Pioneer of Solid Modeling Association

Alberto Paoluzzi

Symposium on Solid and Physical Modeling (SPM-2017)

The context of our start in Solid Modeling: 1984-1988

Solid modeling R&D was at the time using Lisp-machines!

Figure 1: Symbolics LISP Machine, from Google NY office computer museum

Figure 2: IBM XT personal computer

Figure 3: Apple Macintosh SE

We started on IBM first PCs and Apple Macintosh writing Solid Modeling software in Pascal

Minerva: the first solid modeler on a PC (1985–)

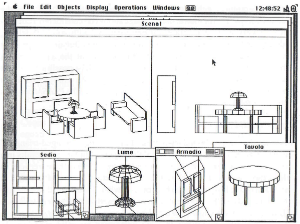


Figure 1: The window-based user interface of *Minerva*, with a structure definition

Figure 4: Apple GUI

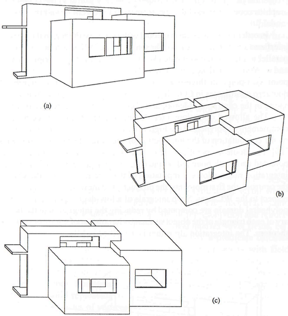


Figure 3: Solid model of the Oud's Mathenesse (Rotterdam, 1923) generated by *Minerva*. (a) Two-point perspective; (b) three-point perspective; (c) dimetric Cavalier axonometric view

Figure 5: Standard views of Mathenesse building

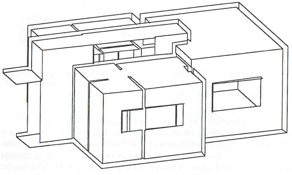


Figure 4: Hidden surface removed view of the complemented Mathenesse building

Figure 6: Boolean Algebra over Linear Polyhedra, CAD 1990

Featuring hierarchical assemblies, Boolean ops, parametric surfaces, integration of polynomials, HSR algorithms, full-fledged Apple GUI

PLaSM: FL-based Language for Solid Modeling (1992-)

Programming language for solid variational geometry (CAD 92, TOG 95)

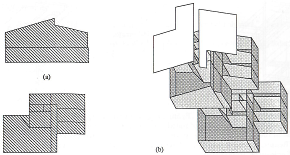


Figure 6: (a) The arguments *Plan* and *Section* of an operation && (intersection cell-by-cell of extrusions) in the *PLASM* language; (b) The two-dimensional polyhedral complex obtained by the evaluation of the expression @2:(*Plan* && *Section*), represented as an exploded drawing

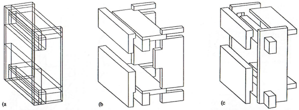


Figure 7: (a) A sequence (Beams, Roofs, Enclosures, Partitions) of isothetic (interpenetrating) pairs of parallelopipeds, for sake of simplicity assumed to be in the same coordinate space; (b) *PLASM* evaluated expression *STRUCT*: (Beams, Roofs, Enclosures, Partitions); (c) STRUCT: (Beams, Partitions, Enclosures, Roofs)

Example 1.5.6 (Building facade)
An example is given in Script 1.5.6, and shown in Figure 1.6a, where a 2D complex is generated by Cartesian product of 1D complexes produced by the `QUOTE` operator. Several other examples of the `QUOTE` operator are given in Section 1.6.2.

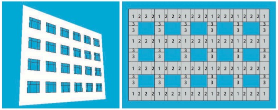


Figure 1.6 (a) the 2D complex generated as `facade:<6,4>` by combining `QUOTE`, product and 1-skeleton operators (b) the generating scheme of `facade` panels

INTRODUCTION TO FL AND PLASM 31

The `facade` generating function works by assembling three 2D Cartesian products of alternating 1D complexes, produced by `Q:xRithm`, `Q:xVoid` and by `Q:yRithm`, `Q:yVoid`, respectively. In particular, the `xRithm` sequence contains the numeric series used in the x direction; analogously `yRithm` for the y direction. Conversely, `xVoid` and `yVoid` host the series with opposite signs of elements. So, the first three Cartesian products in the `STRUCT` sequence produce a sort of checkboard covering that follows the scheme given in Figure 1.6b.

Script 1.5.6 (Building facade)

```
DEF facade (n,m::IsIntPos) = STRUCT:<
   Q:xRithm * Q:yRithm,
   Q:xVoid * Q:yRithm,
   Q:xRithm * Q:yVoid ,
   @1:(Q:xVoid * Q:yVoid) >
WHERE
   xRithm = ##:n:<5,-2,-5,-2> AR 5,
   yRithm = ##:m:<7,-5,-2> AR 7,
   xVoid = AA:-:xRithm,
   yVoid = AA:-:yRithm
END;
```

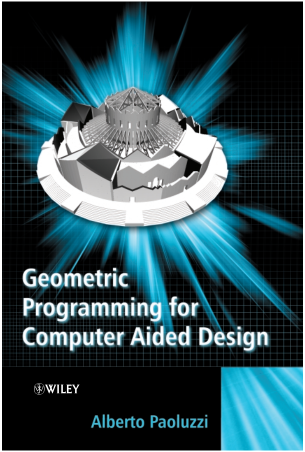


Figure 7: GP4CAD (2003)

Multidimensional representations and algorithms

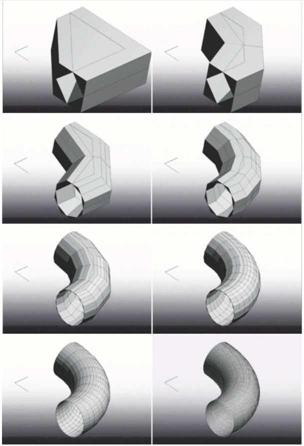


Figure 8: Dataflow refinement of rational B-spline with BSP nodes (SM&A, 2004)

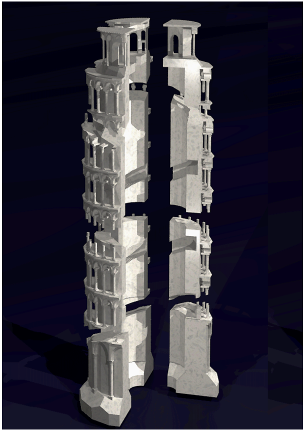


Figure 9: Parallel & distributed computing (IJCGA, 2008)

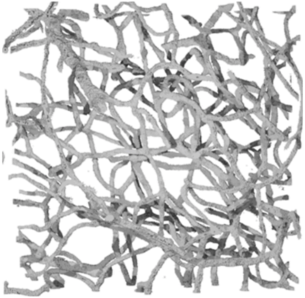


Figure 10: Topologically exact LAR models of microvessels between neurons (CAD&A, 2016)

LAR – Linear Algebraic Representation: $M \Leftrightarrow R$

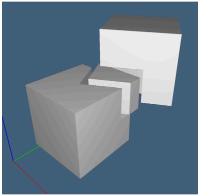
In development with Julia + CUDA + Sparse Tensorflow

models $M =$ category of chain complexes:

$$C_3 \underset{\partial_3}{\overset{\delta_2}{\leftrightarrows}} C_2 \underset{\partial_2}{\overset{\delta_1}{\leftrightarrows}} C_1 \underset{\partial_1}{\overset{\delta_0}{\leftrightarrows}} C_0$$

reprs $R =$ sparse matrices and vectors

(CAD,14) and (in development, 2017)



```
W = [[1.,1.,0.],[1.,0.,0.],[0.,1.,0.],[0.,0.,0.],[0.5,0.5,1.],
[0.5,0.5,0.5],[0.2113,1.,1.],[0.2113,1.,0.5],[0.5,0.5,1.5],
[0.,1.366,0.5],[0.,1.366,1.5],[1.,0.7887,0.5],[1.,1.,0.5],
[1.366,1.,0.5],[0.866,1.866,0.5],[1.,1.,1.],[1.,0.7887,1.],
[0.,1.,1.],[0.,0.,1.],[1.,0.,1.],[1.366,1.,1.5],
[0.866,1.866,1.5]]

CW = [[4,5,6,7,11,12,15,16,
[0,1,2,3,4,5,6,7,11,12,16,17,18,19],
[4,6,7,8,9,10,11,12,13,14,15,16,20,21]]

FW = [(5,7,11,12),(8,10,20,21),(6,7,12,15),
(11,12,15,16),(9,10,14,21),(2,3,17,18),(1,3,18,19),
(4,5,11,16),(13,14,20,21),(4,6,15,16),(4,5,6,7),
(0,1,2,3),(4,6,16,17,18,19),(4,6,7,8,9,10),
(7,9,11,12,13,14),(0,2,6,7,12,17),(0,1,11,12,16,19),
(4,8,11,13,16,20)]
```

Figure 11: user-readable LAR format

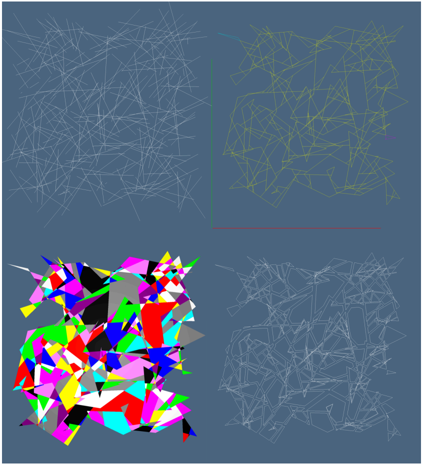
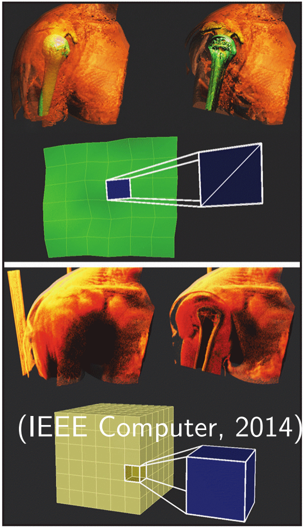


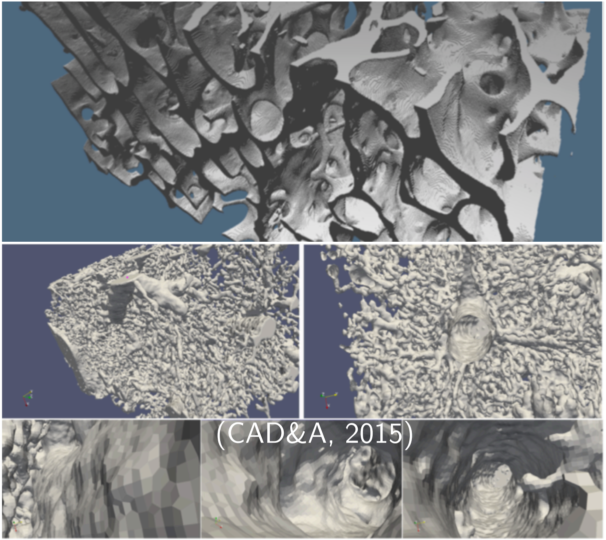
Figure 12: Plane arrangement from random line segments

Challenges: persolized biomedicine & virtual surgery

Solid Modeling Association should enter the IEEE-SA P.3333.2 WG !!



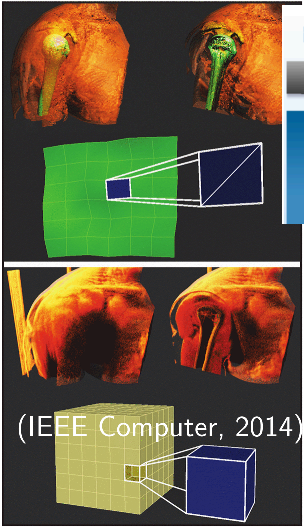
(IEEE Computer, 2014)



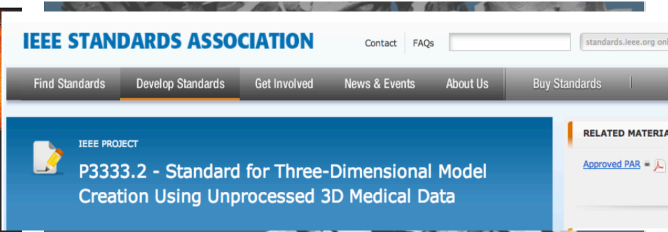
(CAD&A, 2015)

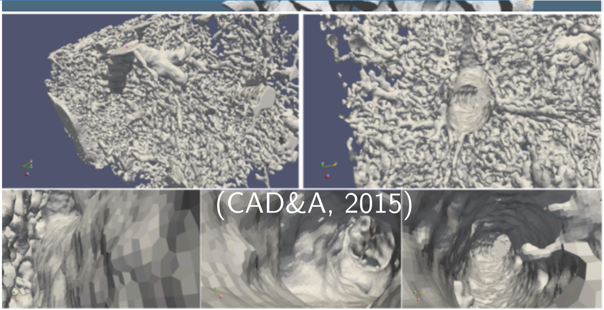
Challenges: persolized biomedicine & virtual surgery

Solid Modeling Association should enter the IEEE-SA P.3333.2 WG !!



(IEEE Computer, 2014)





(CAD&A, 2015)