Parallel \& Distibuted Computing: Lecture 12

Alberto Paoluzzi

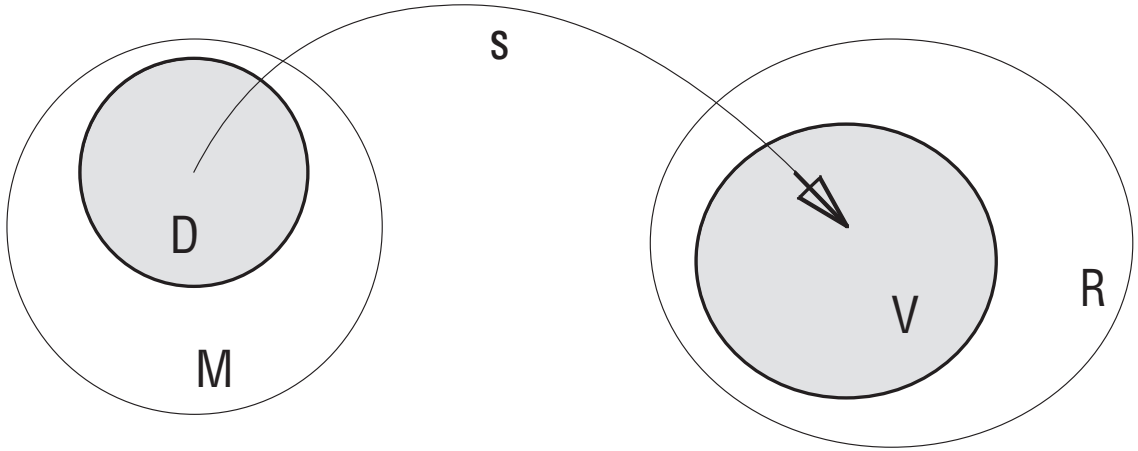
November 27, 2018

Topological computing with Sparse Arrays

LAR scheme

The concept of representation scheme

mapping $s : M \rightarrow R$ from a space of math models M to computer representations R



1. The set M contains the mathematical models of the class of solid objects that the scheme aims to represent.
2. The set R contains symbolic representations, i.e. suitable data structures, built according to some appropriate computer grammar.

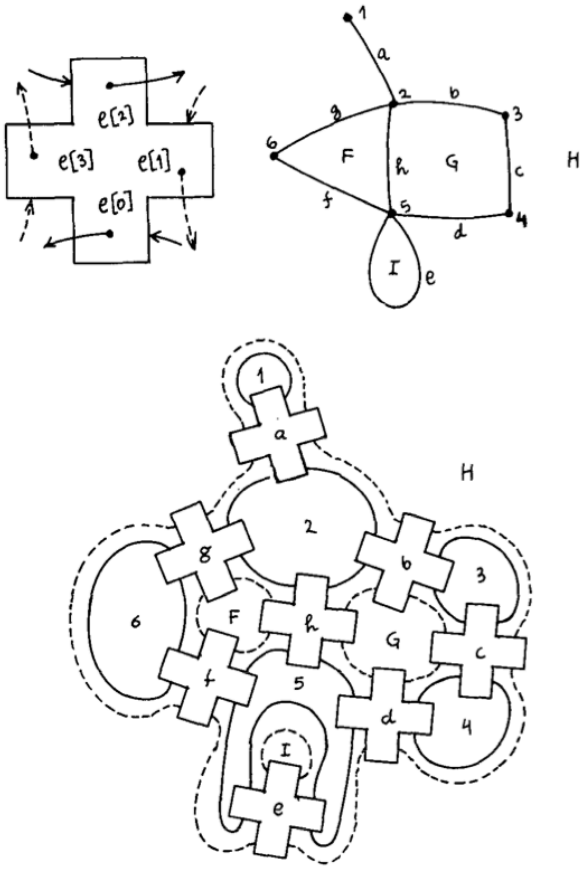
Representation schemes (a long list !!)

Most of such papers introduce or discuss one or more representation schemes ...

1. Requicha, ACM Comput. Surv., 1980 [Req80]
2. Requicha & Voelcker, PEP TM-25, 1977, [RV77]
3. Rossignac & Requicha, Comput. Aided Des., 1991, [RR91]
4. Bowyer, SVLIS, 1994, [Bow95]
5. Baumgart, Stan-CS-320, 1972, [Bau72]
6. Braid, Commun. ACM, 1975, [Bra75]
7. Dobkin & Laszlo, ACM SCG, 1987, [DL87]
8. Guibas & Stolfi, ACM Trans. Graph., 1985, [GS85]
9. Woo, IEEE Comp. Graph. & Appl., 1985, [Woo85]
10. Yamaguchi & Kimura, Comp. Graph. & Appl., 1995, [YK95]
11. Gursoz & Choi & Prinz, Geom.Mod., 1990, [WTP90]
12. S.S.Lee & K.Lee, ACM SMA, 2001, [LL01]
13. Rossignac & O'Connor, IFIP WG 5.2, 1988, [RO90]
14. Weiler, IEEE Comp. Graph. & Appl., 1985, [Wei85]
15. Silva, Rochester, PEP TM-36, 1981, [Sil81]
16. Shapiro, Cornell Ph.D Th., 1991, [Sha91]
17. Paoluzzi et al., ACM Trans. Graph., 1993, [PBCF93]
18. Pratt & Anderson, ICAP, 1994, [PA94]
19. Bowyer, Djinn, 1995, [BS95]
20. Gomes et al., ACM SMA, 1999, [GMR99]
21. Raghothama & Shapiro, ACM Trans. Graph., 1998, [RS98]
22. Shapiro & Vossler, ACM SMA, 1995, [SV95]
23. Hoffmann & Kim, Comput. Aided Des., 2001, [HK01]
24. Raghothama & Shapiro, ACM SMA, 1999, [RS99]
25. DiCarlo et al., IEEE TASE, 2008, [DMPS09]
26. Bajaj et al., CAD&A, 2006, [BPS06]
27. Pascucci et al., ACM SMA, 1995, [PFP95]
28. Paoluzzi et al., ACM Trans. Graph., 1995, [PPV95]
29. Paoluzzi et al., Comput. Aided Des., 1989, [PRS89]
30. Ala, IEEE Comput. Graph. Appl., 1992, [Ala92]

Quad-Edge data structure

(Guibas & Stolfi, ACM Transactions on Graphics, 1985)



(a) Edge record showing Next links.
(b) A subdivision of the sphere.
(c) Data structure for the subdivision

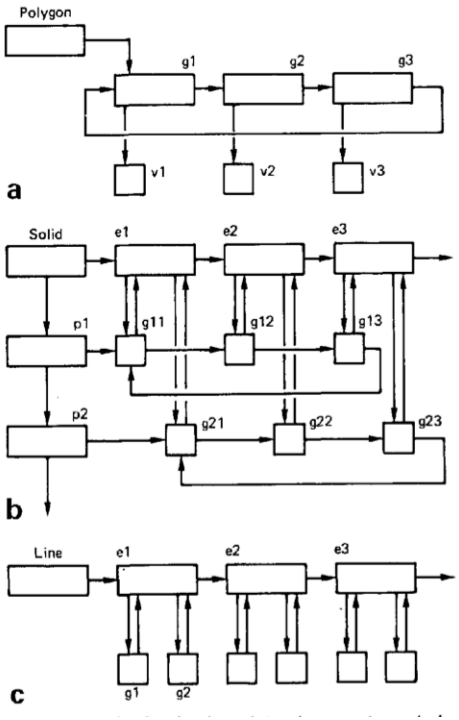
Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams

largely used in computational geometry algorithms and in geometric libraries

Hybrid Edge data structure

(Kalay, Computer-Aided Design, 1989)

a topological data structure for vertically integrated geometric modelling



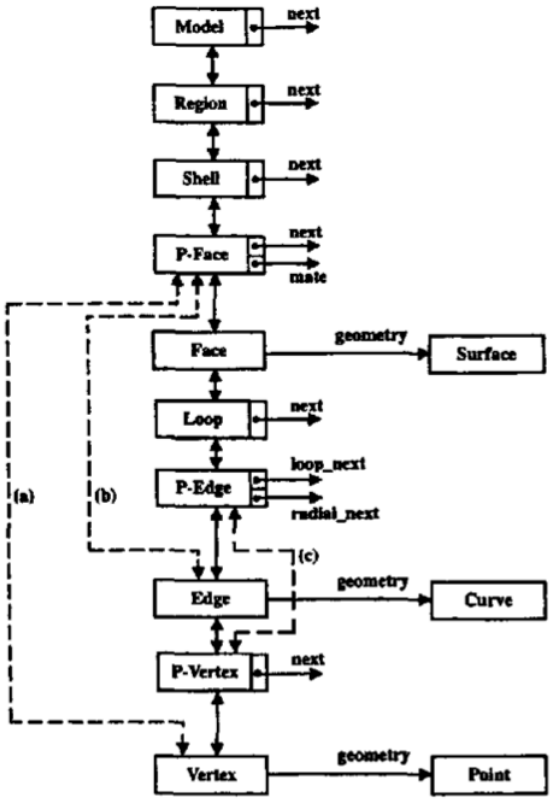

```
type   vptr   = ^vertex;
       gptr   = ^segment;
       pptr   = ^polygon;
       eptr   = ^edge;
       sptr   = ^solid;
       lptr   = ^lines;

solid  = record
                snext     : sptr;
                elist     : eptr;
                plist     : pptr;
                case cavity : boolean
                  false : (child  : sptr)
                  true  : (parent : sptr)
         end;

lines  = record
                lnext     : lptr;
                edge      : eptr;
         end;
```

```
vertex  = record
                form      : point
          end;

segment = record
                gnext     : gptr;
                form      : line;
                gpoly     : pptr;
                gedge     : eptr;
                gvert     : vptr
          end;

polygon = record
                pnext     : pptr;
                glist     : gptr;
                form      : plane;
                case hole : boolean of
                  false   : (child  : pptr);
                  true    : (parent : pptr)
          end;

edge    = record
                enext     : eptr;
                eseg      : array [1..2] of gptr;
                esolid    : sptr
          end;
```

Partial-Entity data structure

(Sang Hun Lee & Kunwoo Lee, ACM Solid Modeling, 2001)

Compact Non-Manifold Boundary Representation Based on Partial Topological Entities

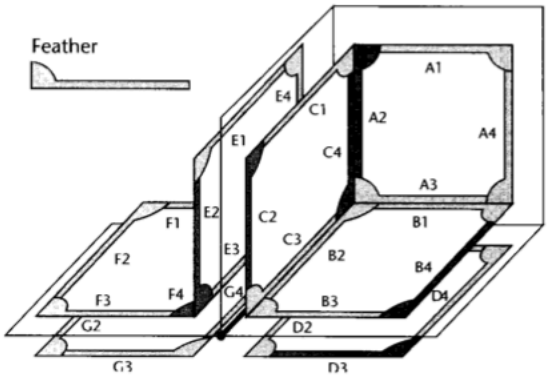



```
class Entity {
   int        _id;
   Attribute  *_attribute;
};
class Model : public Entity {
   Model      *_next;          // next model
   Region     *_region;        // list of regions
};
class Region : public Entity {
   Region     *_next;          // link field of the region list of a model
   Model      *_model;         // parent model
   Shell      *_shell;         // peripheral shell
};
class Shell : public Entity {
   Shell      *_next;          // next void shell
   Region     *_region;        // parent region
   Pface      *_pface;         // partial face
};
class Pface : public Entity {  // partial face (p-face) class
   Pface      *_next;          // next p-face
   Shell      *_shell;         // parent shell
   Entity     *_child;         // child entity: a face, an edge, or a vert
   Orient     _orient;         // orientation flag w.r.t. the face normal
   Pface      *_mate;          // mate p-face
};
class Face : public Entity {
   Pface      *_pface;         // one of two incident p-faces
   Loop       *_loop;          // peripheral loop
   Surface    *_geometry;      // surface
};
```

```
class Loop : public Entity {
   Loop       *_next;          // next hole loop
   Face       *_face;          // parent face
   Pedge      *_pedge;         // a p-edge in a loop
};
class Pedge : public Entity {  // partial edge (p-edge) class
   Loop       *_loop;          // parent loop
   Entity     *_child;         // child entity: an edge or a p-vertex
   Orient     *_orient;        // orientation flag w.r.t. the edge direction
   Pvertex    *_pvertex;       // start p-vertex
   Pedge      *_looped_prev;   // previous p-edge in the loop cycle
   Pedge      *_looped_next;   // next p-edge in the loop cycle
   Pedge      *_radial_prev;   // previous p-edge in the radial cycle
   Pedge      *_radial_next;   // next p-edge in the radial cycle
};
class Edge : public Entity {
   Entity     *_parent;        // parent entity: a p-edge or a p-face
   Pvertex    *_pvertex[2];    // two end p-vertices
   Curve      *_geometry;      // curve
};
class Pvertex : public Entity { // partial vertex (p-vertex) class
   Pvertex    *_next;          // another p-vertex associated with _vertex
   Entity     *_parent;        // parent entity: an edge or a p-edge
   Vertex     *_vertex;        // mother vertex
};
class Vertex : public Entity {
   Entity     *_parent;        // parent entity: a p-vertex or a p-face
   Point      *_geometry;      // position
};
```

Coupling Entities data structure

(Yamaguchi & Kimura, IEEE Computer Graphics and Applications, 1995)



Fan
B4 & D3 :D3 = FM(B4), B4 = FM(D3)

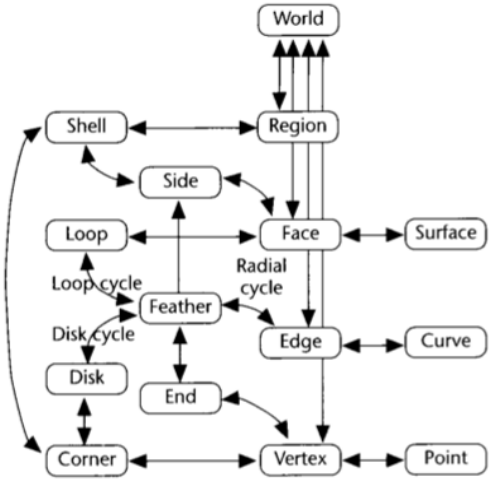
Blade
C2 & E2 :E2 = BM(C2), C2 = BM(E2)

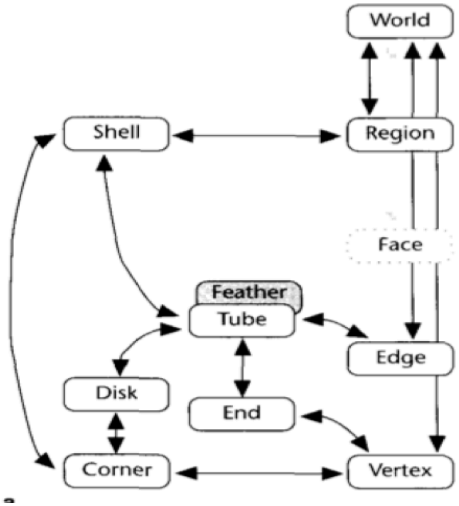
Wedge
A2 & C4 :C4 = WM(A2), A2 = WM(C4)

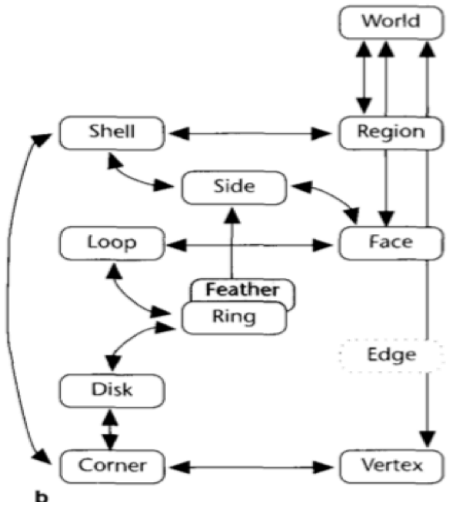
Loop cycle
A1>>A2>>A3>>A4 :A2 = CCL(A1), A3 = CCL(A2)
A4>>A3>>A2>>A1 :A1 = CL(A2), A2 = CL(A3)

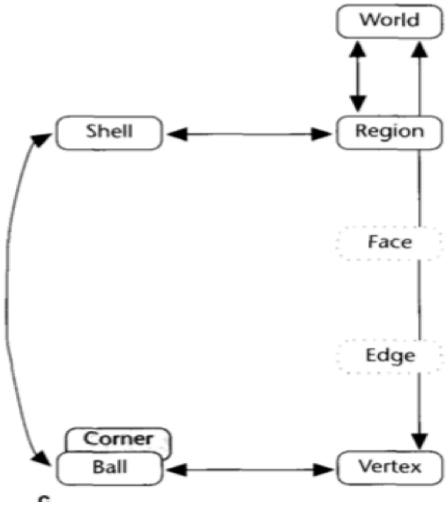
Radial cycle
D2>>C3>>F4 :C3 = CCR(D2), F4 = CCR(C3)
F4>>C3>>D2 :D2 = CR(C3), C3 = CR(F4)

Disk cycle
A3>>C4>>B2 :C4 = CCD(A3), B2 = CCD(C4)
B2>>C4>>A3 :A3 = CD(C4), C4 = CD(B2)



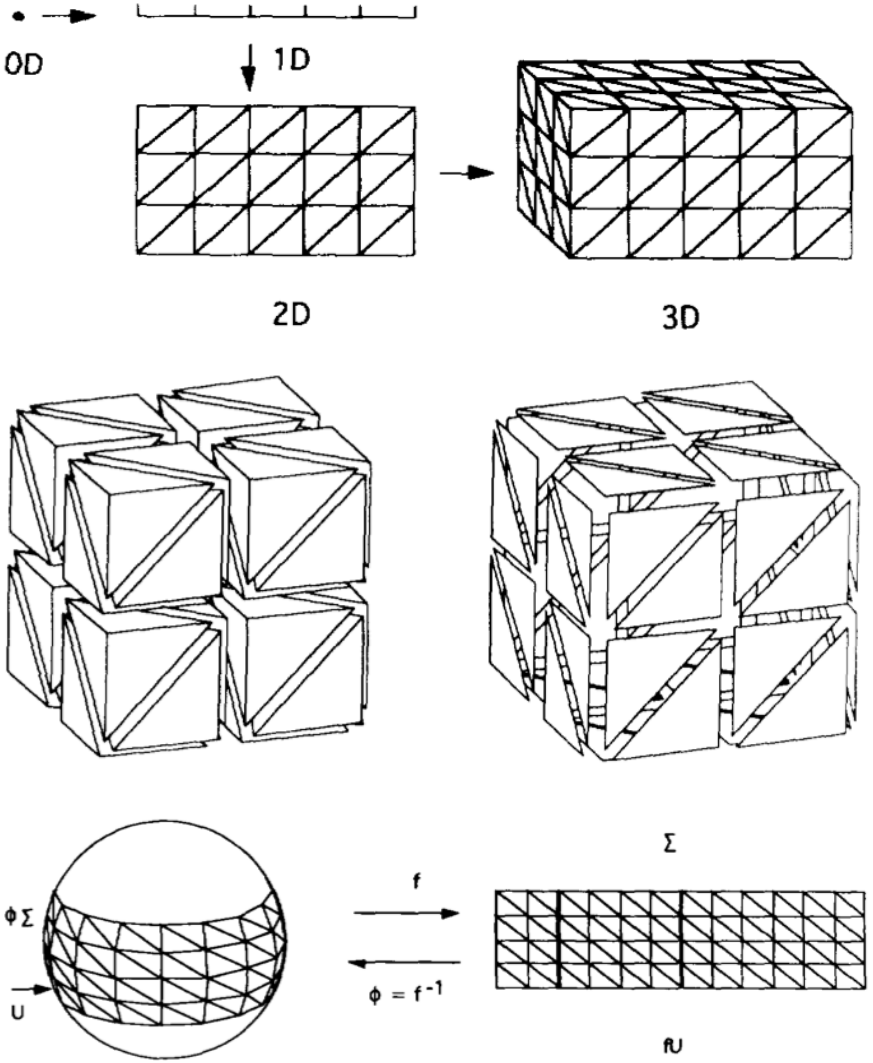






Winged data structure

(Paoluzzi, Bernardini, Cattani & Ferrucci, ACM Transactions on Graphics, 1993)



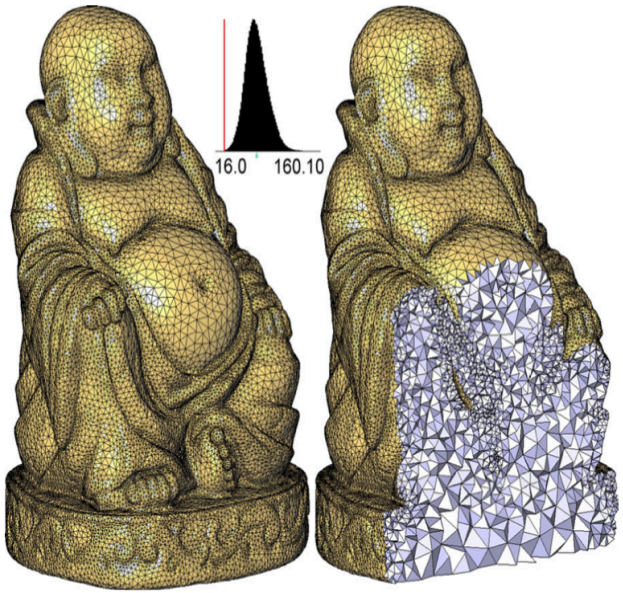
Dimension-Independent Modeling with Simplicial Complexes

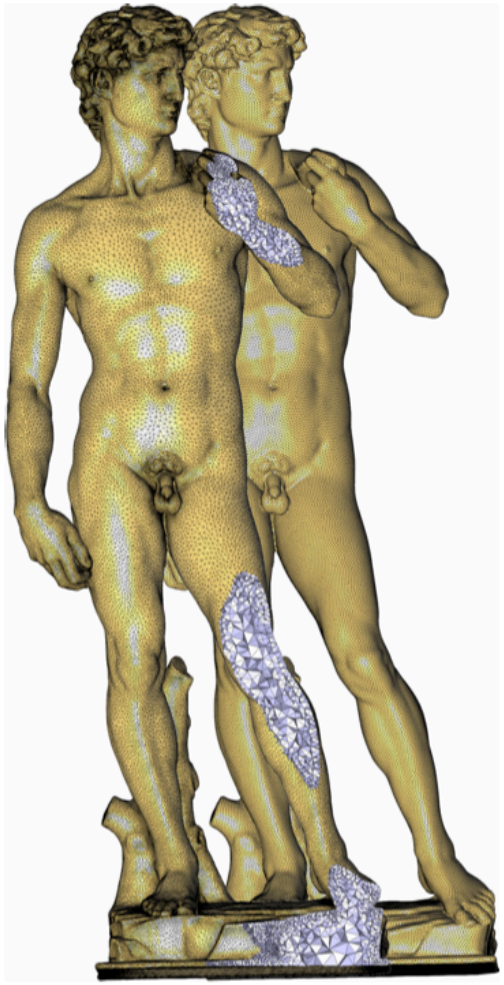
used for simplicial grid data structures & isogeometric simulations

Isotropic Tetrahedron Mesh

(Tournois, Wormser, Alliez & Desbrun, ACM Transactions on Graphics, 2009)

Interleaving Delaunay Refinement and Optimization for Practical Isotropic Tetrahedron Mesh Generation

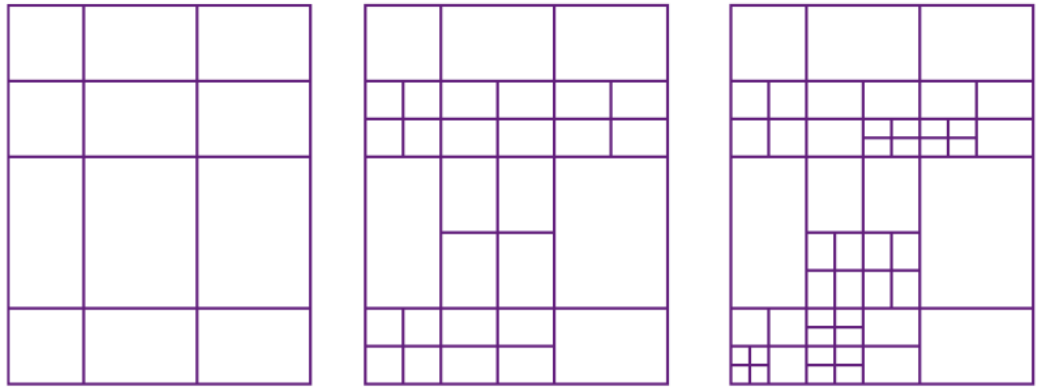




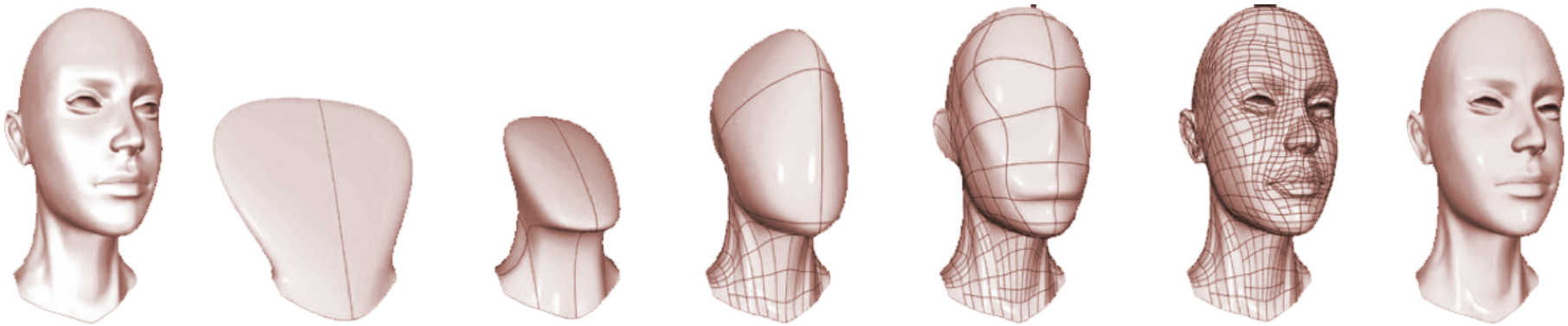
PHT-splines (2D local refinement)

(Jiansong Deng, Falai Chen, Xin Li, Changqi Hu, Weihua Tong, Zhouwang Yang, Yuyu Feng, Graphical Models, 2008)

Polynomial Splines over Hierarchical T-meshes

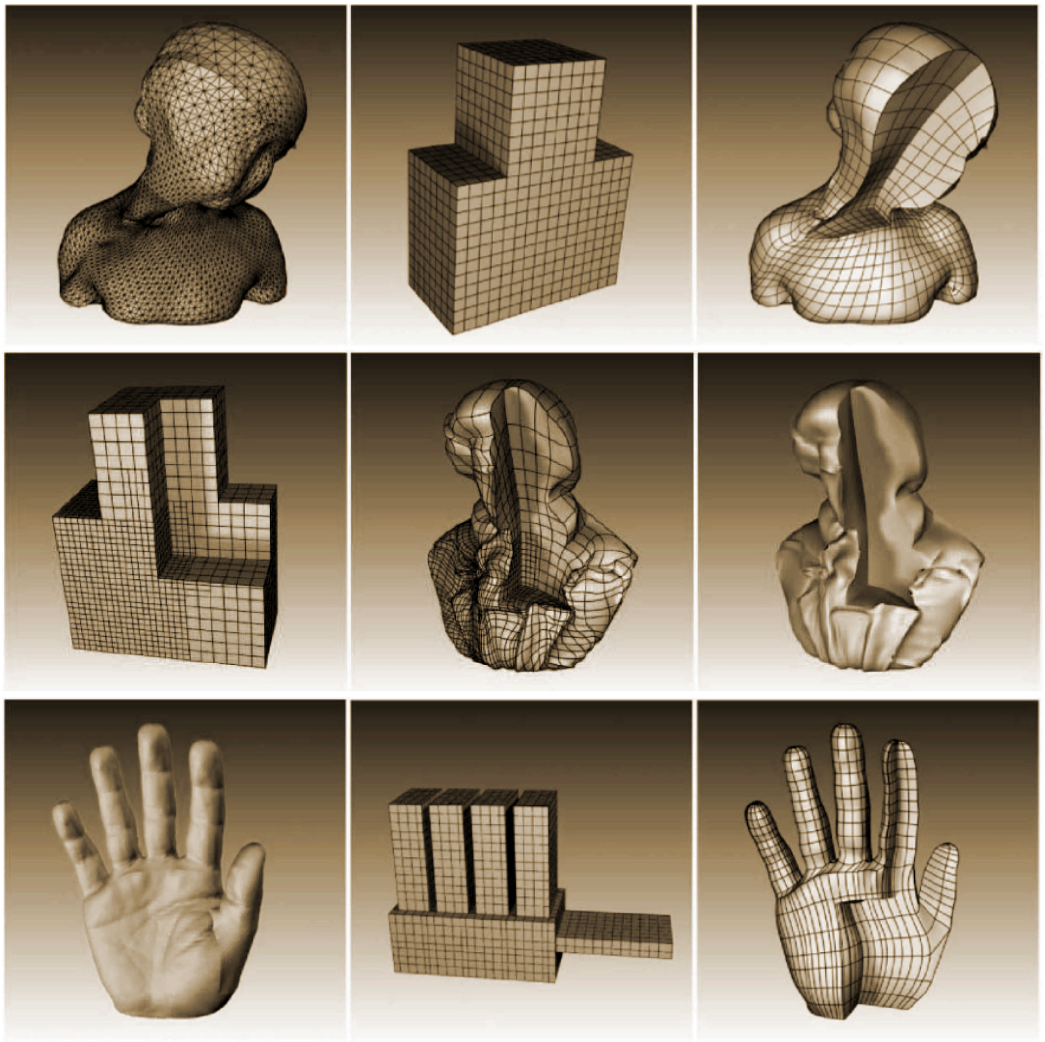


hierarchical T-mesh



Restricted Trivariate Polycube (RTP) Splines

(Kexiang Wang, Xin Li, Bo Li, Huanhuan Xu, Hong Qin, IEEE Trans. Vis. Comp. Graphics, 2012)



in the framework of isogeometric analysis, trivariate tensor-prod B-splines/NURBS are directly used to model smooth geometry and material attributes of solid objects for physical simulation

Look at big geometric data

Challenge in CAD/CAE/CAM (PLM): mature technology, hard to parallelise

Need: **Rethinking the foundations of geometric and topological computing**

Computational problems in science and technology must deal with increasingly complex geometric information and applications.

Complexity of geometric information stems from dramatic increase in size, diversity, and complexity of geometric data:

- point clouds,
- boundary meshes,
- NURBs representations,
- finite element meshes,
- 3D imagery,
- and so on

Rethinking some foundations

Dealing with Big Data and scalable architectures

Googles map-reduce

Emerging applications (e.g. space, nano & bio technology, medical 3D) require the convergence of shape synthesis and analysis from:

- computer imaging
- computer graphics
- computer-aided geometric design
- discrete meshing of domains
- physical simulations

The goals of unification, scalability, and massively parallel distributed computing

call for **rethinking the foundations** of geometric and topological computing

GOAL: 10^3 **times faster and** 10^4 **times bigger**

LAR = algebraic topology + linear algebra

A new standard for model topology representation

Define a standard for model topology using a very general and simple repr scheme:

Models: (co)chain complexes $\rightarrow$ Reprs: sparse binary matrices

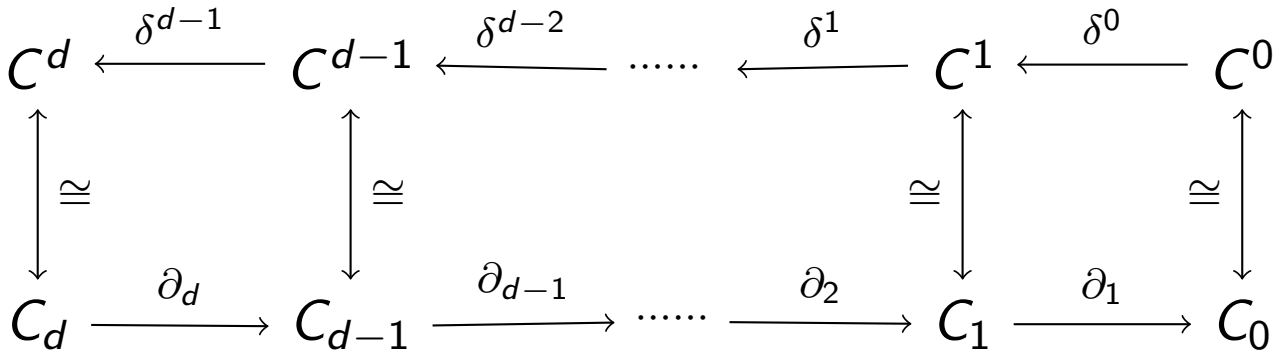
LAR based on chains, the domains of discrete integration, and cochains, the discrete prototype of differential forms, so naturally integrating the geometric shape with the physical properties

- *SIAM-ACM Geometric and Physical Modeling* (GD/SPM), Denver, Nov 11–14, 2013
- *Computer-Aided Design*, Volume 46, January 2014, Pages 269–274

Chain and cochain complex

$C_0, C_1, C_2, C_3 \equiv V, E, F, P$ $\quad \nabla_p \equiv$ *Laplacian* $\quad \delta_0, \delta_1, \delta_2 \equiv$grad,curl,div

cochains (all maps, discrete fields) and coboundary maps (δ^d operators)



chains (linear spaces of model subsets) and boundary maps (∂_d operators)

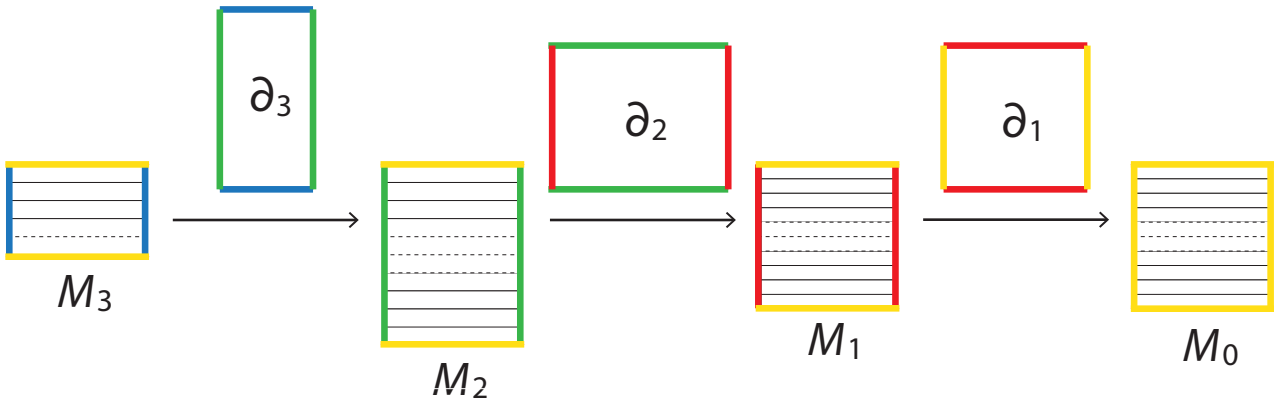
$$\boxed{\delta^p = \partial_{p+1}^\top} \qquad \boxed{\Delta_p = (\delta^p)^\top \delta^p + (\delta^{p-1})^\top \delta^{p-1}}$$

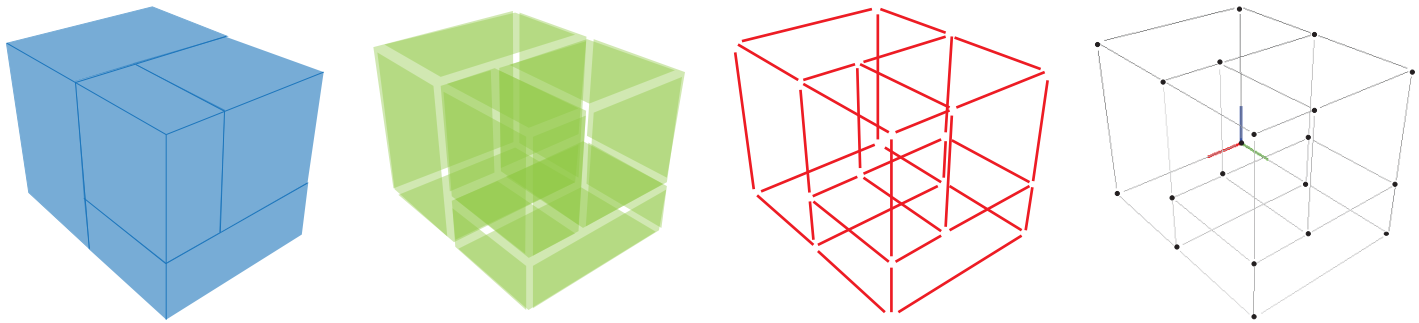
$$C^3 \xleftarrow{\text{div}} C^2 \xleftarrow{\textbf{curl}} C^1 \xleftarrow{\textbf{grad}} C^0$$

Chain complex (of chain spaces)

Sequence of linear spaces (over Z_2 or Z_3) of d-cell subsets

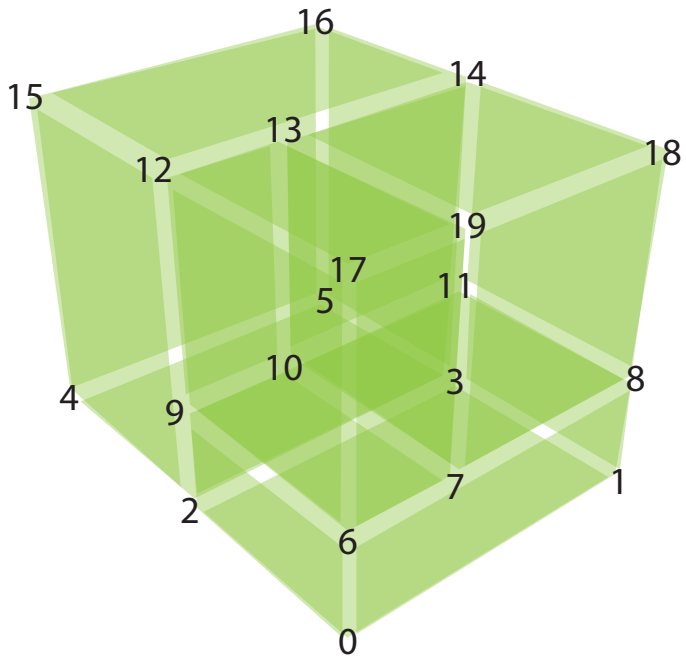
Unit *d*-chains (single *d*-cell subsets), are the standard bases (M_d rows) of *d*-chain spaces





Characteristic matrix of d-chain spaces

Matrix representation of the standard basis (the *d*-cells as subsets of vertices)



$$M_3 = \begin{pmatrix} 0&0&1&1&1&1&0&0&0&1&1&1&1&1&1&1&1&0&0&0 \\ 1&1&1&1&0&0&1&1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&1&1&0&1&1&0&0&0&1&0&1 \\ 0&0&0&0&0&0&0&1&1&0&1&1&0&1&1&0&0&0&1&1 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&1&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&1&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&1&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&1&0&0&1&0&1&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0&0&0&0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&1&0&1&0 \\ 0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&1&0 \end{pmatrix}$$

Compressed Sparse Row (CSR) matrix storage

We already know it ... :-)

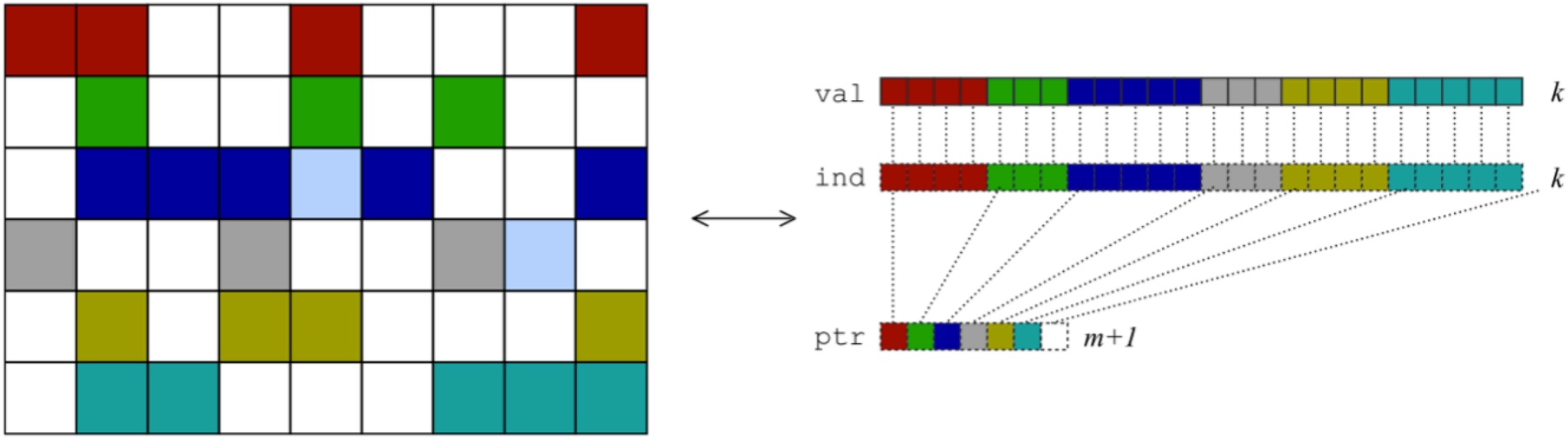
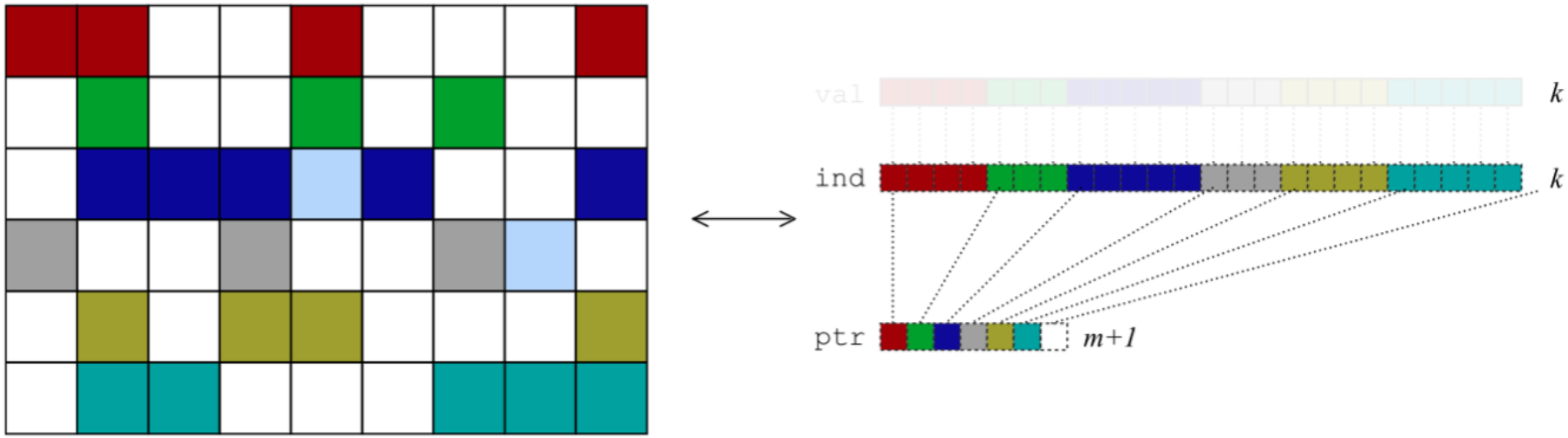


image from

Samuel Williams, Leonid Oliker, Richard Vuduc, John Shalf, Katherine Yelick, and James Demmel, Optimization of sparse matrix-vector multiplication on emerging multicore platforms, Proceedings of the 2007 ACM/IEEE conference on Supercomputing (New York, NY, USA), SC '07, ACM, 2007, pp. 38:1–38:12.

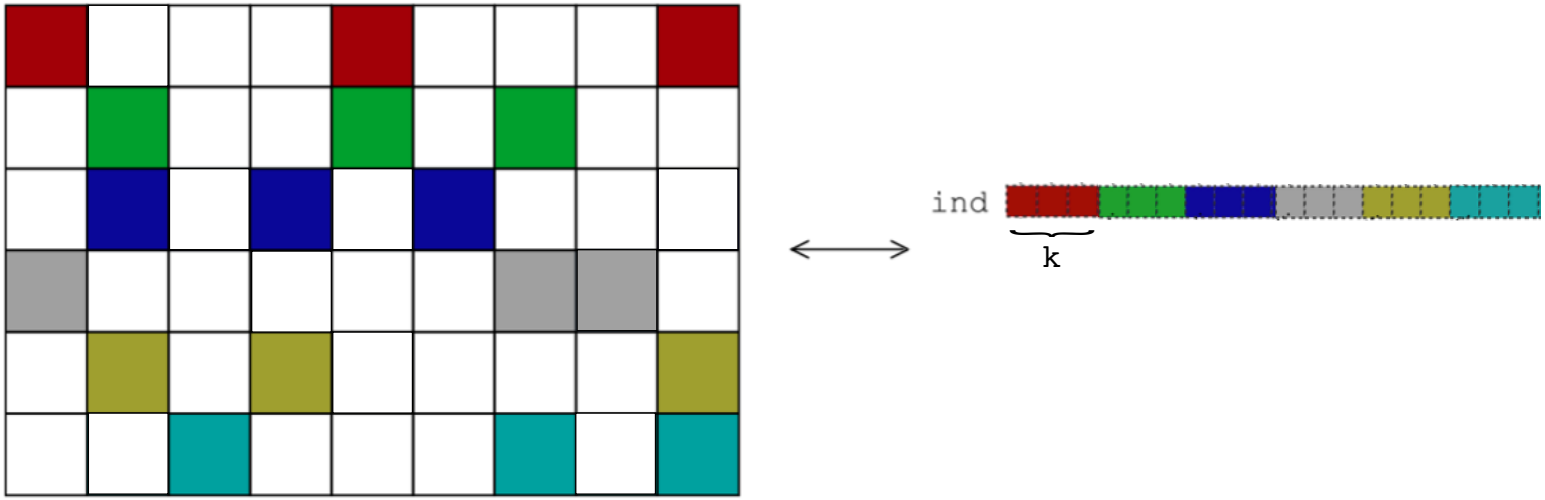
CSR storage of a binary matrix



of course, non-zero values (all ones) do not require storage

CSR storage of a binary matrix M_d such that:

$$M_d \, \mathbf{1} = \mathbf{k}$$

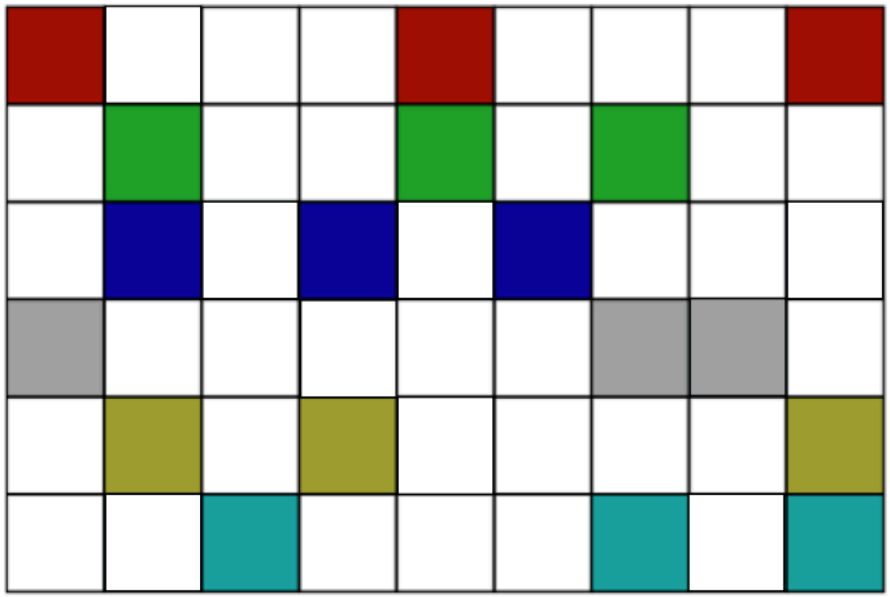


important cases:

- regular simplicial d-complexes: $k = d + 1$
- regular cuboidal d-complexes: $k = 2^d$

Storage of LAR(X) $\equiv$ CSR(M_d) matrix

for triangulated B-reps:
(very common case)



Remark (Storage occupancy)

$$|FV| = 2|E| \quad !!!$$

Remark (All topology representation)

$$|VE| + |VF| = 4|E|$$

Remark (Any topological queries)

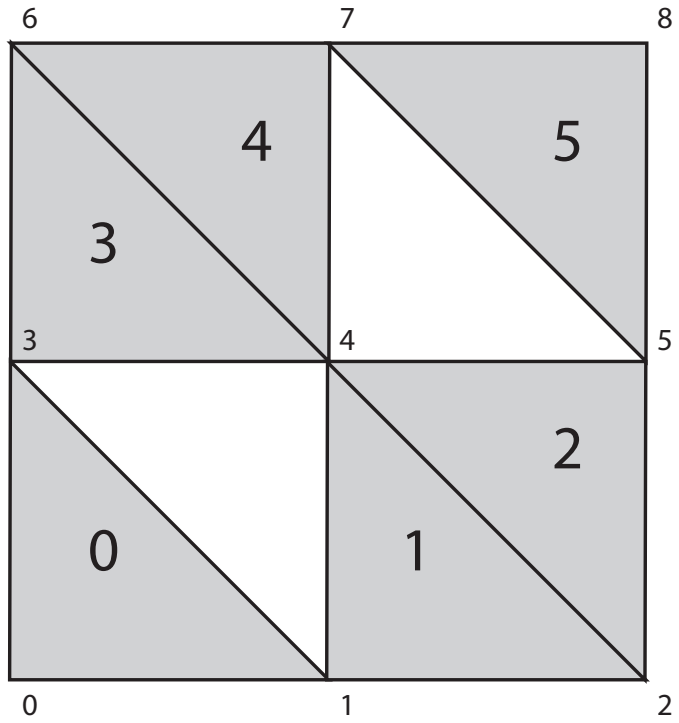
single SpMV multiplication

Example: input of simplicial complex

Reduced and compressed LAR

$\mathrm{FV} := \mathrm{CSR}(M_2)$

```
FV = [[0,1,3],
      [1,2,4],
      [2,4,5],
      [3,4,6],
      [4,6,7],
      [5,7,8]]
```



Remark

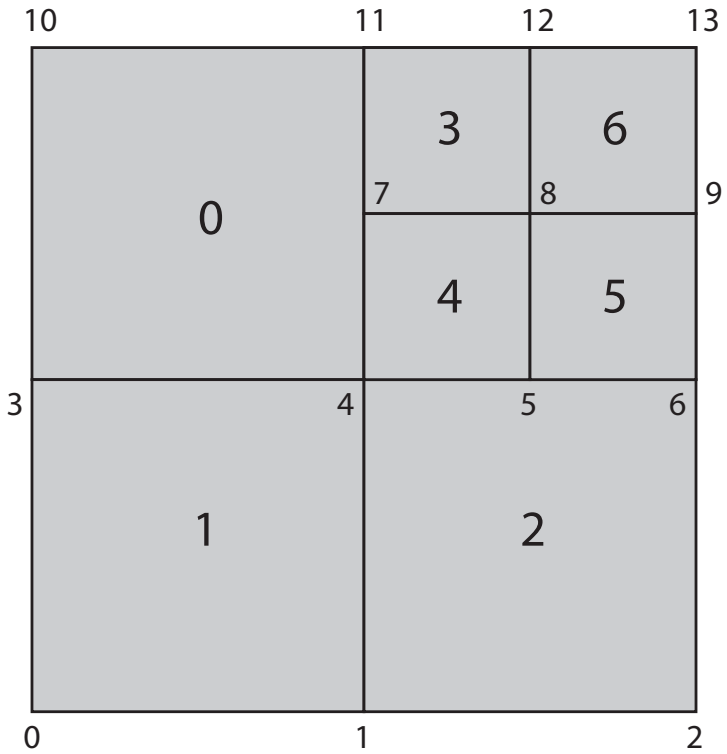
The bulk of common graphics formats !! *(e.g.: .OBJ or .PLY)*

Example: input of Tmesh

Reduced (non compressed) LAR: of course, it is the same in 3D

$FV := \mathrm{CSR}(M_2)$

```
FV = [[3,4,7,10,11],
      [0,1,3,4],
      [1,2,4,5,6],
      [7,8,11,12],
      [4,5,7,8],
      [5,6,8,9],
      [8,9,12,13]]
```



Remark

The bulk of common graphics formats !! (e.g.: `IndexedFaceSets`)

Algorithm: computation of $\partial_p = \delta_{p1}^t$

Knowledege of ∂_p gives the boundary of every p-chain by a single *SpMV* product

1. compute the numbers of vertices of $(d-1)$-cells $\mu_{p-1}^i \in \Lambda_{p-1}$:

$$k_i := \sharp \mu_{p-1}^i \qquad (\mathbf{k} = M_{p-1}\mathbf{1})$$

2. Compute

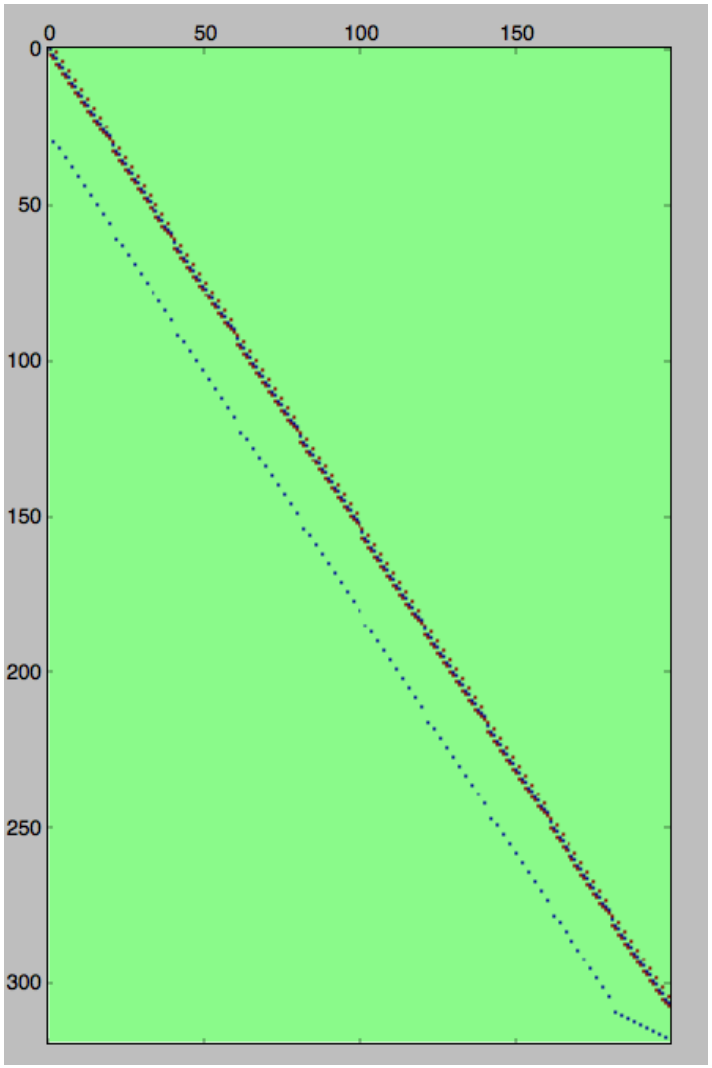
$$M_{p-1}^p := M_{p-1}\, M_p^t$$

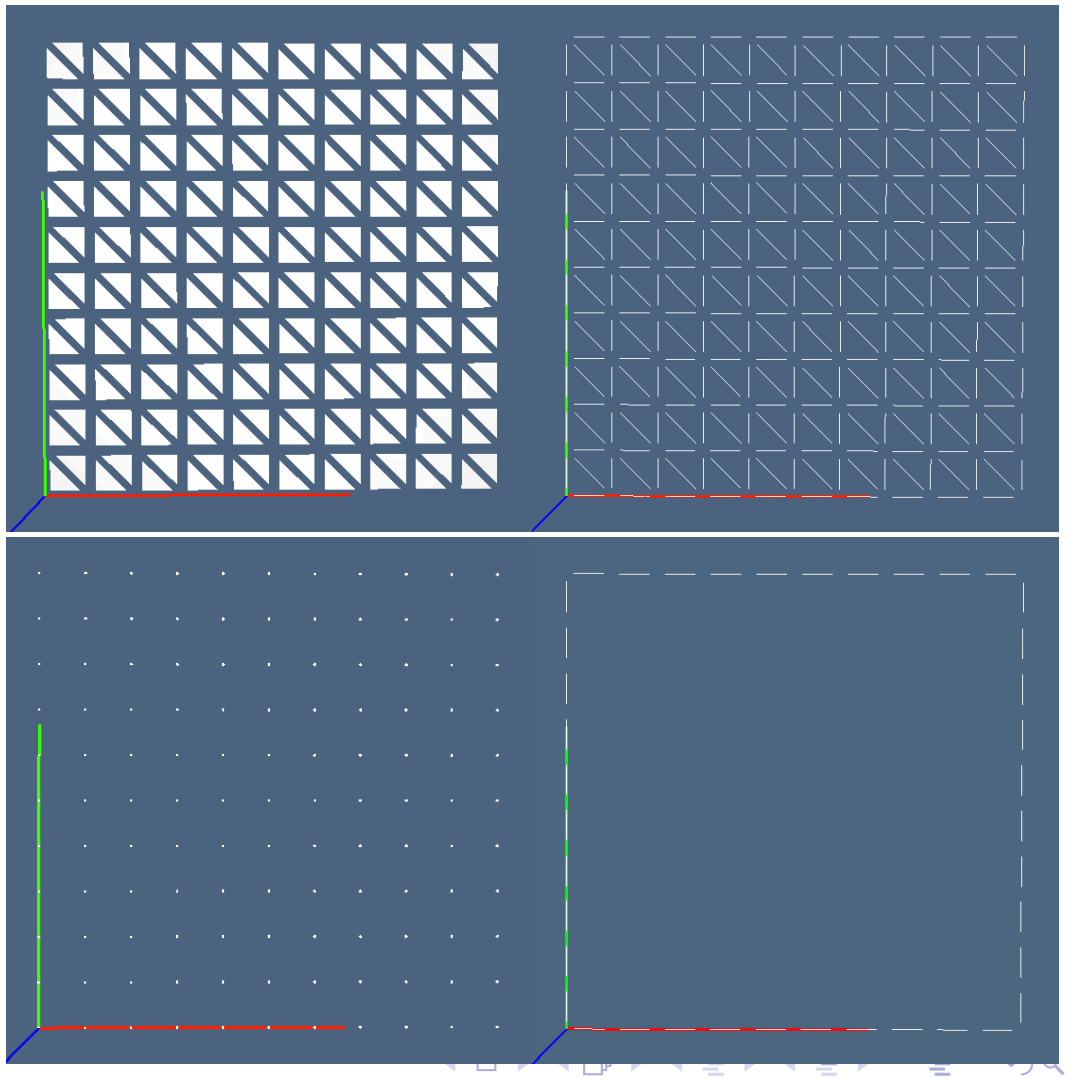
3. For each $1 \leq i \leq |\Lambda_{p-1}|$, and for each $1 \leq j \leq |\Lambda_p|$:

$$\text{if} \quad M_{p-1}^p(i,j) = k_i \quad \begin{cases} \text{then} & [\partial_p](i,j) := 1 \\ \text{else} & [\partial_p](i,j) := 0 \end{cases}$$

Example: 2D simplicial grid

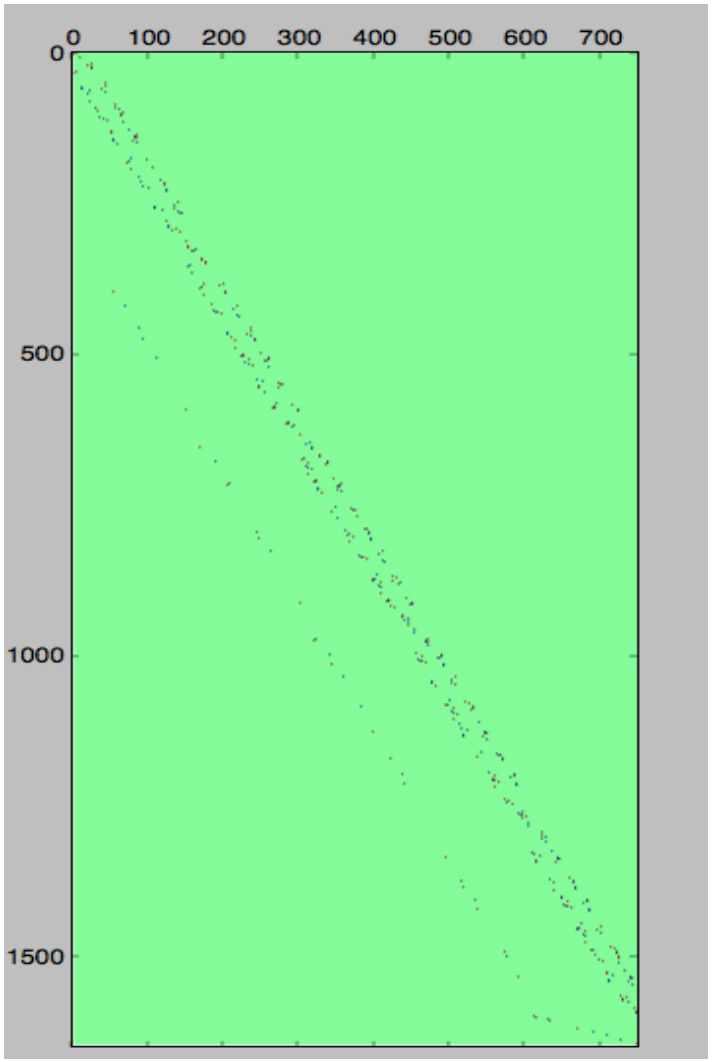
Boundary computation via a single *SpMV* product

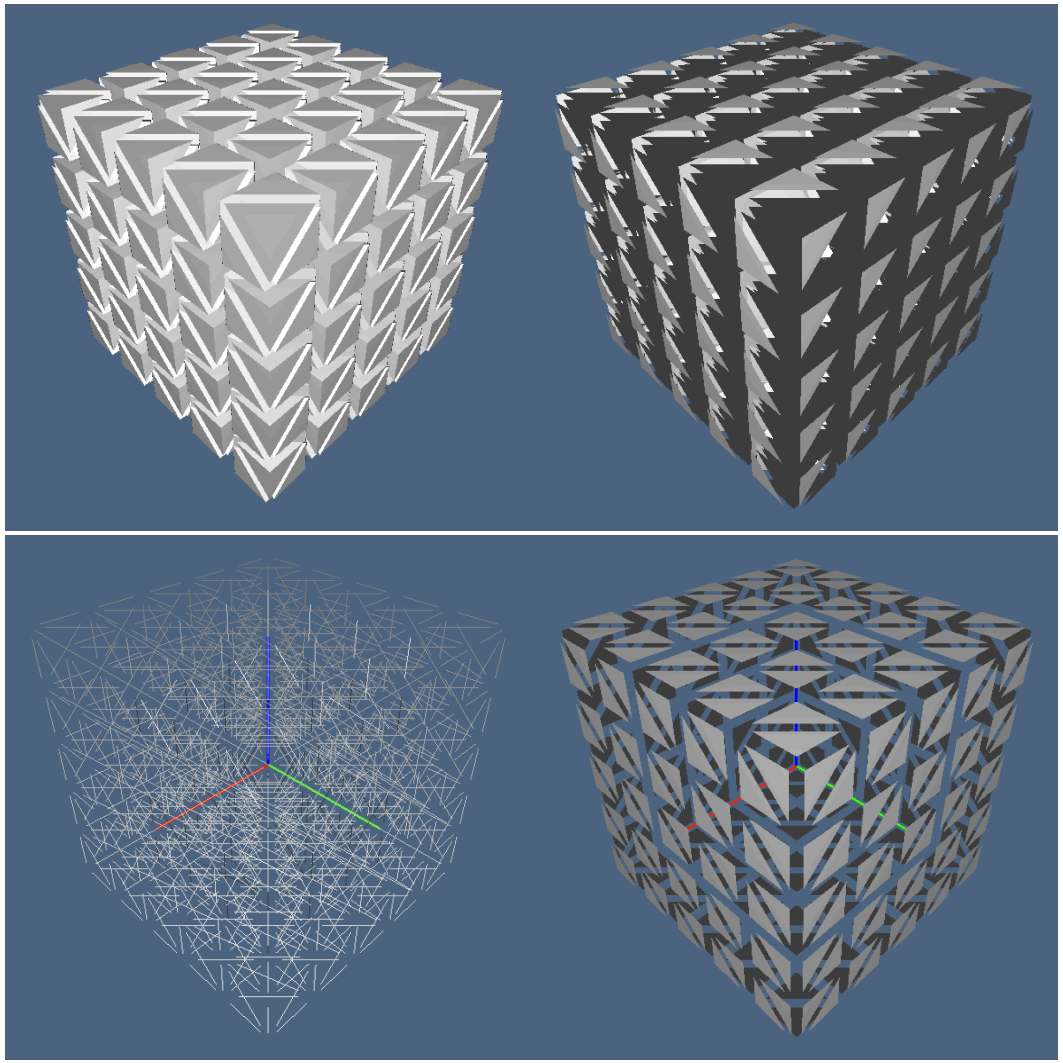




Example: 3D simplicial grid

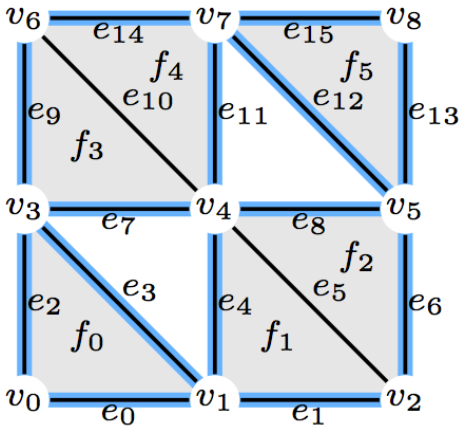
Boundary computation via a single *SpMV* product





Incidence and adjacency relations

Computation of any topological query via a single *SpMV* product



for convex-cell complexes (ex: simplicial or cuboidal) only the CSR(M_d) is needed:

	$M_1^t \Leftarrow$	$M_q^t \Leftarrow$	M_d^t
M_1		$\Downarrow$	
$\Uparrow$			
M_p	$\Longrightarrow$	$M_pM_q^t$	
$\Uparrow$			
M_d			

$M_1^tM_1$ = VV =
```
[[0,1,3],
 [0,1,2,3,4],
 [1,2,4,5],
 [0,1,3,4,6],
 [1,2,3,4,5,6,7],
 [2,4,5,7,8],
 [3,4,6,7],
 [4,5,6,7,8],
 [5,7,8]]
```

M_1^t = VE =
```
[[0,2],
 [0,1,3,4],
 [1,5,6],
 [2,3,7,9],
 [4,5,7,8,10,11],
 [6,8,12,13],
 [9,10,14],
 [11,12,14,15],
 [13,15]]
```

M_2^t = VF =
```
[[0],
 [0,1],
 [1,2],
 [0,3],
 [1,2,3,4],
 [2,5],
 [3,4],
 [4,5],
 [5]]
```

M_1 = EV =
```
[[0,1],
 [1,2],
 [0,3],
 [1,3],
 [1,4],
 [2,4],
 [2,5],
 [3,4],
 [4,5],
 [3,6],
 [4,6],
 [4,7],
 [5,7],
 [5,8],
 [6,7],
 [7,8]]
```

$M_1M_1^t$ = EE =
```
[[0,1,2,3,4],
 [0,1,3,4,5,6],
 [0,2,3,7,9],
 [0,1,2,3,4,7,9],
 [0,1,3,4,5,7,8,10,11],
 [1,4,5,6,7,8,10,11],
 [1,5,6,8,12,13],
 [2,3,4,5,7,8,9,10,11],
 [4,5,6,7,8,10,11,12,13],
 [2,3,7,9,10,14],
 [4,5,7,8,9,10,11,14],
 [4,5,7,8,10,11,12,14,15],
 [6,8,11,12,13,14,15],
 [6,8,12,13,15],
 [9,10,11,12,14,15],
 [11,12,13,14,15]]
```

$M_1M_2^t$ = EF =
```
[[0,1],
 [0,1,2],
 [0,3],
 [0,1,3],
 [0,1,2,3,4],
 [1,2,3,4],
 [1,2,5],
 [0,1,2,3,4],
 [1,2,3,4,5],
 [0,3,4],
 [1,2,3,4],
 [1,2,3,4,5],
 [2,4,5],
 [2,5],
 [3,4,5],
 [4,5]]
```

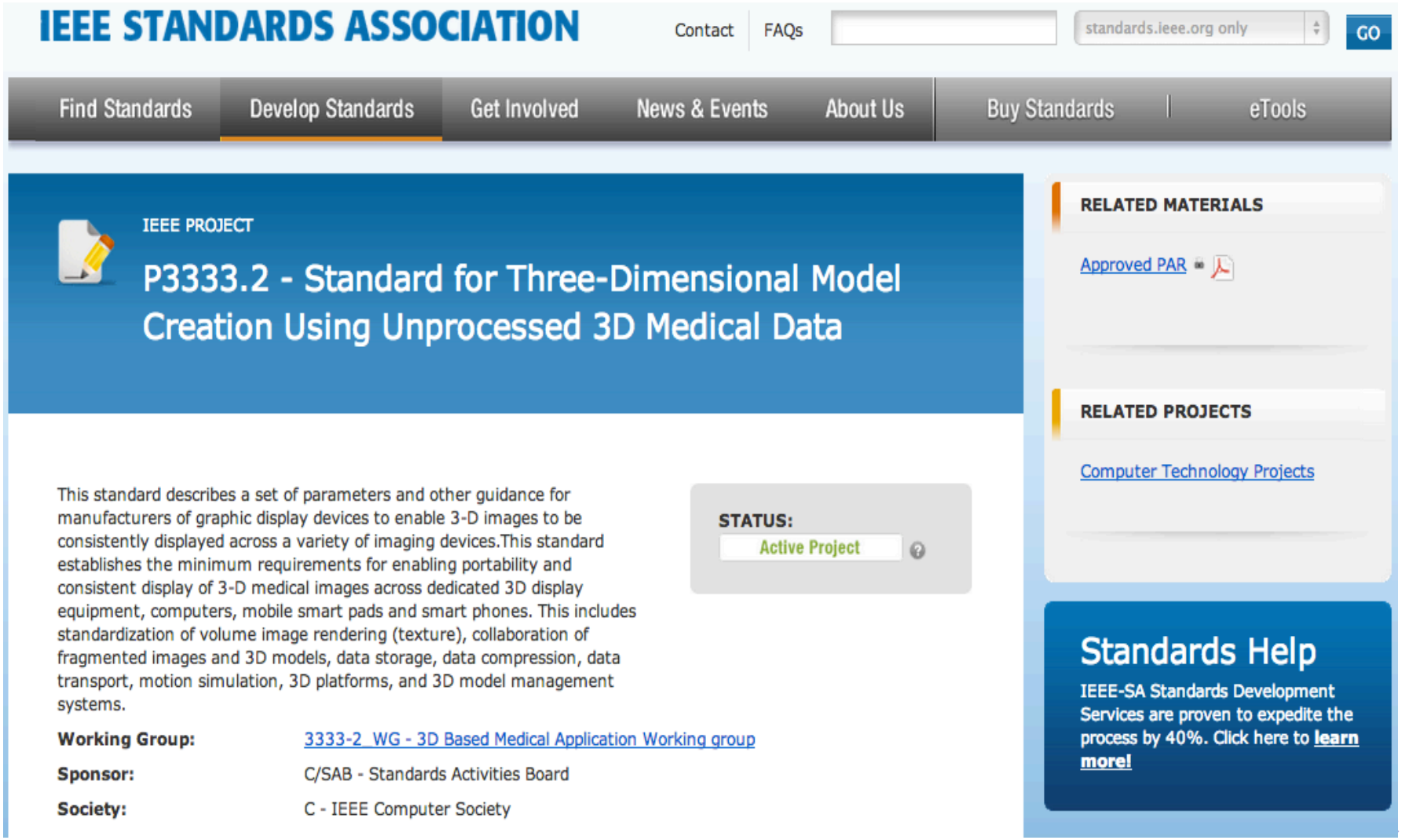
M_2 = FV =
```
[[0,1,3],
 [1,2,4],
 [2,4,5],
 [3,4,6],
 [4,6,7],
 [5,7,8]]
```

$M_2M_1^t$ = FE =
```
[[0,1,2,3,4,7,9],
 [0,1,3,4,5,6,7,8,10,11],
 [1,4,5,6,7,8,10,11,12,13],
 [2,3,4,5,7,8,9,10,11,14],
 [4,5,7,8,9,10,11,12,14,15],
 [6,8,11,12,13,14,15]]
```

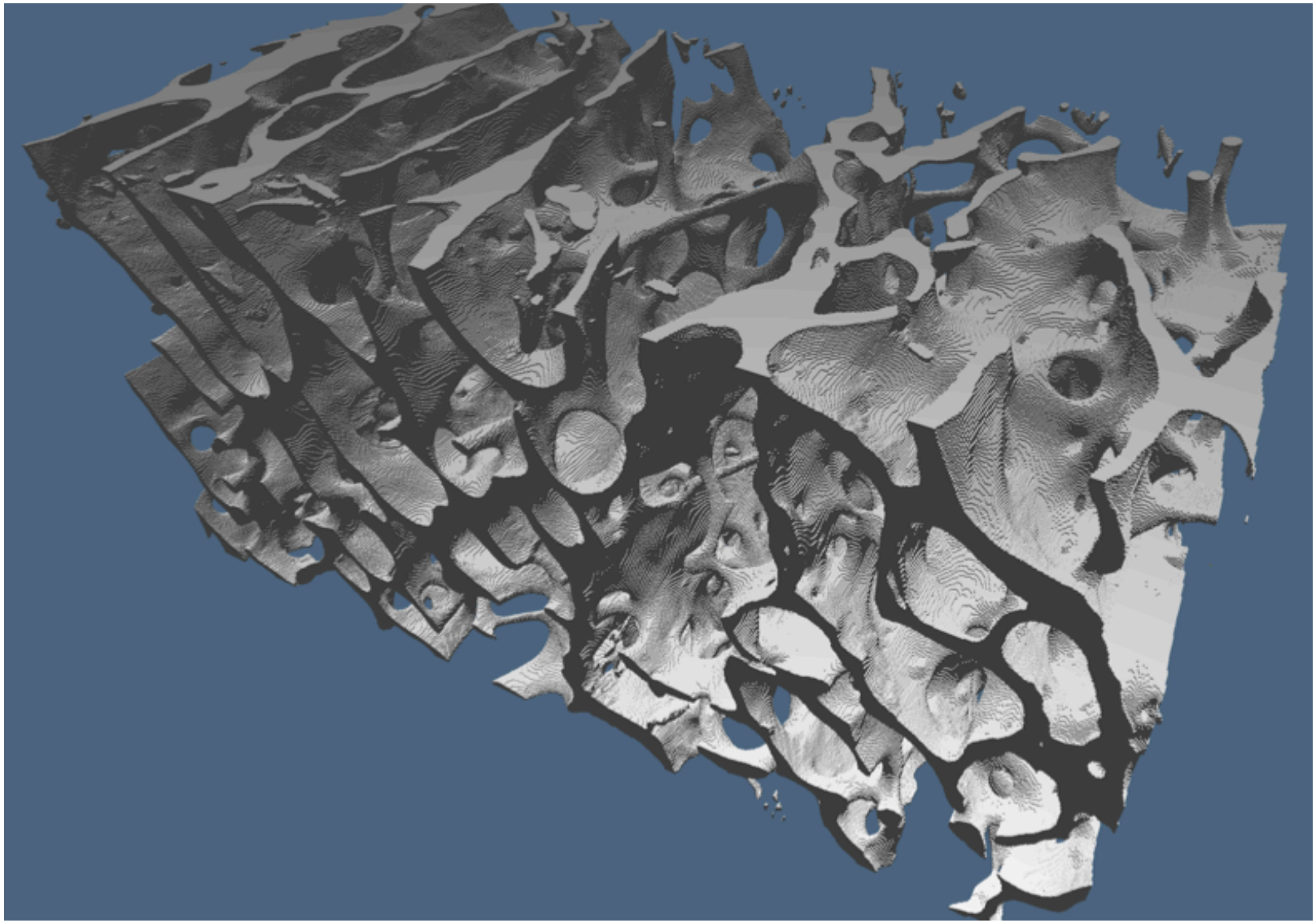
$M_2M_2^t$ = FF =
```
[[0,1,3],
 [0,1,2,3,4],
 [1,2,3,4,5],
 [0,1,2,3,4],
 [0,1,2,3,4,5],
 [2,4,5]]
```

Models from medical images

LAR was developed in the framework of a 3D medical standard

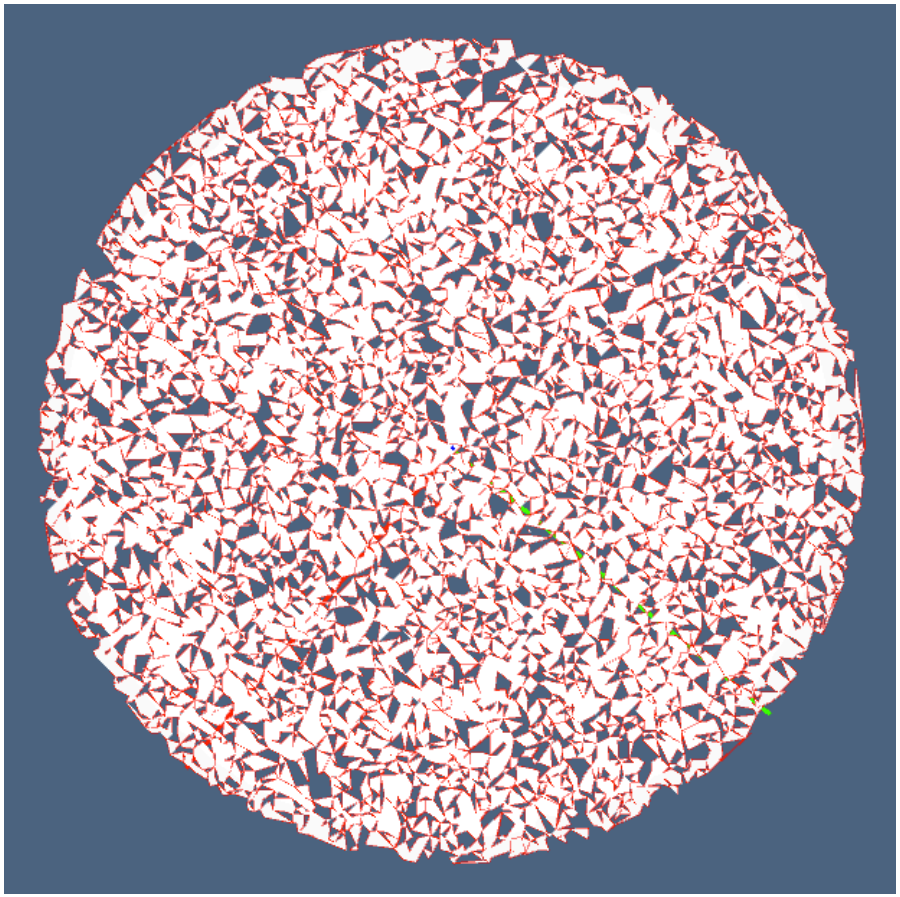


Solid model from 3D image (spongy bone)

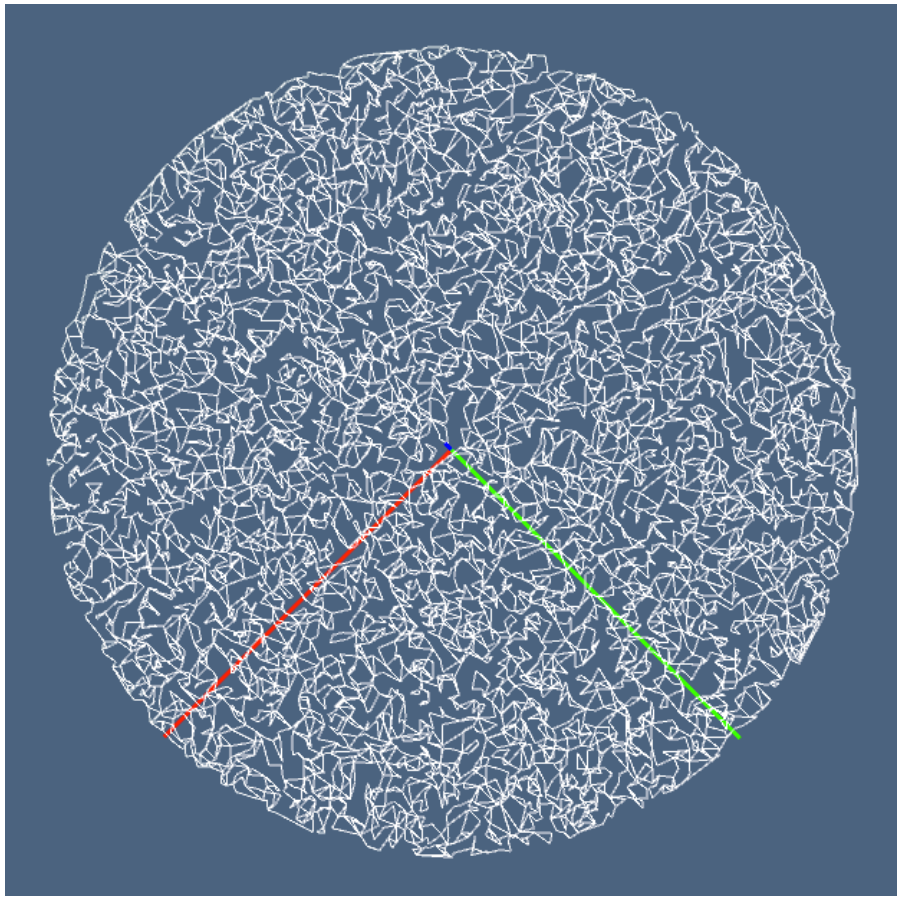


A random polygon and its boundary

Euler characteristic $\chi = k_0 - k_1 + k_2$ is pretty $\neq 1$



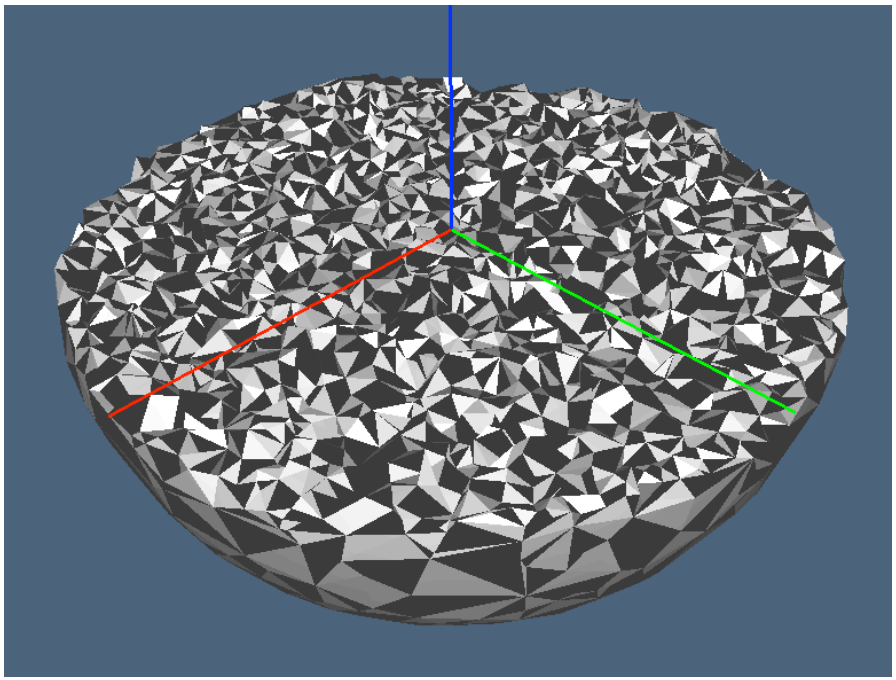
chain c_2



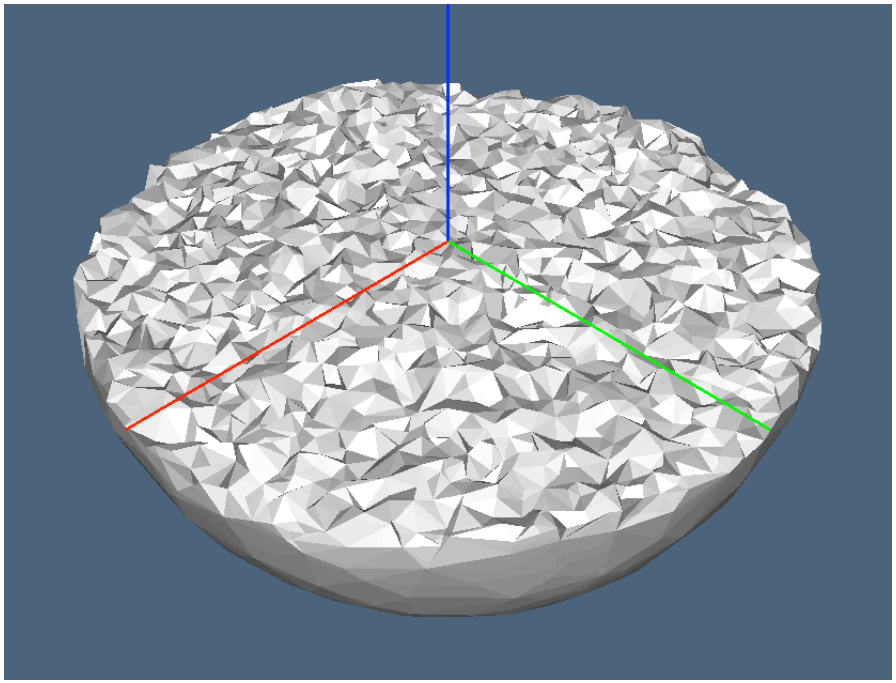
chain $c_1 = \partial_2(c_2)$

A random polyhedron and its oriented boundary

To recover the orientation of 2-cells (unit chains) from that of their coboundary is easy



chain c_3



chain $c_2 = \partial_3(c_3)$

Space Arrangement

Definitions

- Arrangement is the decomposition of the d-dimensional affine or projective space,
- into connected and (relatively open) cells of lower dimensions
- induced by an intersection of a finite collection of geometric objects

Formally:

- arrangement $\mathcal{A}(\mathcal{S}) = X$,
- where $X := \bigcup_{k=0}^{d} X_k$,
- X_k is called the k-skeleton of the cellular complex X, usually with $d \in \{2, 3\}$,
- providing a cellular decomposition of the space $\mathbb{E}^d$ where the $\{$underlying space$\}$ (point-set) of $\mathcal{S}$ is embedded.

Computational goal

Given the input collection $\mathcal{S}$, how to compute and represent the chain complex

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

where C_p $(0 \leq p \leq d)$, with $d \in \{2, 3\}$, is a linear space of p-chains (sets of p-cells with algebraic structure), and where $\delta_{p-1} = \partial_p^\top$.

Arrangement of cellular complexes: a quick tour

Paoluzzi, Shapiro, DiCarlo, 2018 (submitted)

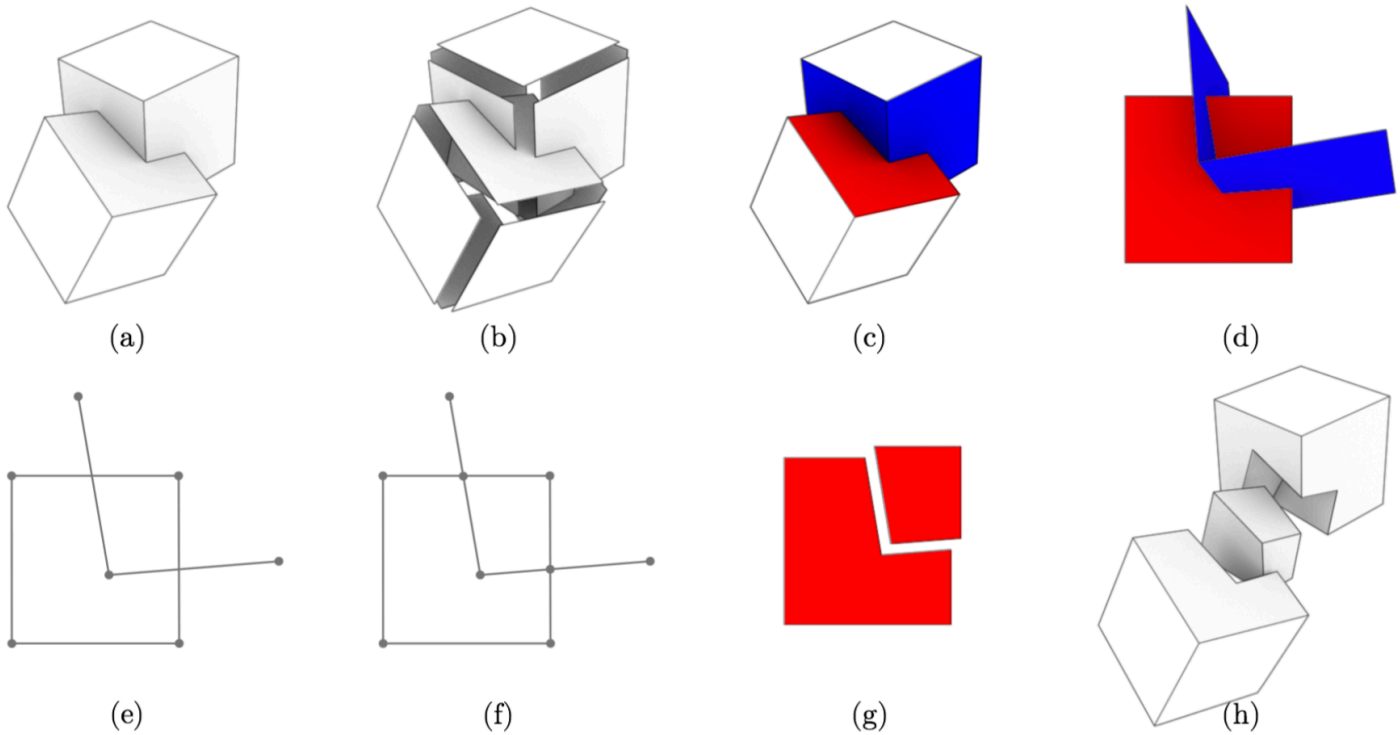


Fig. 1. Cartoon display of the Merge algorithm: (a) the two input solids; (b) the exploded input collection B of 2-cells embedded in $\mathbb{E}^3$; (c) 2-cell σ (red) and the set $\Sigma(\sigma)$ (blue) of possible intersection; (d) affine map of $\sigma \cup \Sigma$ on $z = 0$ plane; (e) reduction to a set of 1D segments in $\mathbb{E}^2$; (f) pairwise intersections; (g) regularized plane arrangement $\mathcal{A}(\sigma \cup \Sigma)$; (h) exploded 3-complex extracted from the 2-complex $X_2(\cup_{\sigma \in B} \mathcal{A}(\sigma \cup \Sigma))$ in $\mathbb{E}^3$. The LAR representation of both the input data (not a complex) and the output data (3-complex with three 3-cells) is given in Appendix A

Topological gift-wrapping

Arrangement of cellular complexes: a quick tour

Paoluzzi, Shapiro, DiCarlo, 2018 (submitted)

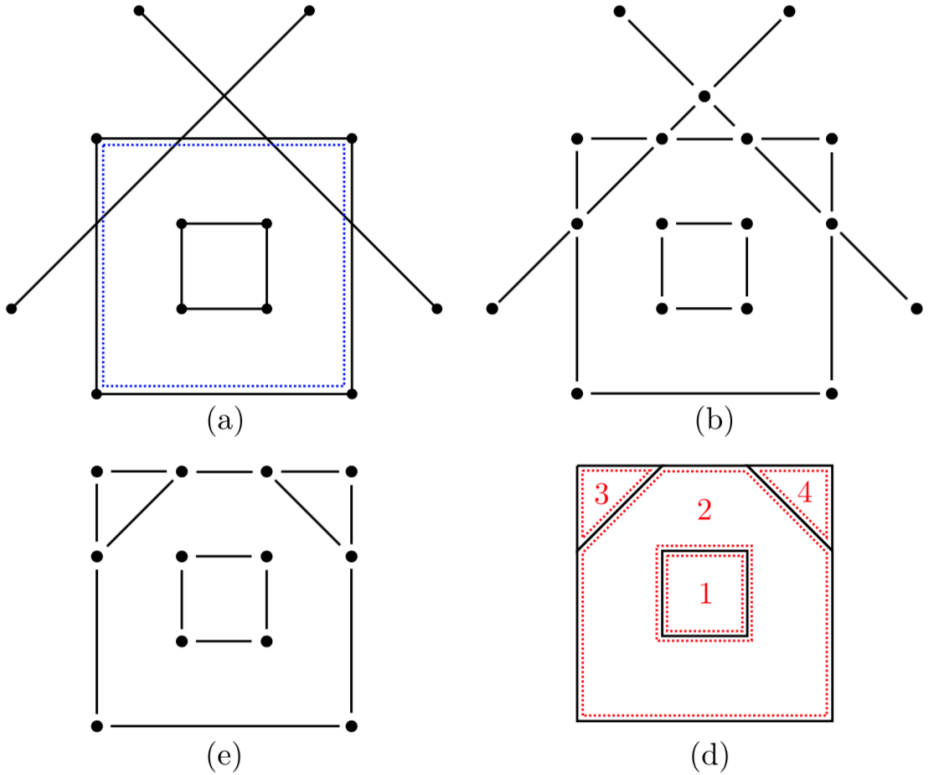


Fig. 2. Basic case: computation of the regularized arrangement of a set of lines in $\mathbb{E}^2$: (a) the input, i.e. the 2-cell σ (blue) and the line segment intersections of $\Sigma(\sigma)$ with $z = 0$; (b) all pairwise intersections; (c) removal of the 1-subcomplex external to σ. To this purpose an efficient and robust point classification algorithm [?] is used, ; (d) interior cells of the regularized 2-complex $X_2 = \mathcal{A}(\sigma \cup \Sigma)$ generated as arrangement of $\mathbb{E}^2$ produced by $\sigma \cup \Sigma$.

Arrangement of cellular complexes: a quick tour

Paoluzzi, Shapiro, DiCarlo, 2018 (submitted)

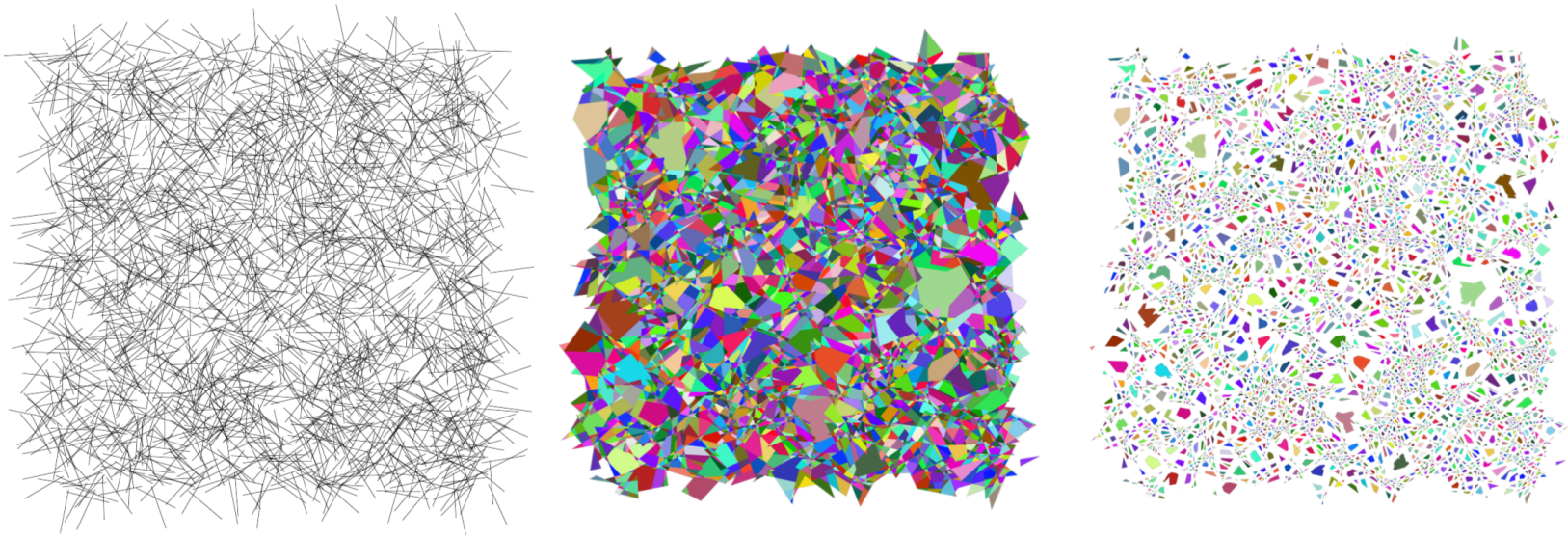


Fig. 5. The regularized 2D arrangement X_2 of the plane generated by a set of random line segments. Note that cells $\sigma \in X_2$ are not necessarily convex. The Euler characteristic is $\chi = \chi_0 - \chi_1 + \chi_2 = 11361 - 20813 + 9454 = 2$.

Arrangement of cellular complexes: a quick tour

Paoluzzi, Shapiro, DiCarlo, 2018 (submitted)

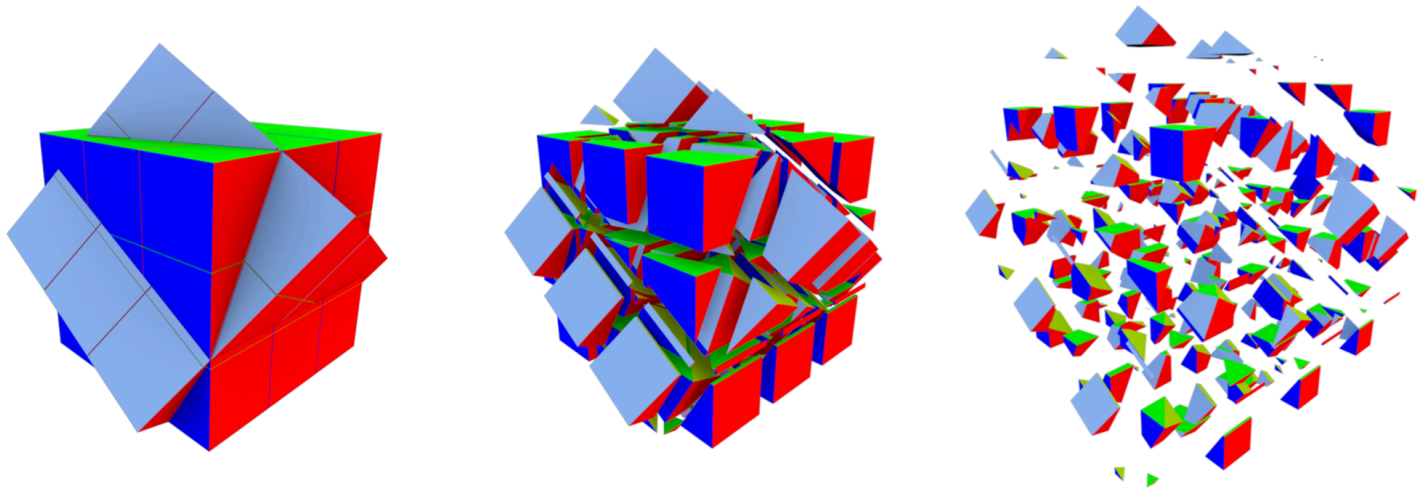


Fig. 6. Merge of two rotated $3 \times 3 \times 3$ cuboidal 3-complexes: (a) the initial positions of the two input 3-complexes, each with 3^3 unit cubic cells; (b) 3-cells of merged complex, with space explosion of small scaling parameter; (c) larger space explosion. Each exploded cell is translated by a scaling-induced movement of its centroid. Output cells are not necessarily convex. In the merged complex we get 236 three-cells and 816 two-cells, both possibly non-convex. Ratio new/old 3-cells is 4.3704.

Arrangement of cellular complexes: a quick tour

Paoluzzi, Shapiro, DiCarlo, 2018 (submitted)

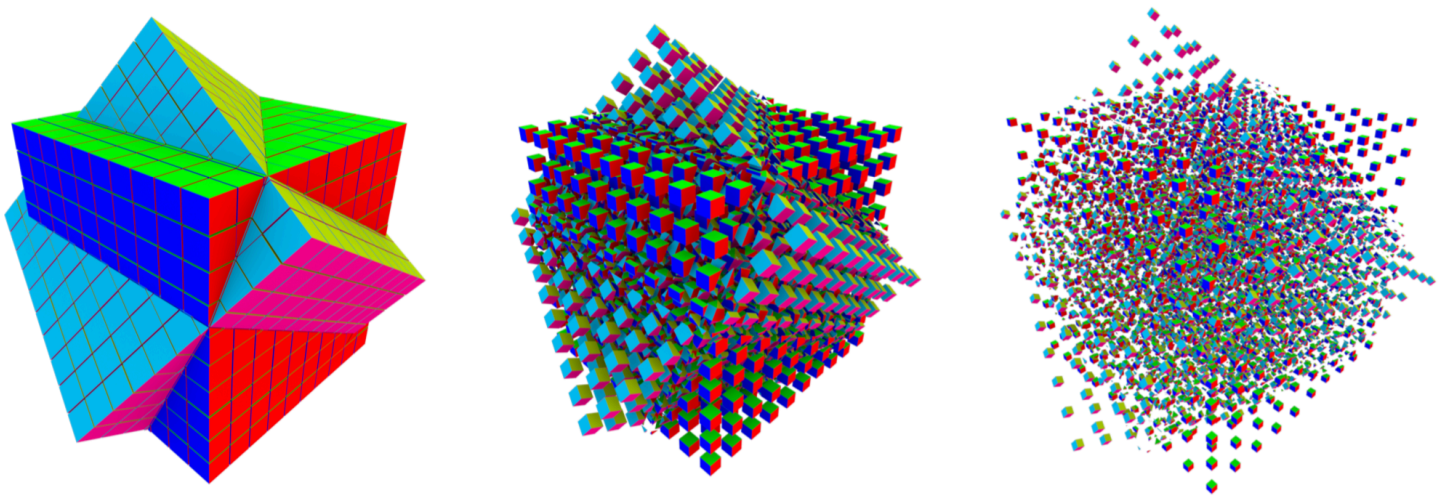


Fig. 7. Merge of two 10^3 3-complexes. In the merged complex we get 8655 3-cells, 26600 2-cells (faces), 26732 1-cells (edges), and 8787 0-cells (vertices). Please note that the Euler characteristic is $\chi = \chi_0 - \chi_1 + \chi_2 - \chi_3 = 8787 - 26732 + 26600 - 8655 = 0$, since the decomposed 3-complex (no its exploded image!) is homeomorphic to the n-sphere, where $\chi = 1 + (-1)^n$, with $n = 3$. Ratio new/old 3-cells is 4.3275.

Arrangement of cellular complexes: a quick tour

Paoluzzi, Shapiro, DiCarlo, 2018 (submitted)

ALGORITHM 3: *Fragmentation of 2-cells*

Input: $\mathcal{S}_2 \subset \mathcal{S}_{d-1}$ # *collection* of all 2-cells from $\mathcal{S}_{d-1}$ input in $\mathbb{E}^d$
Output: $[\partial_2]$ # CSC signed matrix
$\widetilde{\mathcal{S}_2} = \emptyset$ # initialisation of collection of local fragments
for $\sigma \in \mathcal{S}_2$ **do** # for each 2-cell σ in the input set
  $M = SubManifoldMap(\sigma)$ # affine transform s.t. $\sigma \mapsto x_3 = 0$ subspace
  $\Sigma = M\,\mathcal{I}(\sigma)$ # apply the transformation to (possible) incidencies to σ
  $\mathcal{S}_1(\sigma) = \emptyset$ # collection of line segments in $x_3 = 0$
  for $\tau \in \Sigma$ **do** # for each 2-cell τ in Σ
    $\mathcal{P}(\tau), \mathcal{L}(\tau) = \emptyset, \emptyset$ # intersection points and int. segment(s) with $x_3 = 0$
    for $\lambda \in X_1(\tau)$ **do** # for each 1-cell λ in $X_1(\tau)$
      if $\lambda \not\subset \{\mathbf{q} \,|\, x_3(\mathbf{q}) = 0\}$ **then** $\mathcal{P}(\tau)$ += $\{\mathbf{p}\}$ # append the intersection point of λ with $x_3 = 0$
    end
    $\mathcal{L}(\tau) = Points2Segments(\mathcal{P}(\tau))$ # Compute a set of collinear intersection segments
    $\mathcal{S}_1(\sigma)$ += $\mathcal{L}(\tau)$ # accumulate intersection segments with σ generated by τ
  end
  $X_2(\sigma) = \mathcal{A}(\mathcal{S}_1(\sigma))$ # arrangement of σ space induced by a soup of 1-complexes
  $\widetilde{\mathcal{S}_2}$ += $M^{-1}\,X_2$ # accumulate local fragments, back transformed in $\mathbb{E}^d$
end
$[\partial_1] = QuotientBases(\widetilde{\mathcal{S}_2})$ # identification of 0- and 1-cells using *kd*-trees and canonical LAR
$[\partial_2] = Algorithm\_1([\partial_1])$ # output computation GRANDE STRONZATA
return $[\partial_2]$

Arrangement of cellular complexes: a quick tour

Paoluzzi, Shapiro, DiCarlo, 2018 (submitted)

ALGORITHM 1: *Computation of signed* $[\partial_d^+]$ *matrix*

```
/* Pre-condition: d equal to space dimension, s.t. (d − 1)-cells are shared by two d-cells */
/*                                                                                        */
Input: [∂_{d−1}] # CSC signed matrix
Output: [∂_d^+] # CSC signed matrix
m, n = [∂_{d−1}].size; marks = Zeros(n)  # initializations
while Sum(marks) < 2n do
    σ = Choose(marks)        # select the (d − 1)-cell seed of the column extraction
    if marks[σ] == 0 then [c_{d−1}] = [σ]
    else if marks[σ] == 1 then [c_{d−1}] = [−σ]
    [c_{d−2}] = [∂_{d−1}] [c_{d−1}]        # compute boundary c_{d−2} of seed cell
    while [c_{d−2}] ≠ [] do        # loop until boundary becomes empty
        corolla = []
        for τ ∈ c_{d−2} do        # for each hinge τ cell
            [b_{d−1}] = [τ]^t [∂_{d−1}]        # compute the τ coboundary
            pivot = {|b_{d−1}|} ∩ {|c_{d−1}|}        # compute the τ support
            if τ > 0 then adj = Next(pivot, Ord(b_{d−1}))        # compute the new adj cell
            else if τ < 0 then adj = Prev(pivot, Ord(b_{d−1}))
            if ∂_{d−1}[τ, adj] ≠ ∂_{d−1}[τ, pivot] then corolla[adj] = c_{d−1}[pivot]        # orient adj
            else corolla[adj] = −(c_{d−1}[pivot])
        end
        [c_{d−1}] += corolla        # insert corolla cells in current c_{d−1}
        [c_{d−2}] = [∂_{d−1}] [c_{d−1}]        # compute again the boundary of c_{d−1}
    end
    for σ ∈ c_{d−1} do marks[σ] += 1        # update the counters of used cells
    [∂_d^+] += [c_{d−1}]        # append a new column to [∂_d^+]
end
return [∂_d^+]
```

Julia implementation

Julia implementation (in the works)

https://github.com/cvdlab/LinearAlgebraicRepresentation.jl

References