Title:
Next Generation BIM From Point Clouds in Julia

Authors:
Maria Teresa Graziano, marteresa28@gmail.com, Roma Tre University
Michele Vicentino, mvicentino@geoweb.it, Geoweb, Rome, Italy
Alberto Paoluzzi, alberto.paoluzzi@uniroma3.it, Roma Tre University
Antonio Bottaro, abottaro@geoweb.it, Geoweb, Rome, Italy

Keywords:
Point Cloud, BIM, Potree, LAR, Boolean Algebra, VectMapTree

DOI: 10.14733/cadconfP.2021.359-363

Introduction:

This article introduces a new approach to reconstructing the shapes of buildings and their parts, starting from already registered point clouds generated by any instrumentation, such as 3D laser scanners and LiDAR technology and techniques, from ground, drones, and satellites. When points are acquired both on outside and inside surfaces (fast and low-cost technologies are available), our methods allow for a geometric reconstruction of interior spaces and hierarchical modeling of building subsystems and components.

In this approach, registered point clouds are stored, viewed interactively on the web, and grouped using out-of-core potrees. Subtrees are filtered w.r.t. the algebraic varieties to which they belong. The boundary of each 2D sub-variety is generated by the product of a chain of cells with a sparse boundary matrix. Finally, the 3D arrangement produced by discovered polygons provides the generators (the basis) of a linear space of solid objects. The solid basis is produced by linear algebraic operations using the chain operators generated from the point cloud dataset. According to their Euler tensor matrices, the discovered 3D building blocks are finally classified as partially ordered elements of a finite Boolean algebra and assembled into functional subsystems of the building fabric, envelope, and structure.

These new methods, well based on geometric and topological concepts and advanced geometric and solid modeling techniques, are being implementd on a set of Julia's open-source prototype packages.

A short synthesis of this paper:

First, we need to remind some background notions about 3D point clouds, and in particular about potrees (partially ordered octrees) and their construction from datasets generated through remote sensing methods. Then we show a simple way to discovering the geometrical structure of such datasets. An algorithm, actually a simple variation of postorder potree traversal, is introduced to discover the whole set of plane equations to which the scanned points belong and classify the data as belonging to affine 2D or 1D covectors, or to their intersection points. Then, the topological structure of the point cloud is elucidated by computing the (co)chain complex of the space partition generated by the input clouds.

In particular, we exploit the fact that a partition of the whole space is both an algebraic lattice and a finite Boolean algebra, whose generators (atoms, or basis) correspond one-to-one to the columns of the boundary matrices previously computed. The computations of Euler matrices of each solid block, given

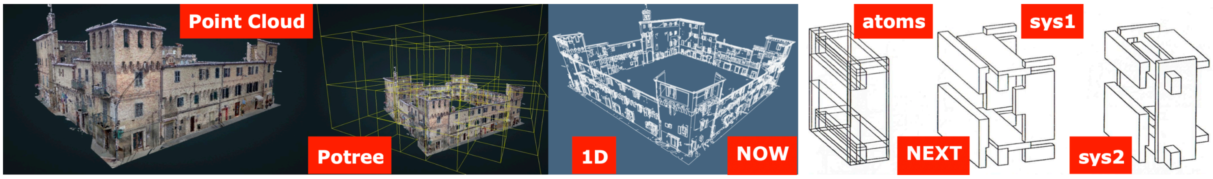

Fig. 1: (a) ; (a) 3D point cloud; (b) the potree; (c) 1D model according to current technology ; (d) atoms of finite Boolean Algebra; (e) two solid models of BIM type, according to different assembly of atoms.

by volume integrals of the coordinate field of $\mathbb{E}^3$, over such chains, finally provides strong clues about the set-membership of each solid atom w.r.t. the building subsystems of spaces, bearing structure, envelope, horizontal or vertical space partitions, and communication components. After providing a short synthesis of new ideas introduced in this paper, the conclusion section compares the previous approaches against the better features of our methods. It indicates also where more research is yet needed.

Roots of this approach:

This paper is grounded on decades of research in geometric and solid modeling. A.P. (pioneer of SMA, 217) in the '80s wrote software tools at Engineering School in Rome, for industrialized methods in design and construction of public housing, school, and hospital programs. New design algorithms, languages, methods, and systems [1] also came from his CAD/PLM Labs (IBM SUR Award 2004). *Recent breakthroughs* [2, 3] of applied computer science discovered a sparse matrix computing of geometry, topology, and algebra of arrangements of $\mathbb{E}^3$, i.e., the foundations of the approach.

Mathematical Background:

A short synthesis of the mathematical concepts this paper relies on is given here. The interested reader may find an introduction and several examples in [2] and [3]. In particular, we use a decompositive representation [4] with cell complexes, using a quite general definition of cells, which may be non-convex and contain internal holes [5]. Our main algebraic topological tools are "chain complexes", linear spaces of "chains" that we identify with cell *subsets* of the same dimension: 0D for points, 1D for polylines, 2D for polygons, 3D for solid cells. Therefore, we have [2] linear chain spaces C_0, C_1, C_2, and C_3 of 0-, 1-, 2-, 3-chains, repectively, connected by linear "boundary" $\partial_p : C_p \to C_{p-1}$ maps and "coboundary" $\delta_p : C_p \to C_{p+1}$ maps. We identify the chain spaces with their dual spaces of "cochains" $C^p : C_p \to \mathcal{F}$, where $\mathcal{F}$ is a field of coefficients, so that it is possible to set $\delta^p = \partial_{p+1}^\top$, and write a chain complex as:

$$C_\bullet = (C_p, \partial_p) := C_3 \underset{\partial_3}{\overset{\delta_2}{\leftrightarrows}} C_2 \underset{\partial_2}{\overset{\delta_1}{\leftrightarrows}} C_1 \underset{\partial_1}{\overset{\delta_0}{\leftrightarrows}} C_0.$$

A p-chain basis is an ordered set of singleton p-chains, one-to-one mapping to p-cells of a cellular complex. After having fixed a basis, the linearity of chain spaces allows to uniquely express *any* chain as a linear combination of the basis elements with coefficients from the field. The chain spaces are so parametrized by tuples of coordinates from $\mathcal{F} = \{0, 1\}$ (for *unoriented* chains) or from $\mathcal{F} = \{-1, 0, 1\}$ (for *oriented* chains). In conclusion, our algebraic-topological calculus of geometric shapes is reduced to simple algebraic operations (sum, product, etc.) between sparse matrices (for the operators) and/or sparse vectors (for the chain coordinates). It was amazing to discover [3] that the colums of ∂_3 are bijective with generators of a finite Boolean algebra for solid geometry. This formal apparatus is similar to some used in recent developments of data science and neural nets, and it seems fair to imagine that in upcoming times our methods will be considered as belonging to the larger Data Science field.

Clouds of Discrete Points:

Point Clouds have been scanned, since thirty years ago, at ever-increasing resolutions on the surfaces of both artificial and natural objects and scenes. Point acquisition may use very different technologies, ranging from low-range camera snapshots to multidimensional imaging from drones or space, using electromagnetic radiation (structured laser lights of high spatial coherence) with high-frequency pulsed light (LiDAR) rather than ultrasonic radio waves or shots from a combination of cameras. Other scanners, mainly in medical applications and underground inspections, both terrestrial and space-based, may return multidimensional information, also characterized as big data sets of tensor arranged points. Point Cloud is the term that best summarizes the commonalities of such broad categories of big datasets.

Each point carries three colors other than three coordinates for its position vector with respect to some fixed coordinate frame. The cardinality of a set of shots, captured with different camera positions and orientations, may normally vary from tens of thousands to many billions of points. Therefore, a point cloud can be considered a huge set of discrete 6D points embedded in 3D Euclidean space. The registration, performed with automatic techniques at acquisition time or later in the background, moves all point subsets into only one coordinate system. In this paper, we consider already registered point clouds of whatever size. The color is not taken into account here, even if later it may greatly contribute to the realism of the reconstructed geometries. We start by efficiently discovering the equations of the set of 3D planes the points belong to, within a variable range of numerical error, which is always present.

Geometrization of Point Subsets:

Let us visualize a *potree* [6, 7] as a huge abstract tree where each node is a container of many points (order of hundreds or thousands) and where non-leafs have at most 8 sons. Points are located within a member of the octet of 3D semispaces defined by three orthogonal planes and associated with the corner volumes of the father node. This paper's main idea allows for rapid discovery of all planes of points by searching each (fairly small) subset of points associated with the leaves of the cloud potree. The key observation is that all potree leaves contain a small number of coplanar sets of points in a small space volume. Therefore, all planes are easily discovered by searching only the potree leaves and using a greedy algorithm. From this picture, it is easy to imagine that points in a leaf stay in a spatial region of minimal size and hence are located on a fairly small number of inner or outer surfaces, typically on a very small number of planes.

Given the ordered *potree* of a point cloud, an isomorphic ordered tree, called *CoMapTree* (see next section) can be constructed inductively by a postorder traversal, in such a way that each node contains the signatures of planes to which the node points belong to. Of course, the leaves would contain a tiny number of plane signatures, whereas each node would contain the union of signatures of its sons. A great degree of redundancy will be here efficiently recognized and eliminated. In fact, the same plane might be contained in various descendants of a node, and these plane instances should be unified by a single signature before continuing the traversal. The root node will, of course, contain a single signature for each plane generated by the point cloud hierarchy traversed.

Geometric Packing:

There are few main kind of operations to be executed during the potree traversal. They are listed below.

Identification An α-complex [8], with a small α value, is computed for the points of each leaf node. So the points are clustered into disjoint sets of adjacent triangles. Then, a greedy procedure starts from a triangle identifying a plane and moves to adjacent coplanar points. When the cluster ends, the plane equation is recomputed by least-squares minimization. The best fit in the least-squares sense minimizes the sum of squared residuals. This is repeated on each cluster, finally identifying all the covectors fitting the data subset.

Unification In logic and computer science, unification is an algorithmic process of solving equations.

Proceedings of CAD'21, Barcelona, Spain, July 5-7, 2021, 358-363

A solution to a unification problem is called a substitution. A unification algorithm should compute here a complete and minimal substitution set for a given pattern of covectors. That is a set covering all its solutions and containing no redundant members. In particular, starting from two or more covectors, it should compute a minimal set of unit normal vectors as keys of different dictionaries and include the residuals of the parallel equations in the collection of values associated with each normal.

Resolustion When solving a system of (linear) equations in three variables x, y, z, denoting affine varieties in 3D, we may have (a) no solutions, when at least two of them are parallel, and hence are unified (see above); (b) an infinite number of solutions, depending on two parameters when the sets are coincident, and just one of the equations is not eliminated; (c) an infinite number depending on one parameter when they intersect, so that the covectors are grouped and written in normalized parametric form; (d) or just a single solution, i.e., the intersection point of a parametric line against a covector.

2D Arrangement of affine planes:

Given a potree P, an inorder traversal of its CoMapTree $\Pi(P)$ will extract a collection of undirected planar graphs $G(P) := \{\pi_k\}$, where $\pi_k = (V_k, E_k)$ with vertices V_k and edges E_k embedded in planes $\pi \in \Pi(P)$.

First, we compute the 2-skeleton $X = (X_0, X_1, X_2)$ of the space partition induced by $\Pi(P)$ in $\mathbb{E}^3$, i.e. its set of 2-, 1-, and 0-cells (faces, edges, and vertices) generated by intersections of π planes within the 3D volumes of $\Pi(P)$. For this purpose we calculate the *chain 2-complex* $C_\bullet = (C_p, \partial_p)$, $p = 0, 1, 2$.

Arrangement of 2-cells in each plane Let be given a collection of planar graphs $g_k = (V_k, E_k)$, with $E_k = (v_{k_i}, v_{k_j})$. The planar arrangement of each $\Pi(g_k)$ plane [2] is generated by: (a) mapping the plane to $z = 0$ semispace; (b) computing each 2D arrangement $\mathcal{A}(E_k)$; (c) assembling the output coboundary matrices $[\delta_0^k], [\delta_1^k]$ as diagonale blocks of the sparse accumulator matrices Δ_0, Δ_1, respectively [9].

Congruence; generation of 2.5D skeleton It is easy to see that the boundaries of planar graphs, when mapped back in 3D, may have distinct but coincident vertices and edges. Hence, to get a 2-complex in 3D without cell repetitions and redundancy, we need to enforce the boundary compatibility of all p-cells, $p = 0, 1, 2$. This action is performed by enforcing the algorithm to consolidate congruence [9], mapping the assembled block-diagonal matrices $[\Delta_0], [\Delta_1]$ to those of global operators δ_0, δ_1. Looking to this dataset from a graphics viewpoint, they correspond to a 2.5D vector model of the scanned building.

3D Arrangement of solid blocks:

In the next step we complete our plan, assessing the yet unknown linear space C_3 from C_0, C_1, C_2 and δ_2 from δ_0, δ_1, so discovering the full topological structure of the built environment from which the point cloud P was generated. The very last step will be using the C_3 basis vectors as generators of a finite Boolean algebra, from whence to get a solid model of the whole built environment, and a characterization of the construction subsystems operable from any Building Information Modeling system. It is interesting to notice that every basis element of C_3 space, generated as an oriented B-rep by TGW algorithm [2], is also a generator (atom) of the finite Boolean algebra induced in $\mathbb{E}^3$ as the arrangement of the C_2 basis.

Hierarchical and/or functional modeling:

All such 3D atoms are also irreducible elements of a finite algebraic lattice isomorphic to 2^{U_3}, the set of all subsets of U_3, i.e., the basis of C_3 chain space. The partially ordered lattice is closed with respect to the *join* and *meet* operations. These, coinciding with Boolean union and intersection, induce a partial order (and the respective inverse order) such that the result of the operation for any two elements is the least upper bound (or greatest lower bound) of the elements with respect to this partial order.

It is worthwhile to split this lattice into two sublattices, corresponding to the system of architectural (empty) spaces of the reconstructed building and its (solid) constructive elements. The word "system" [] is used here as an arrangement of things or a group of related items working toward a common goal, like collecting components where the whole is greater than the sum of its parts. That means that one cannot understand the parts of something without understanding the thing as a whole, a concept common in architectural and engineering language and speech. The functional building subsystems concern the bearing structure, the external envelope, the vertical and horizontal space partitions, and the communication elements [10].

Providing topological 3D structure:

Once the $[\delta_0]$ and $[\delta_1]$ coboundary matrices have been computed, it is a simple matter to apply the Topological Gift Wrapping (TGW) in 3D to compute the matrix of the boundary operator $\partial_3 = \delta_2^\top$ from $\delta_2 : C_2 \to C_3$. This matrix contains by columns the basis of linear space C_3, with each element represented as a 2-cycle, i.e., as a closed 2-chain, which is a 2-chain without boundary (a closed shell surface). In other words, the matrices $[\delta_0]$, $[\delta_1]$, and $[\delta_2]$ shown below contain implicitly the boundary representations of all solid objects and empty spaces contained in the point cloud. Such gometric models are easy and fast to recover, exporting the result in any standard graphics format.

Conclusions:

The algorithmic pipeline introduced here allows to transform external and internal point clouds of built environments into solid models of the construction, categorized by fabric subsystems, and to get an interactively traversable model of the building internal spaces. The first ones may be used, e.g., for a large program planning of reinforcement against earthquakes of existing schools, hospitals, etc., the last ones for rebuilding and refurbishment of commerce buildings and/or used for virtual and augmented reality applications. Examples and pseudocodes will be shown in the complete paper.

Acknowledgement:

This research was supported in part by grants and scholarships from Geoweb, owned by CNG (Italian National Council of Surveyors), also geo-mapping subsidiary of Sogei, the ICT company fully owned by the Italian Ministry of Economy and Finance.

References:

[1] Paoluzzi, A.: Geometric Programming for Computer Aided Design, John Wiley & Sons, Chichester, UK, 2003. https://doi.org/10.1002/0470013885

[2] Paoluzzi, A.; Shapiro, V.; DiCarlo, A.: Furiani, F.; Martella, G.; Scorzelli, G.: Topological computing of arrangements with (co)chains, *ACM Trans. Spatial Algorithms Syst.*, vol. 7, no. 1, Oct. 2020. https://doi.org/10.1145/3401988

[3] Paoluzzi, A.; Shapiro, V.; DiCarlo, A.; Scorzelli, G.: and Elia Onofri, Finite Boolean Algebras for Solid Geometry using Julia's Sparse Arrays, *arXiv e-prints*, https://arxiv.org/abs/1910.11848

[4] Requicha, A.A.G.: Representations for rigid solids: Theory, methods and systems, *ACM Computing Surveys*, vol. 12, no. 4, pp. 437–464, Dec. 1980. https://doi.org/10.1145/356827.356833

[5] DiCarlo, A.; Paoluzzi, A.; Shapiro, V.: Linear algebraic representation for topological structures, *Computer-Aided Design*, vol. 46, pp. 269–274, 01 2014. https://doi.org/10.1016/j.cad.2013.08.044

[6] Schütz, M.: *Potree: Rendering Large Point Clouds in Web Browsers*, Ph.D. thesis, 09 2016. https://www.cg.tuwien.ac.at/research/publications/2016/SCHUETZ-2016-POT/

[7] Schütz, M.; Ohrhallinger, S.; and Wimmer, M.: Fast out-of-core octree generation for massive point clouds, *Computer Graphics Forum*, vol. 39, no. 7, pp. 155–167, November 2020. https://doi.org/10.1111/cgf.14134

[8] Edelsbrunner, H.; and Mücke, E.P.: Three-dimensional alpha shapes, *ACM Trans. Graph.*, vol. 13, no. 1, pp. 43–72, Jan. 1994. https://doi.org/10.1111/cgf.14134

[9] DelMonte, G.; Onofri, E.; Scorzelli, G.; Paoluzzi, A.: Local congruence of chain complexes, *arXiv e-prints*, https://arxiv.org/abs/2004.00046, Mar. 2020.

[10] Carrara, G.; Paoluzzi, A.: A system approach to building program planning, Tech. Rep. CABD LAB RR.80-02, Inst. of Architecture, Building and Construction, School of Engineering, La Sapienza, Rome, Italy, 1980.