poincaré duality, *Geometriae Dedicata*, 147(1), 2010, 357–387. <http://dx.doi.org/10.1007/s10711-010-9458-y>
- [6] Baumgart, B. G.: Winged edge polyhedron representation, Tech. Rep. Stan-CS-320, Stanford University, Stanford, CA, USA, 1972.
- [7] Bidarra, R.; Bronsvort, W. F.: Semantic feature modelling, *Computer-Aided Design*, 32(3), 2000, 201–225. [http://dx.doi.org/10.1016/S0010-4485\(99\)00090-1](http://dx.doi.org/10.1016/S0010-4485(99)00090-1)
- [8] Bowyer, A.: *SvLis Set-theoretic Kernel Modeler: Introduction and User Manual*. Information Geometers, 1995.
- [9] Bowyer, A.; Geometric Modelling Society. *Introducing Djinn: A Geometric Interface for Solid Modelling*. Information Geometers [for] the Geometric Modelling Society, 1995.
- [10] Braid, I. C.: The synthesis of solids bounded by many faces, *Commun. ACM*, 18(4), 1975, 209–216. <http://dx.doi.org/10.1145/360715.360727>
- [11] Buluç, A.; Gilbert, J. R.: Parallel sparse matrix-matrix multiplication and indexing: Implementation and experiments, *SIAM Journal of Scientific Computing (SISC)*, 34(4), 2012, 170–191. <http://dx.doi.org/10.1137/110848244>
- [12] Debbaut, C. *Multilevel Modelling of Hepatic Perfusion in Support of Liver Transplantation Strategies*, PhD thesis, Ghent University, 2013.
- [13] DiCarlo, A.; Milicchio, F.; Paoluzzi, A.; Shapiro, V.: Chain-based representations for solid and physical modeling. *Automation Science and Engineering, IEEE Transactions on*, 6, 3 (July 2009), 454–467. <http://dx.doi.org/10.1109/tase.2009.2021342>
- [14] DiCarlo, A.; Paoluzzi, A.; Shapiro, V.: Linear algebraic representation for topological structures, *Computer-Aided Design*, 46, 2014, 269–274. <http://dx.doi.org/10.1016/j.cad.2013.08.044>
- [15] Dobkin, D. P.; Laszlo, M. J.: Primitives for the manipulation of three-dimensional subdivisions. In *Proceedings of the third annual symposium on Computational geometry (New York, NY, USA, 1987)*, SCG '87, ACM, 86–99. <http://dx.doi.org/10.1145/41958.41967>
- [16] Furiani, F.; Paoluzzi, C.; Paoluzzi, A.: Algebraic mining of solid models from images, In *ESA-EUSC-JRC: Image Information Mining 2014 Conference (Romanian Space Agency (ROSA), Bucharest, Romania, March, 5–7 2014)*. <http://dx.doi.org/10.2788/2585210.2788/25852>
- [17] Giuvarasteanu, I.: Scanning electron microscopy of vascular corrosion casts—standard method for studying microvessels, *Romanian Journal of Morphology and Embryology = Revue Roumaine De Morphologie Et Embryologie*, 48(3), 2007, 257–261.
- [18] Gomes, A.; Middleditch, A.; Reade, C.: A mathematical model for boundary representations of n-dimensional geometric objects, In *Proceedings of the fifth ACM symposium on Solid modeling and applications (New York, NY, USA, 1999)*, SMA '99, ACM, 270–277. <http://dx.doi.org/10.1145/304012.304039>
- [19] Guibas, L.; Stolfi, J.: Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, *ACM Trans. Graph.*, 4(2), 1985, 74–123. <http://dx.doi.org/10.1145/282918.282923>
- [20] Hoffmann, C. M.; Kim, K.-J.: Towards valid parametric CAD models, *Computer-Aided Design*, 33(1), 2001, 81–90. [http://dx.doi.org/10.1016/S0010-4485\(00\)00073-7](http://dx.doi.org/10.1016/S0010-4485(00)00073-7)

- [21] Hunter, P.; Chapman, T.; Coveney, P. V.; De Bono, B.; Diaz, V.; Fenner, J.; Frangi, A. F.; Harris, P.; Hose, R.; Kohl, P.; Lawford, P.; McCormack, K.; Mendes, M.; Omholt, S.; Quarteroni, A.; Shublaq, N.; Sk^oar, J.; Stroetmann, K.; Tegner, J.; Thomas, S. R.; Tollis, I.; Tsamardinos, I.; Van Beek, J. H. G. M.; Viceconti, M.: A vision and strategy for the virtual physiological human: 2012 update, *Interface Focus*, 3(2), 2013. <http://dx.doi.org/10.1098/rsfs.2013.0004>
- [22] Hunter, P.; Coveney, P. V.; de Bono, B.; Diaz, V.; Fenner, J.; Frangi, A. F.; Harris, P.; Hose, R.; Kohl, P.; Lawford, P.; McCormack, K.; Mendes, M.; Omholt, S.; Quarteroni, A.; Sk^oar, J.; Tegner, J.; Randall Thomas, S.; Tollis, I.; Tsamardinos, I.; van Beek, J. H. G. M.; Viceconti, M.: A vision and strategy for the virtual physiological human in 2010 and beyond, *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 368(1920), 2010, 2595–2614. <http://dx.doi.org/10.1098/rsta.2010.0048>
- [23] Jirik, M.: Computer-aided Multilevel Analysis of Liver Parenchyma, PhD thesis, University of West Bohemia, 2015.
- [24] Jirik, M.; Ryba, T.; Svobodova, M.; Mirka, H.; Liska, V.: Lisa: Liver surgery analyser software development, In 11th World Congress on Computational Mechanics (WCCM XI) (Barcelona, Spain, 2014).
- [25] Moon, Y. L.; Sugamoto, K.; Paoluzzi, A.; DiCarlo, A.; Kwak, J.; Shin, D. S.; Kim, D. O.; Lee, D. H.; Kim, J.: Standardizing 3D medical imaging, *IEEE Computer*, 47(4), 2014, 76–79. <http://dx.doi.org/10.1109/MC.2014.103>
- [26] Nickerson, D.; Garny, A.; Nielsen, P.; Hunter, P.: Standards and tools supporting collaborative development of the virtual physiological human. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, 2013, 5541–5544. <http://dx.doi.org/10.1109/embc.2013.6610805>
- [27] Paoluzzi, A.: *Geometric Programming for Computer Aided Design*, John Wiley & Sons, Chichester, UK, 2003. <http://dx.doi.org/10.1002/0470013885>
- [28] Paoluzzi, A.; Bernardini, F.; Cattani, C.; Ferrucci, V.: Dimension-independent modeling with simplicial complexes, *ACM Trans. Graph.*, 12(1), 1993, 56–102. <http://dx.doi.org/10.1145/169728.169719>
- [29] Paoluzzi, A.; Pascucci, V.; Vicentino, M.: Geometric programming: a programming approach to geometric design, *ACM Trans. Graph.*, 14(3), 1995, 266–306. <http://dx.doi.org/10.1145/212332.212349>
- [30] Paoluzzi, A.; Ramella, M.; Santarelli, A.: Boolean algebra over linear polyhedral, *Computer-Aided Design*, 21(10), 1989, 474–484. [http://dx.doi.org/10.1016/0010-4485\(89\)90055-9](http://dx.doi.org/10.1016/0010-4485(89)90055-9)
- [31] Pascucci, V.; Ferrucci, V.; Paoluzzi, A.: Dimension-independent convex-cell based HPC: representation scheme and implementation issues, In *Proceedings of the third ACM Symposium on Solid Modeling and Applications* (New York, NY, USA, 1995), SMA '95, ACM, 163–174. <http://dx.doi.org/10.1145/218013.218055>
- [32] Pratt, M. J.; Anderson, B. D.: A shape modelling api for the STEP standard, In *Fourteenth International Conference on Atomic Physics*, 1994, 1–7. [http://dx.doi.org/10.1016/s0010-4485\(01\)00052-5](http://dx.doi.org/10.1016/s0010-4485(01)00052-5)
- [33] Raghothama, S.; Shapiro, V.: Boundary representation deformation in parametric solid modeling, *ACM Trans. Graph.*, 17(4), 1998, 259–286. <http://dx.doi.org/10.1145/293145.293148>
- [34] Raghothama, S.; Shapiro, V.: Consistent updates in dual representation systems, In *Proceedings of the fifth ACM symposium on Solid modeling and applications* (New York, NY, USA, 1999), SMA '99, ACM, 65–75. <http://dx.doi.org/10.1145/304012.304019>
- [35] Requicha, A. A. G.; Voelcker, H. B.: *Constructive solid geometry*, Tech. Rep. TM-25, Production Automation Project, Univ. of Rochester, 1977.
- [36] Requicha, A. G.: Representations for rigid solids: Theory, methods, and systems, *ACM Comput. Surv.*, 12(4), 1980, 437–464. <http://dx.doi.org/10.1145/356827.356833>
- [37] Ricken, T.; Dahmen, U.; Dirsch, O.: A biphasic model for sinusoidal liver perfusion remodeling after outflow obstruction, *Biomechanics and Modeling in Mechanobiology*, 9(4), 2010, 435–450. <http://dx.doi.org/10.1007/s10237-009-0186-x>
- [38] Rossignac, J. R.; O'Connor, M. A.: SGC: a dimension-independent model for point sets with internal structures and incomplete boundaries, In *Geometric modeling for product engineering*, North-Holland, 1990.
- [39] Rossignac, J. R.; Requicha, A. A. G.: Constructive non-regularized geometry, *Computer-Aided Design*, 23(1), 1991, 21–32. [http://dx.doi.org/10.1016/0010-4485\(91\)90078-B](http://dx.doi.org/10.1016/0010-4485(91)90078-B)
- [40] Shapiro, V.: Representations of semi-algebraic sets in finite algebras generated by space decompositions, PhD thesis, Cornell University, Ithaca, NY, USA, 1991. UMI Order No. GAX91-31407.
- [41] Shapiro, V.; Vossler, D. L.: What is a parametric family of solids? In *Proceedings of the third ACM symposium on Solid modeling and applications* (New York, NY, USA, 1995), SMA '95, ACM, 43–54. <http://dx.doi.org/10.1145/218013.218029>
- [42] Silva, C. E.: Alternative definitions of faces in boundary representations of solid objects, Tech. Rep. TM-36, Production Automation Project, Univ. of Rochester, 1981.
- [43] Soille, P.: *Morphological Image Analysis: Principles and Applications*, 2nd ed. Springer-Verlag New York, Inc.; Secaucus, NJ, USA, 2003. <http://dx.doi.org/10.1007/978-3-662-05088-0>
- [44] Spini, F.; Sportillo, M.; Virgadamo, M.; Marino, E.; Bottaro, A.; Paoluzzi, A.: HIJSON: Cartographic Document for Web Modeling of Interactive Indoor Mapping. In *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference (2015)*, A. Giachetti; S. Biasotti; M. Tarini, Eds.: The Eurographics Association. <http://dx.doi.org/10.2312/stag.20151290>
- [45] Tavares, A.; De Cruz, M.; Fernandes, S.: Augmented reality in collaborative virtual environment for discrete event systems modeling and simulation, In *14th Symposium on Virtual and Augmented Reality*, 2012, 155–164. <http://dx.doi.org/10.1109/svr.2012.32>
- [46] Teutsch, H. F.: The modular microarchitecture of human liver, *Hepatology*, 42(2), 2005, 317–325. <http://dx.doi.org/10.1002/hep.20764>
- [47] Viceconti, M.: Collaborative modeling and simulation: The virtual physiological human vision. In *Enabling Technologies: 20th IEEE International Workshops on Infrastructure for Collaborative*

- Enterprises, 2011, 229–234. <http://dx.doi.org/10.1109/wetice.2011.77>
- [48] Weiler, K. J.: Edge-based data structures for solid modeling in curved-surface environments, IEEE Computer Graphics and Applications, 5(1), 1985, 21–40. <http://dx.doi.org/10.1109/MCG.1985.276271>
- [49] Williams, S.; Oliker, L.; Vuduc, R.; Shalf, J.; Yelick, K.; Demmel, J.: Optimization of sparse matrix-vector multiplication on emerging multicore platforms. In Proceedings of the 2007 ACM/IEEE conference on Supercomputing (New York, NY, USA, 2007), SC '07, ACM, 38:1–38:12. <http://dx.doi.org/10.1145/1362622.1362674>
- [50] Woo, T.: A combinatorial analysis of boundary data structure schemata, Computer Graphics & Applications, IEEE 5, 3 (March 1985), 19–27. <http://dx.doi.org/10.1109/MCG.1985.276337>