Tensor Product Volumes

Asaf Tzur, Class 236716, CS, Technion

Content

- Reminder and Prologue
- Volumetric modelling
    - *A B-spline based Framework for Volumetric Object Modeling (F. Massarwi, G.Elber)*
- Graded Materials
    - *Fabricating Functionally Graded Material Objects Using Trimmed Trivariate Volumetric Representations (B.Ezair, G.Elber, D.Dikovsky)*
- Questions

Reminder: Tensor Product Surfaces

A tensor product surface might look like:

$$h(u, v) = \sum_{j=0}^{n} \sum_{i=0}^{m} c_{i,j} \, f_i(u) \, g_j(v)$$

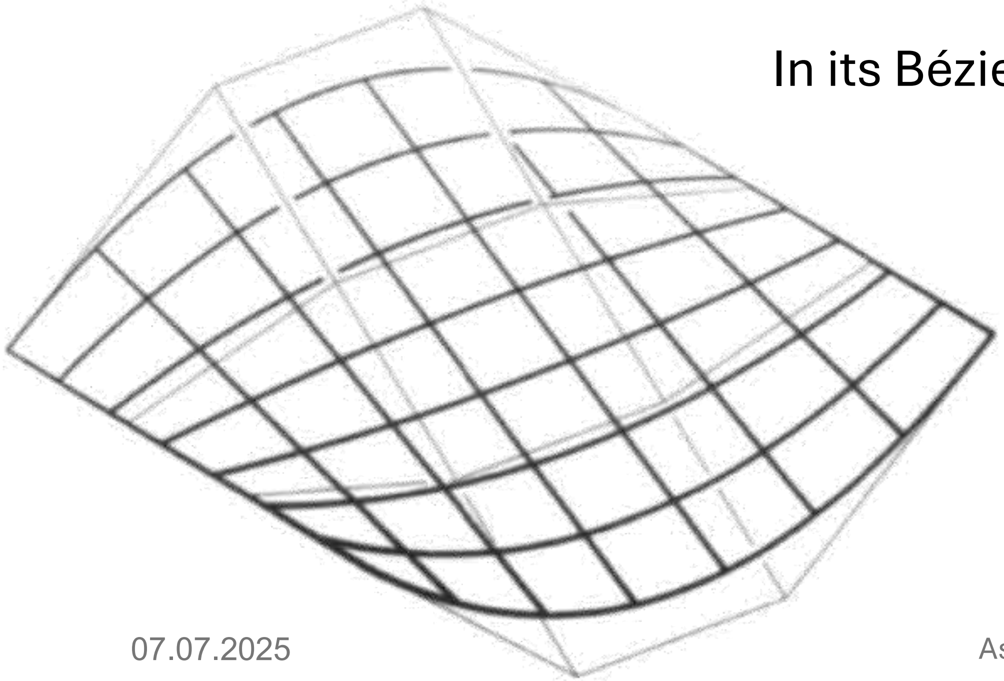
In its general form

Reminder: Tensor Product Surfaces

A tensor product surface might also look like:

$$\sigma_{m,n}(u,v) = \sum_{j=0}^{n}\sum_{i=0}^{m} P_{i,j}\,\theta_{i,m}(a,b;u)\,\theta_{j,n}(c,d;v)$$

In its Bézier form

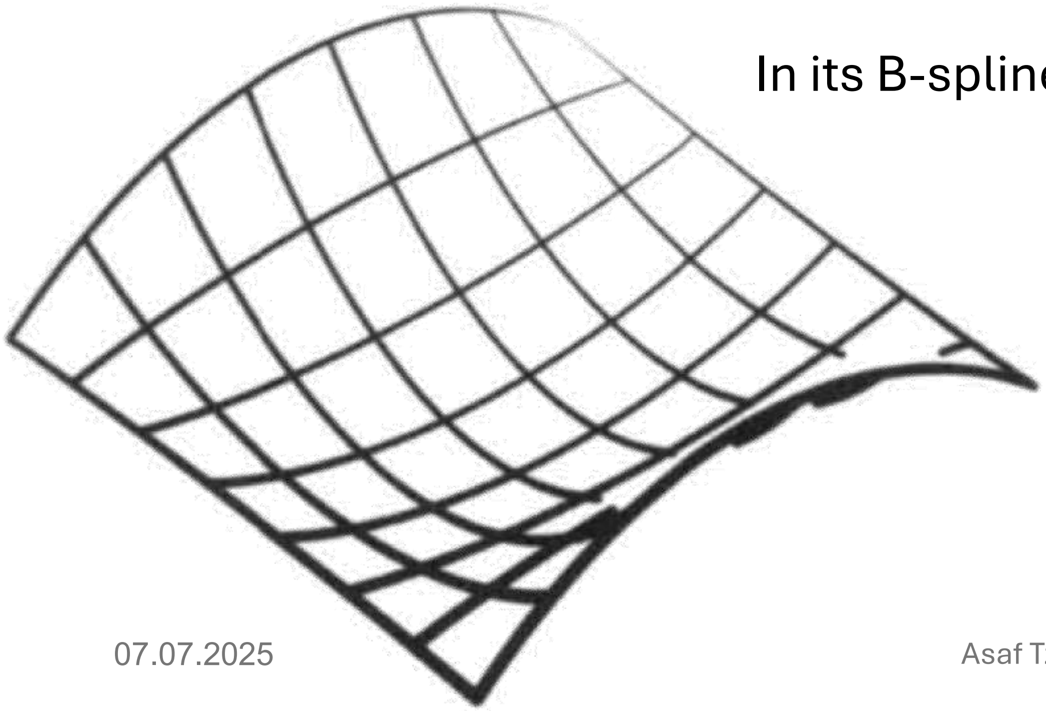


Reminder: Tensor Product Surfaces

A tensor product surface might also look like:

$$\sigma\,(u,v) = \sum_{j=0}^{n}\sum_{i=0}^{m} P_{i,j}\; B_{i,\mathrm{K}_u,\tau_u}(u)\; B_{j,\mathrm{K}_v,\tau_v}(v)$$

In its B-spline form



Reminder: Tensor Product Surfaces

Reminder: Tensor Product Surfaces

When Frank Gehry approached the design of his iconic **Guggenheim Bilbao museum**, he wanted to create something special, using shapes that haven’t been seen before.

He used an aerospace solid modeler to accomplish this.

Source: Wikipedia

Reminder: Tensor Product Surfaces

Source: Wikipedia

Reminder: Tensor Product Surfaces

- The outer structure surfaces are stitched to create a volume; this is a ***Boundary Representation*** of a volume!

- This leads us to our subject of the day: **Tensor Product Volumes**

- But first, A prologue

Prologue *industry*

- The industrial revolution
- In our day-to-day life we use fabricated & designed items

Prologue *CAD*

- **CAD – Computer aided Design**
    1. CAD software were popularized and innovated in the 1960s.
    2. Pierre Bezier, developed his own CAD software (UNISURF) to use in *Renault*.
    3. *BOEING* firstly used CAD software in the design of their 727 airplane, 1962.

Prologue *CAD*

- **State of the art CAD Systems**

  Catia, Nx, Creo, SolidWorks, SolidEdge and Inventor.

- **Boundary Representations (B-REP)**

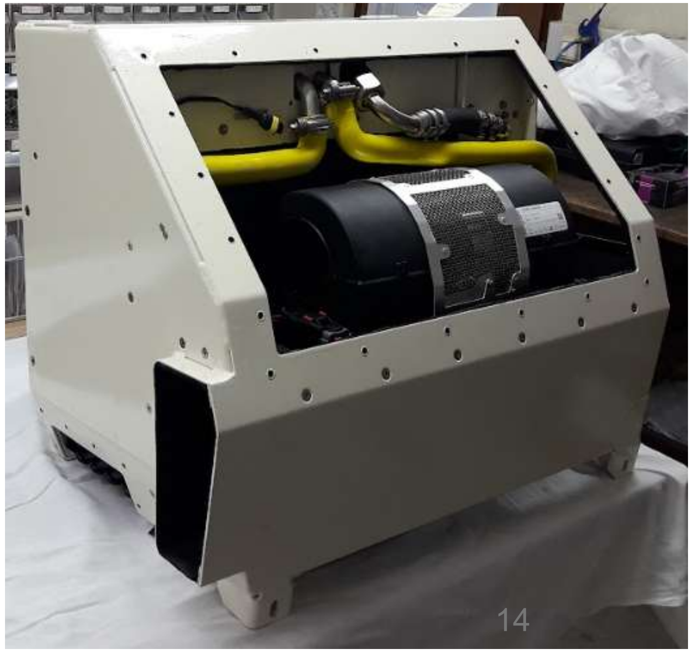
  CAD software are using boundary representations as their method to represent the geometry, where a manifold is described only by its outer surface, without an inner volume representation.

Prologue *traditional methods*

- **Common Manufacturing Methods**
  1. Plastic Deformation of metals: Rolling, Spinning, Drawing, Bending etc.
  2. High Temp: Injection molding, Sand Casting, Welding etc.
  3. Machining: Turning (lathe), Milling

Prologue

TECHNION
Israel Institute
of Technology



Prologue *additive manufacturing*

- Create any shape, without of the many geometrical limitations.
- Lattices, Inner voids, weird angles, complex surfaces, multipart designs, etc.
- Adding material also allowing us to use heterogenous material scheme.

Prologue *additive manufacturing*

- **The needs from CAD software**
  - Different needs
  - B-Rep limits the possibilities
  - Slicing software are filling the design gap.

This can change, using Volumetric Representations in CAD systems.

Tensor Product Volumes

Now, lets dive in:

- *Geometric Modeling with Splines*, R.F. Riesenfeld, E. Cohen, G. Elber. Chapter 21 – an introduction chapter to trivariates.

Tensor Product Volumes *definition*

- A curve is a **univariate** function, a surface is a **bivariate** function

- ***Definition 21.1.*** *The tensor product B-spline function in three variables is called a* **trivariate** B-spline function *and has the form:*

$$T(u, v, w) = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} P_{i,j,k} B_{i,\mathrm{K}_u,\tau_u}(u) B_{j,\mathrm{K}_v,\tau_v}(v) B_{w,\mathrm{K}_w,\tau_w}(w)$$

Where:

$$\mathrm{K}_\$ = the\ curve\ degree$$
$$\tau_\$ = the\ appropriate\ knot\ vector$$

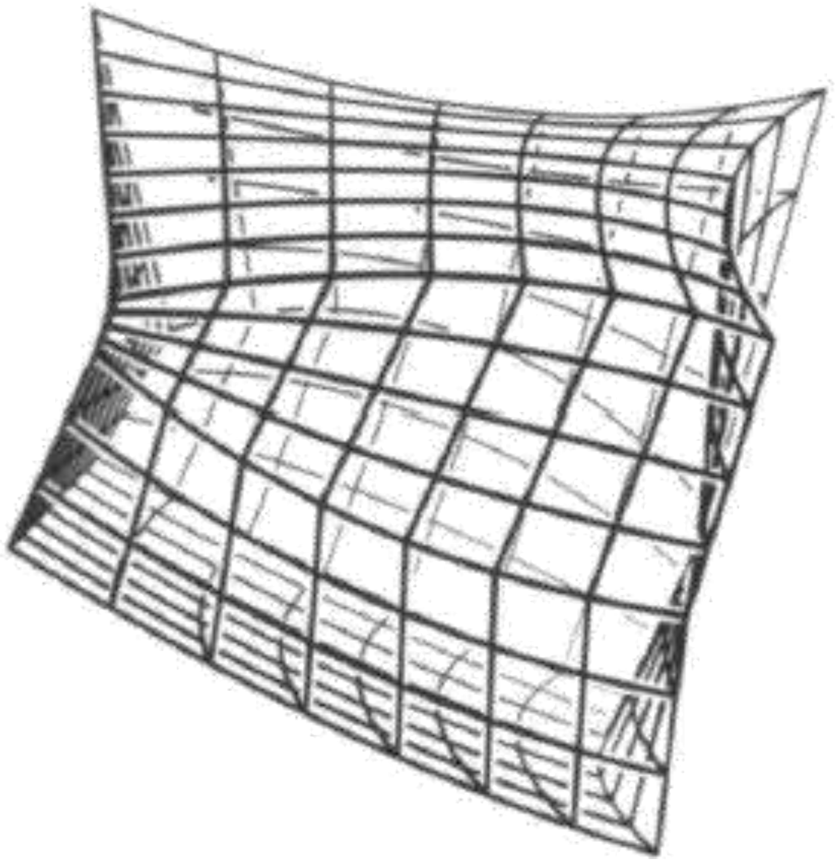
Tensor Product Volumes *intuition*

$$T(u,v,w) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} P_{i,j,k} B_{i,\mathrm{K}_u,\tau_u}(u) B_{j,\mathrm{K}_v,\tau_v}(v) B_{w,\mathrm{K}_w,\tau_w}(w)$$

- Extract the six isoparametric surfaces

U	V	W
$T(0,v,w)$	$T(u,0,w)$	$T(u,v,0)$
$T(1,v,w)$	$T(u,1,w)$	$T(u,v,1)$

- B-Spline surfaces from which isoparametric curves can be extracted.
- A volume from those 6 adjacent patches.



Tensor Product Volumes *scalar*

- The third parameter is time (or any other scalar)

$$T(u, v, t) = (1 - t)\sigma_1(u, v) + t\sigma_2(u, v)$$

- We now have a non volumetric trivariates
- Evaluated for t, a blend function of the two surfaces plots a shape change.

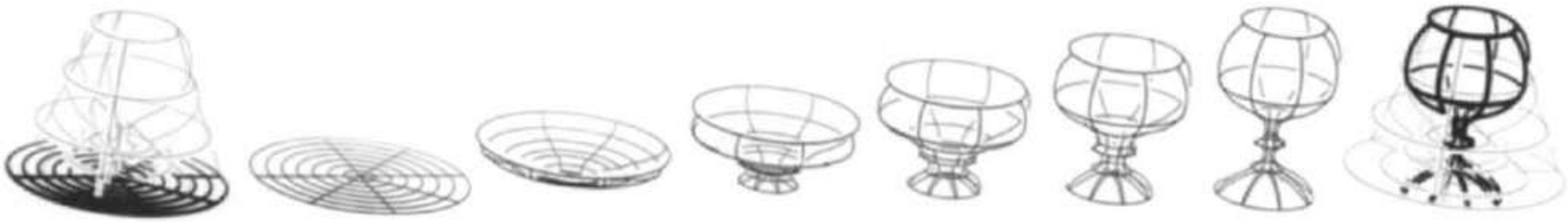


Figure 21.11. A continuous metamorphosis between a disk and a wine glasses, using a ruled volume. The trivariate is shown at the two end locations with the respective boundary (input) isoparametric surfaces in bold.

Volumetric modelling

[A B-spline based Framework for Volumetric Object Modeling]

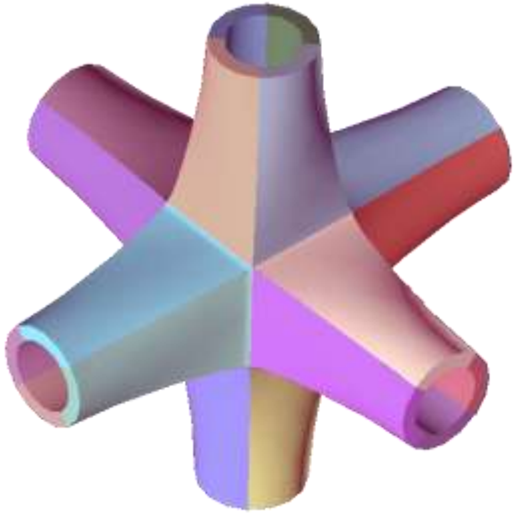
A work by Fady Massarwi and Gershon Elber

Volumetric modelling *introduction*

What is Volumetric Modeling?

- A method to represent 3D objects not only by their boundaries but also by their interior volume.

- This representation includes not only geometry but also internal attributes such as material properties and boundary conditions.

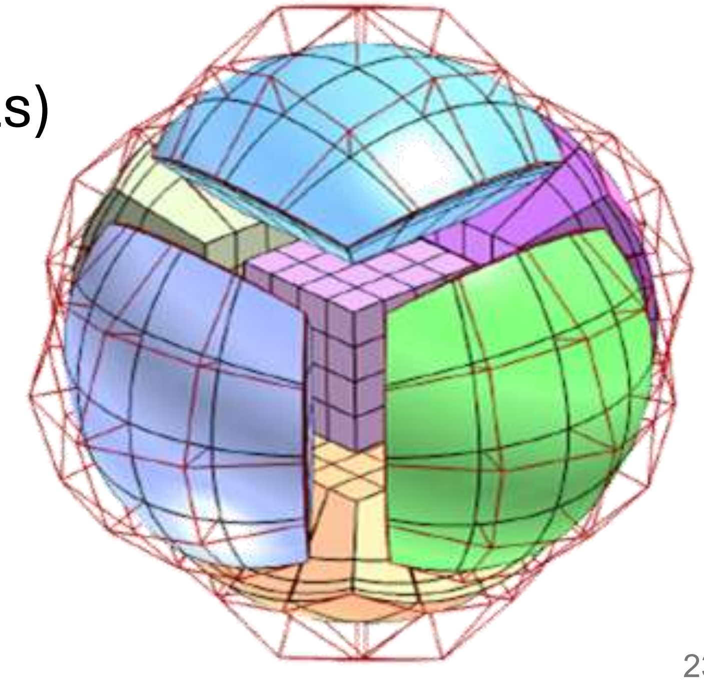


Volumetric modelling *introduction*

How does it work?

- It is a data structure that defines a **V-model** as a complex of volumetric cells (**V-cells**) that can represent both geometry and attributes. A V-model consists of one or more V-cells, each represented by trimmed B-spline trivariates.

- Example: A Sphere Modeled by 7 trivariates (V-cells)
  1 centric and 6 outer



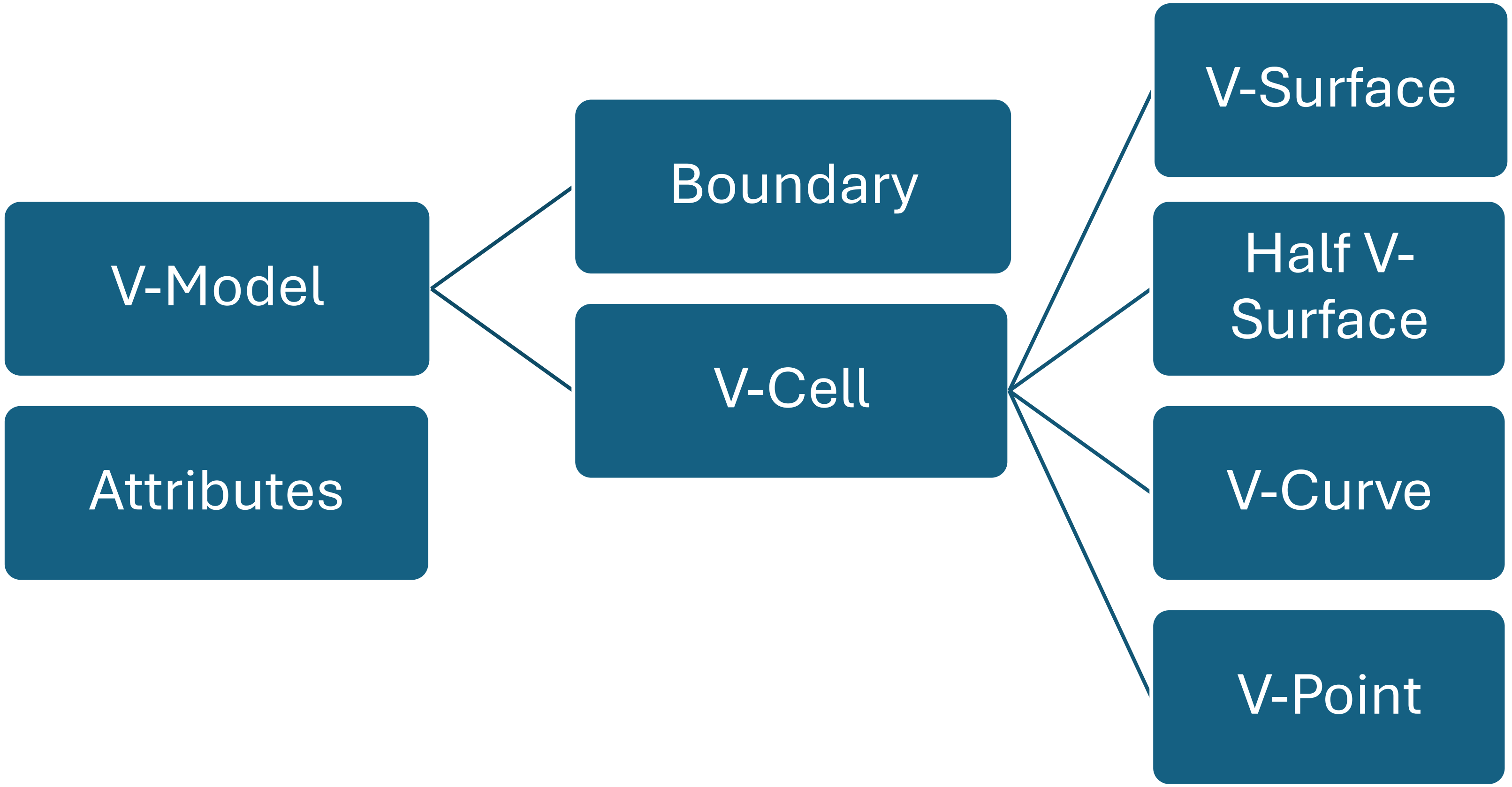
Volumetric modelling *introduction*

Importance of Volumetric Representation in Geometric Modeling

- Additive manufacturing (AM), and especially graded materials.
- Iso geometric analysis (IGA) and meshed based analysis.
- Design of micro-scale bone scaffolds.
- Topological optimizations in porous materials.

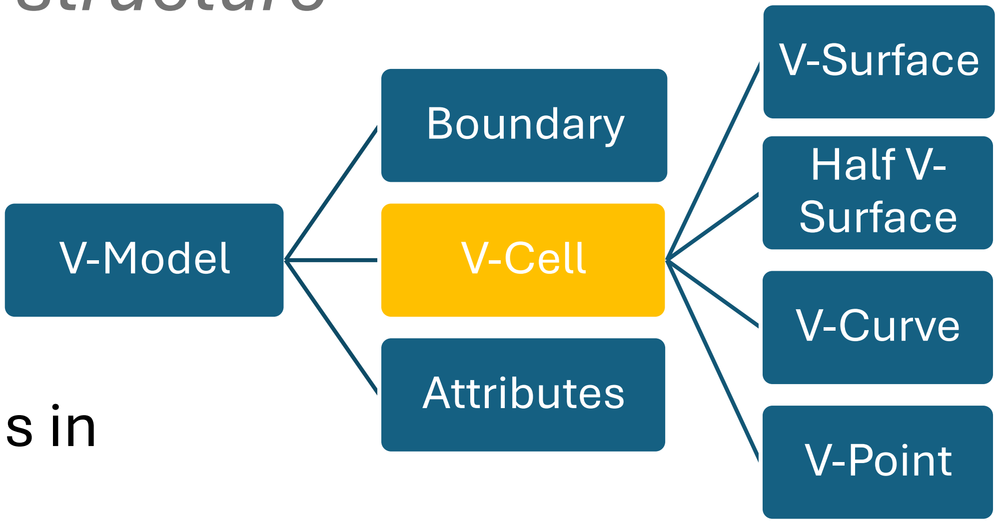
Volumetric modelling *methodology*

The data structure:



Volumetric modelling *data structure*

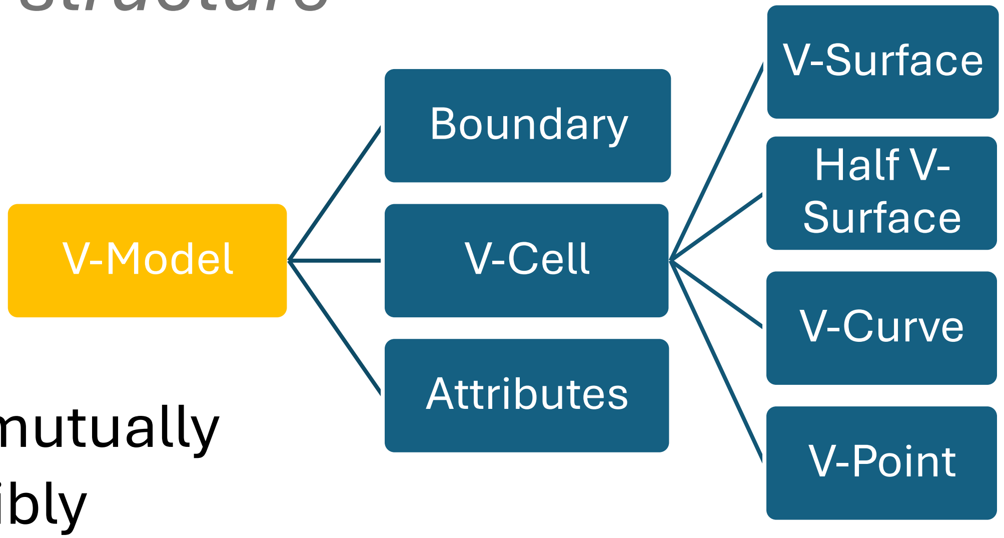
The data structure:



- **Definition 3.2.**
  A V-rep cell (V-cell) is a 3-manifold that is in the intersection of one or more B-spline tensor product trivariates. The sub-domain of the intersection is delineated by trimming surfaces.

Volumetric modelling *data structure*

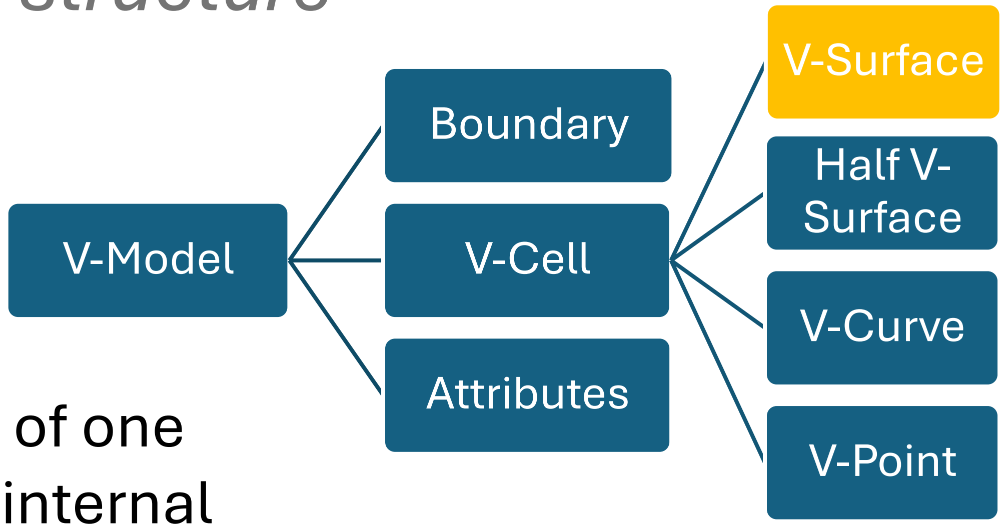
The data structure:



- **Definition 3.3.**
  A V-model is a complex of one or more (mutually exclusive) V-cells. Adjacent V-cells possibly share boundary (trimming) surfaces, curves or points.

Volumetric modelling *data structure*

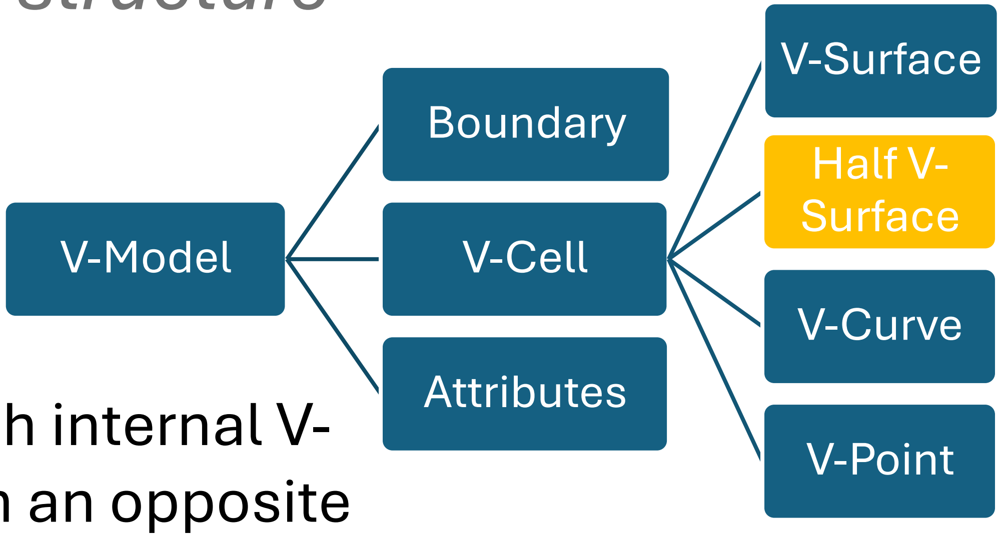
The data structure:



Definition 3.4.
A V-surface is a boundary trimming surface of one (or two adjacent) V-cell(s) of a V-model. An internal V-surface is shared between two adjacent V-cells, and a boundary V-surface belongs to one V-cell which is also a boundary surface of the entire V-model.

Volumetric modelling *data structure*

The data structure:



Definition 3.6.

A half-V-surface is a surface of a V-cell. Each internal V-surface is split into two half-V-surfaces with an opposite orientation. The half-V-surface holds references to:

1. Its own V-cell.
2. The other/neighboring V-cell (if exists).
3. Trimming loops in the parametric space of the surface.

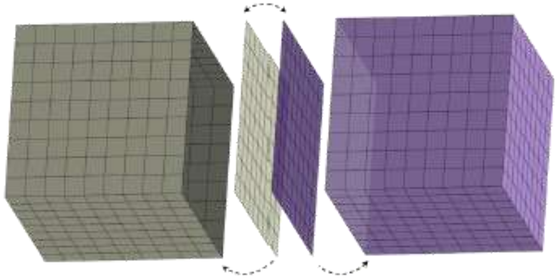
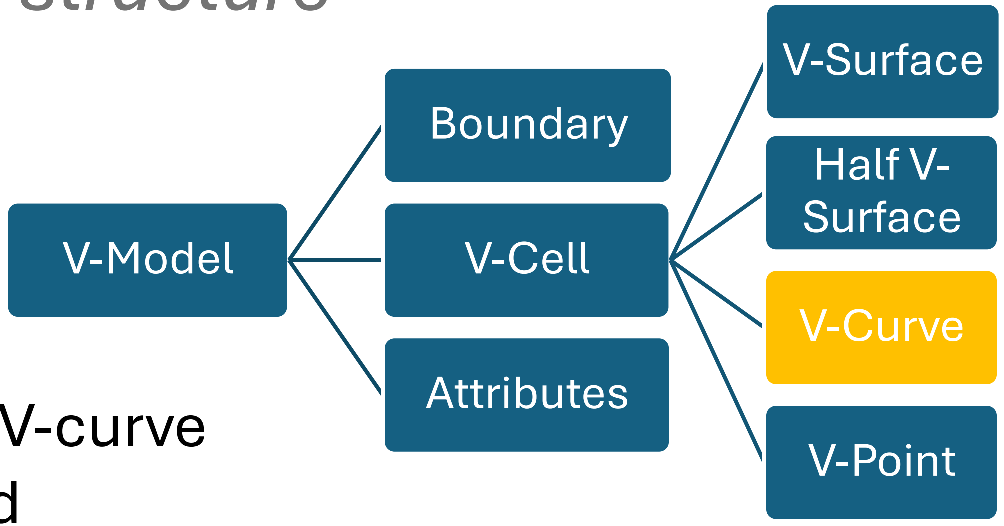


Figure 2: A shared V-surface boundary between two V-cells is split into two half-V-surfaces. Each half-V-surface references its V-cell and also references the adjacent half-V-surface.

Volumetric modelling *data structure*

The data structure:

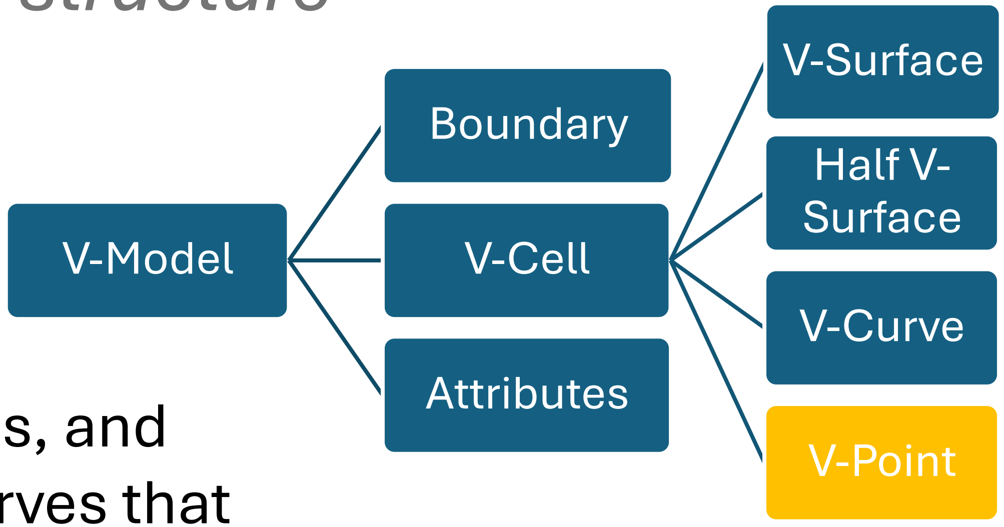


Definition 3.7.
A V-curve is a boundary curve of a V-cell. A V-curve can be shared between several (unbounded number of) V-cells and it hold references to:

1. V-surfaces that share this V-curve.
2. Two end points of the V-curve, of type V-point

Volumetric modelling *data structure*

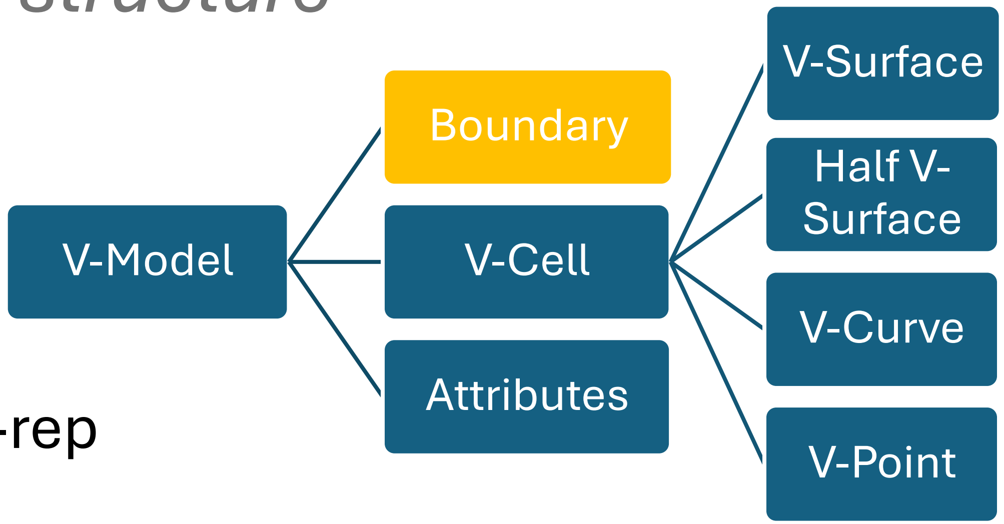
The data structure:



Definition 3.8.
A V-point is an intersection point of V-curves, and holds a list of (unbounded number of) V-curves that start or end at this point.

Volumetric modelling *data structure*

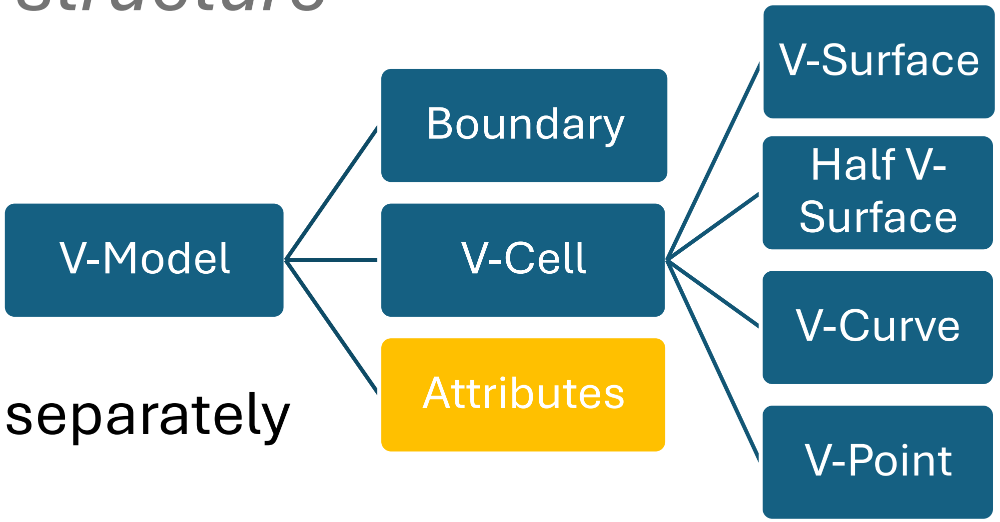
The data structure:



Definition 3.5.
The boundary of V-model VM, is a closed B-rep 2-manifold defined as the union of the boundary V-surfaces in VM. (V-surfaces that belong to one V-cell).

Volumetric modelling *data structure*

The data structure:

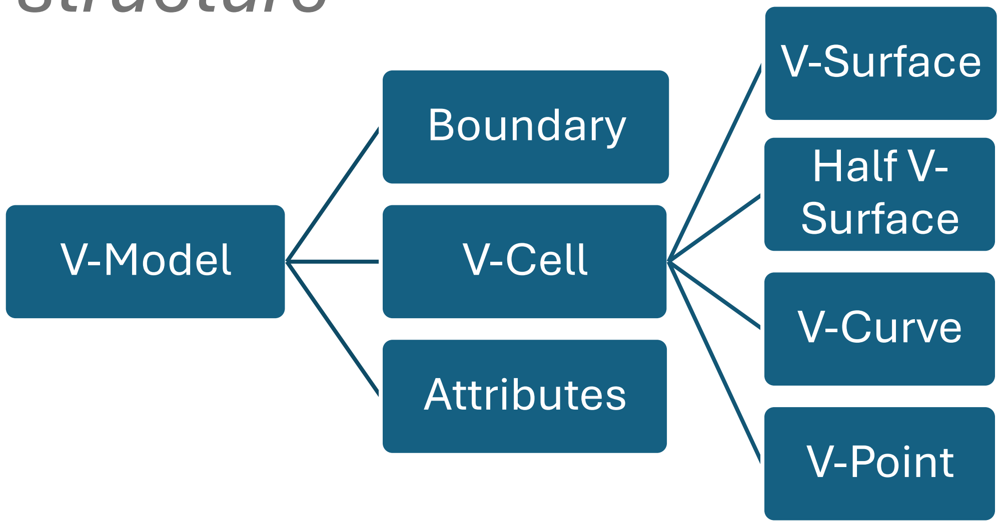


Attributes:

1. Can be stored to any level in the set separately
2. Examples:
    1. Materials
    2. Colors
    3. Math Functions
    4. Simulation data
    5. Strength data

Volumetric modelling *data structure*

The data structure:

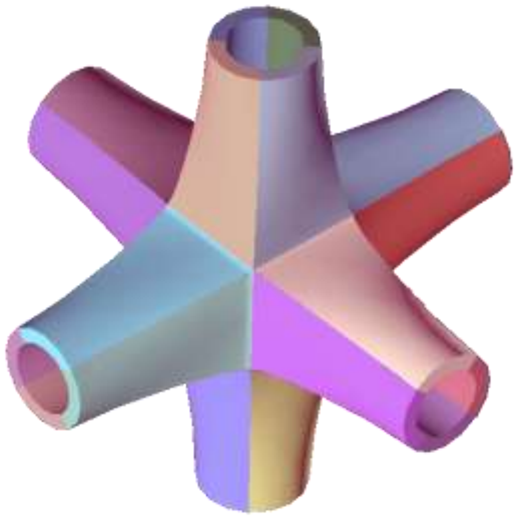


Possible queries on the data set:

1. V-curve's neighboring V-cells.
2. V-cell's neighboring V-cells.
3. Are two V-cells sharing a boundary surface? Sharing a boundary curve? Meet at exactly one point?

Volumetric modelling *operations*

A **Gluing operations** is defined to link V-cells into a coherent V-model with consistent topological relationships.



Volumetric modelling *Boolean operations*

- **Union**
- **Subtraction**
- **Intersection**

Are valid on V-Models

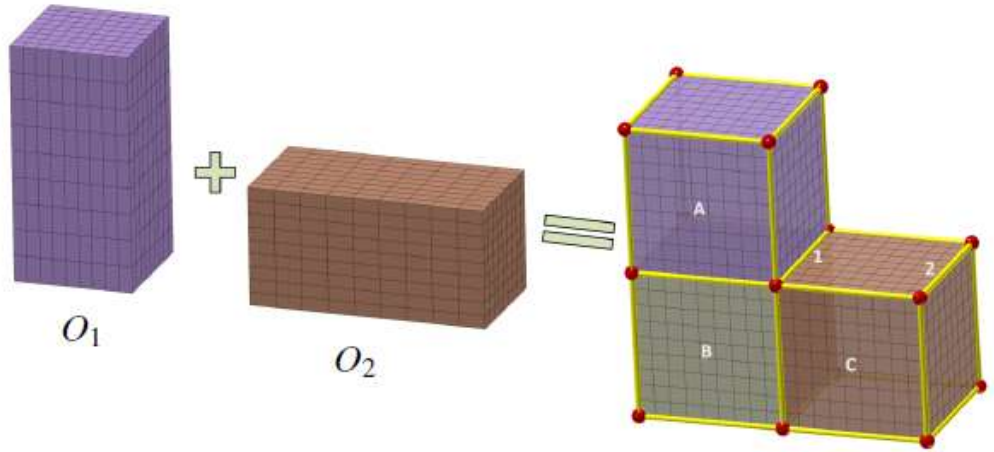


Figure 3: A V-model generated by uniting two box V-models, O_1 and O_2. The result V-model contains three V-cells, A, B and C. V-points are marked in red, and V-curves are displayed in yellow.

Volumetric modelling *Boolean operations*

- Boolean operations are well defined on B-Reps
- The algorithm for V-Rep will be based on the B-Rep
- Extended Boolean for V-Rep includes Added data, such as attributes.

Definition 4.1. *The* boundary *of a given V-cell V_C, ∂V_C, is a closed B-rep manifold defined as the union of the (trimming) half-V-surfaces of V_C. Similarly, let $\mathbf{S}_{\mathcal{T}\text{-}\mathcal{V}}(V_C)$ denote the set of all trivariates of V-cell V_C.*

Volumetric modelling *snippet*

Boolean Intersection:

1. B-Rep intersection operation
2. Following a V-Cell data intersection

Algorithm 1 V-Intersect: Intersection of two V-models

Input: V-models V_M^1, V_M^2;
Output: V_M^{1i2}, V-model of the intersection between V_M^1 and V_M^2;
Algorithm:

$\mathcal{V}_{Cells} := \emptyset$; /* Set of updated/new V-cells */
for all $V_C^i \in \mathcal{S}_{V_C}(V_M^1)$ **do**
  for all $V_C^j \in \mathcal{S}_{V_C}(V_M^2)$ **do**
    $\mathcal{B}_{i,j} := BrepBoolOP(\partial V_C^i, \partial V_C^j, \mathrm{INT})$;
    $\mathcal{TV} := \mathcal{S}_{\mathcal{TV}}(V_C^i) \cup \mathcal{S}_{\mathcal{TV}}(V_C^j)$;
    $\mathcal{V}_{Cells} := \mathcal{V}_{Cells} \cup \{C_{V_C}(\mathcal{B}_{i,j}, \mathcal{TV})\}$;
  end for
end for
V_M^{1i2} := Glue all V-cells in $\mathcal{V}_{Cells}$;

Volumetric modelling *Boolean operations*

V-Rep Possible Union & Blend function:

- Very beneficial for graded materials

$$A(p) = \begin{cases} \frac{d_{i2}^{b1}(p)A_1(p)+d_{i1}^{b2}(p)A_2(p)}{d_{i2}^{b1}(p)+d_{i1}^{b2}(p)}, & p \in V_M^1 \cap V_M^2, \\ A_1(p), & p \in V_M^1 \ \& \ p \notin V_M^2, \\ A_2(p), & p \in V_M^2 \ \& \ p \notin V_M^1. \end{cases}$$

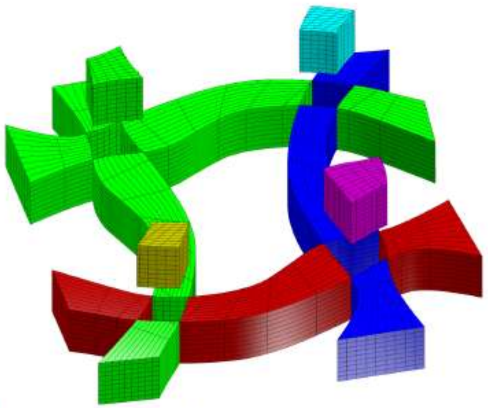


Figure 5: A Union between four V-models. The blending scheme used here simply sums the color attribute in the intersection volumes.

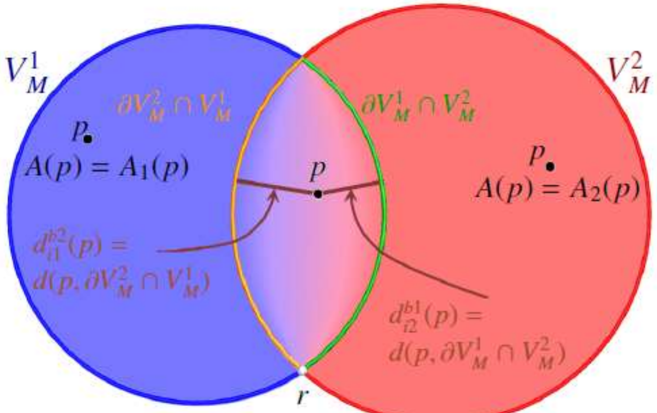


Figure 6: Attribute blending of two V-models V_M^1 and V_M^2. The red color of V_M^1 and the blue color of V_M^2 are blended using the blending scheme proposed in Equation (2).

Volumetric modelling *constructors*

- High-Level V-Rep **constructors** such as extrusion, ruled surface, volumes of revolution, along with **non-singular primitives** (e.g., spheres, tori), is supported.

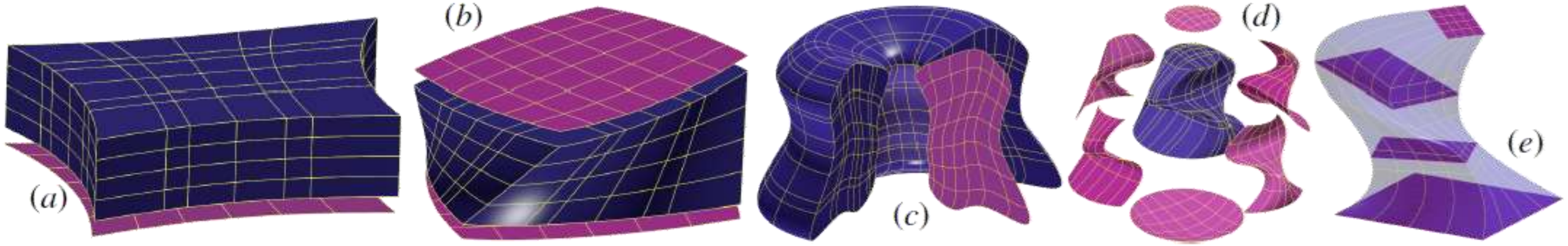
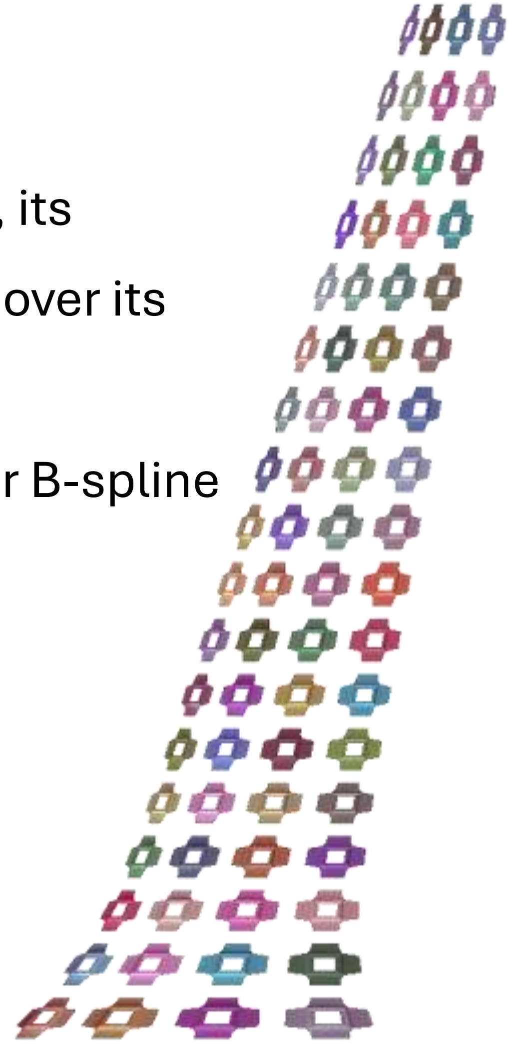


Figure 7: Constructors of trivariates: (a) A volume of extrusion (b) A ruled volume (c) Volume of revolution (d) Volumetric Boolean sum (e) A sweep volume. All these constructors yield a single B-spline trivariate.

Volumetric modelling *main achievements*

The main contributions of this V-rep framework are:

1. A Data structure for accurately representing a general freeform V-model, its geometry as well as scalar, vector, and tensor fields in its interior and/or over its boundaries.
2. Constructors of basic volumetric primitives as a complex of non-singular B-spline trivariates, such as a cylinder, a torus and a sphere and more advanced constructors such as ruled volumes and volumes of revolution.
3. Algorithms for Boolean operations over V-models
4. Almost seamless conversion from B-rep geometry.



Volumetric modelling *challenges and future*

Challenges and Future Directions of V-Rep

- Ensuring continuity across V-cells and handling small V-cells resulting from Boolean operations. Current V-model primitives are only C0 continuous;
- Small V-cells from Boolean operations can complicate .
- Advanced tools for volumetric fillets and blends are required for comprehensive modeling.
- The integration of T-splines and other representations could enhance the V-rep framework's capabilities

Graded Materials

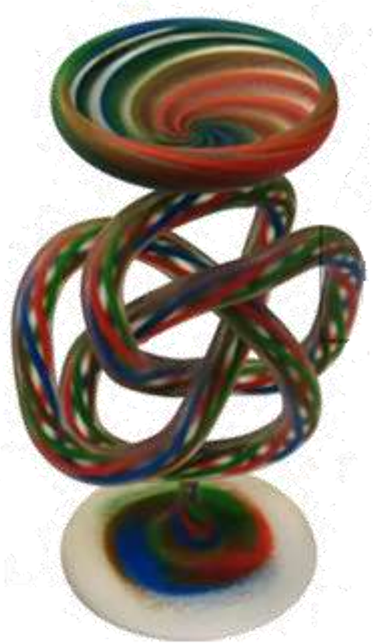
Fabricating Functionally Graded Material Objects Using Trimmed Trivariate Volumetric Representations

B.Ezair, G.Elber, D.Dikovsky

Graded Materials *introduction*

What is FGM?

Functionally graded materials (FGMs) are heterogeneous materials created by mixing/interleaving two or more materials, inside the volume of some object, in a way that varies according to position.



Graded Materials *introduction*

What is presented in the paper?

- A Method to fabricate functionally graded material objects using trimmed volumetric representations.
- An efficient slicing method.
- Manufacturing using additive manufacturing with multi-material 3D printers.

Graded Materials *introduction*

Possible uses (not mentioned in the paper)

Biomedical
- dental implants
- bone replacements
- Inner sole of a shoe also

Aerospace Components
- Thermal insulator combinations

Protective Armor & Structures
- Shock absorbers protectives
- Sound dampening building panels

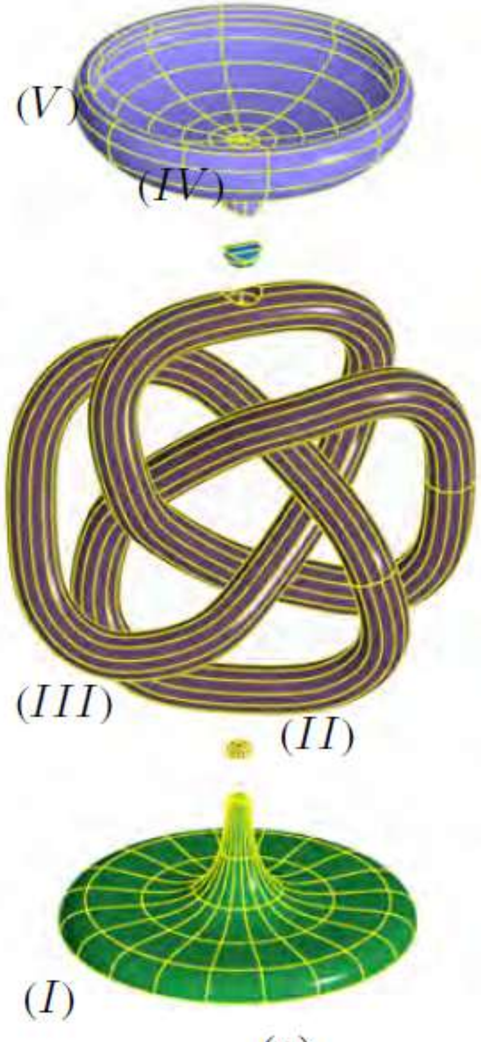
Graded Materials *Methodology*

How it was done:

1. FGM's are V-Cells almost as line in the V-Rep framework, (differs only by the single parametrization usage)
2. Each cell is set in the Euclidian space (u,v,w) and co

$$FGM_{cell}(u, v, w) = \{V(u, v, w), M(u, v, w), \mathcal{S}\}$$

$V(u,v,w) = A\ trivariate, the\ geometrical\ data$
$M(u,v,w) = Material\ Function$
$\mathrm{S} = \text{A list of trimming surfaces}$

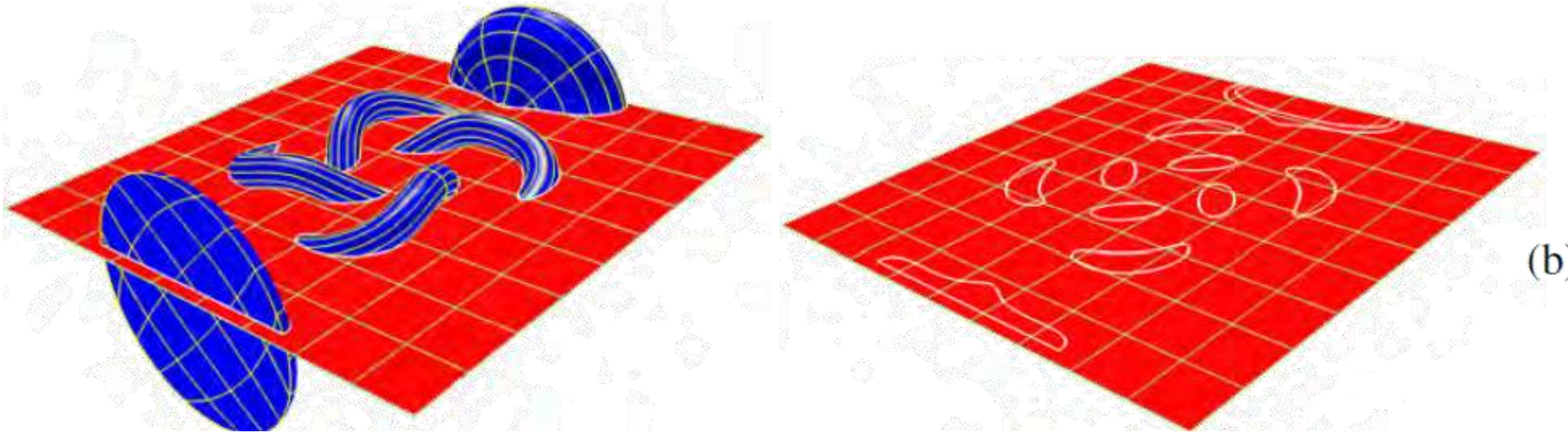


(a)

Graded Materials *Methodology*

Slicing:

1. Slicing the (geometrical data) V-Rep regularly
2. Each Z value gives a planar 2D contour of the 3D model

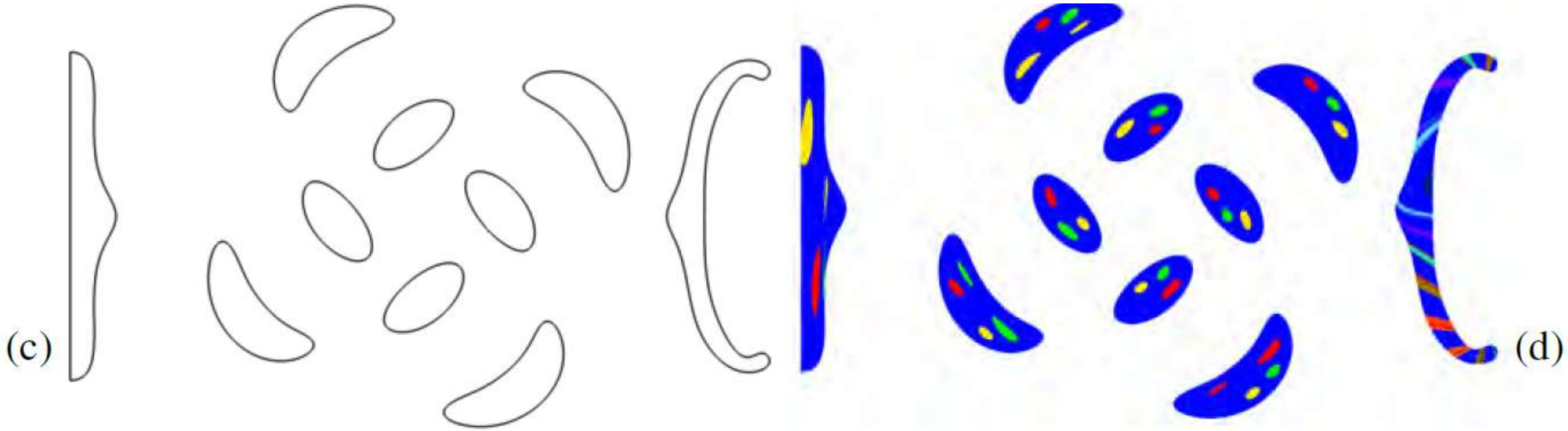


(a) (b)

Graded Materials *Methodology*

Slice Material Data:

1. Filling the slice outline with a representative color
2. The slice has now the grade material data needed for the printer to do the printing.



Graded Materials *deep dive*

XYZ to Parametric Coordination

1. When slicing, the data is flattened to XY location and z is constant.
2. Retrieving the material data requires to go back to (u,v,w) parametrization.
3. This is a “a classical inverse problem”, but computationally intensive

$$x = v_x(u, v, w)\,, y = v_y(u, v, w)\,, z = v_z(u, v, w)$$

$$V(u, v, w) = (v_x(u, v, w), v_y(u, v, w), v_z(u, v, w)) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \sum_{k=0}^{n_w} P_{ijk} B_i(u) B_j(v) B_k(w)\,,$$

Graded Materials *deep dive*

XYZ to Parametric Coordination

1. Built size of 255x252x200 [mm][mm][mm].
2. Resolution of 600DPI in XY -> 23 pixels per mm
3. 6000 X 5800 pixels per slice (Z direction not included)
4. 34,800,000 pixels in total per layer
5. Z direction is even finer! With ~ 63 pixels in a mm

Objet 260 Connex

Data is from Stratasys website:

1. https://support.stratasys.com/en/Printers/PolyJet-Legacy/Objet260-Connex-1-2-3

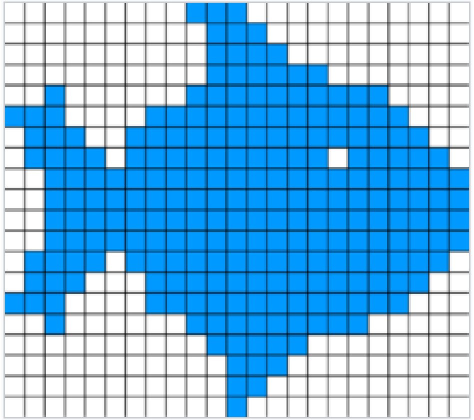
Graded Materials *deep dive*

2 Methods of optimization used in the algorithm:

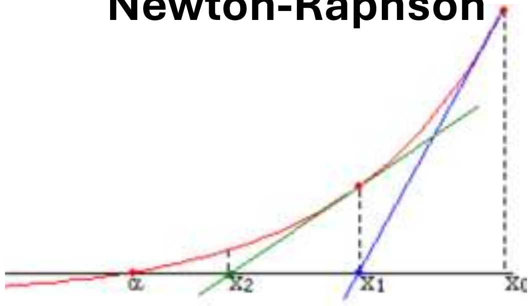
1. Determine if we are inside/outside the outline, as used in computer graphics, using: ***simple rasterization strategy***
   - This saves calculations for areas outside our geometry

2. Numeric tracing, such as Newton iteration method, was used to search data “closed by” a valid point efficiently.

Rasterization



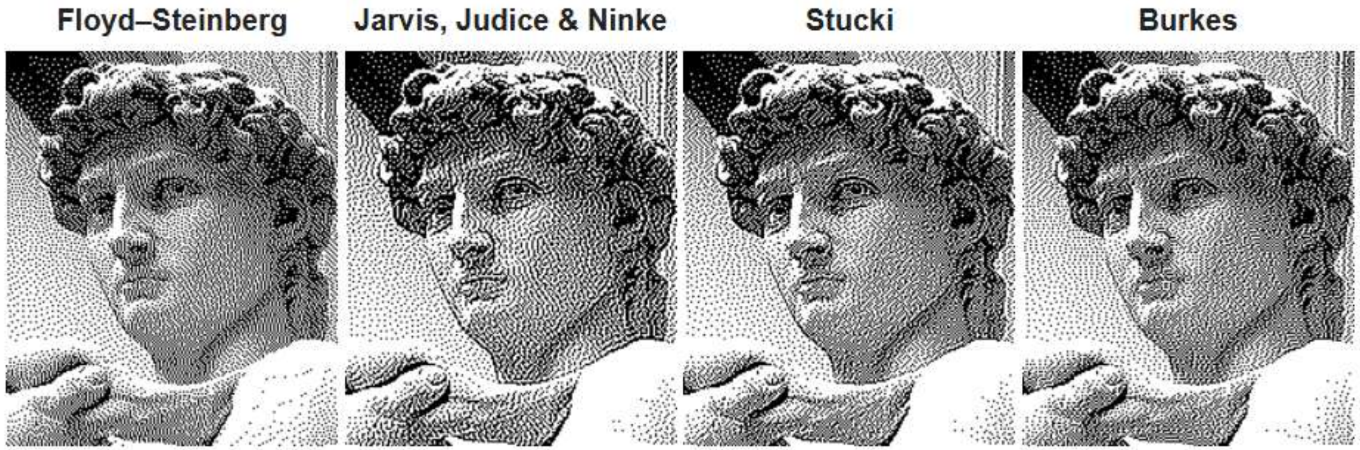
Newton-Raphson



Graded Materials *deep dive*

Dithering:

1. Mostly there is no possibly to print **multi-material** in a single **Pixel**
2. Dithering algorithm was called to tackle the issue



(Wikipedia: a few dithering samples

Graded Materials *snippet*

Algorithm 1 GenerateSliceBitmaps

Input:

(1) $FGM_{obj} = \{FGM_{cell(1)}, \ldots, FGM_{cell(n)}\}$, a complex of n FGM cells;

(2) m, the number of materials;

(3) z, the height of the slice;

(4) x_{res}, y_{res}, the resolution (in pixels) of the required bitmap(s);

Output:

(1) $B = \{B_1, \ldots, B_m\}$, a set of bitmaps;

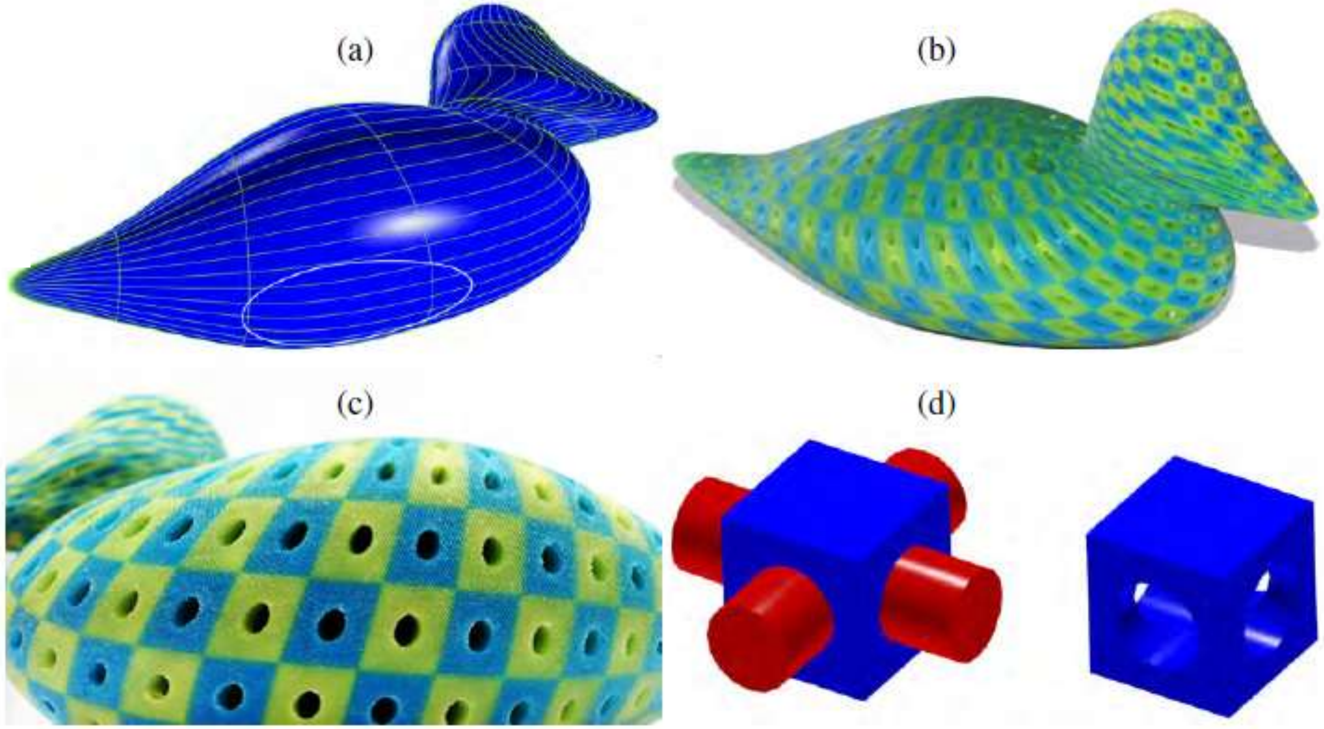
Algorithm:

$(x_{min}, x_{max}, y_{min}, y_{max}) := BoundingRectangleXY(FGM)$;
$(dx, dy) := (\frac{x_{max}-x_{min}}{x_{res}}, \frac{y_{max}-y_{min}}{y_{res}})$;
$M_g := EmptyGrid(x_{res}, y_{res})$; // A 2D matrix of vectors.
for $i = 1; i \leq n; i = i + 1;$ **do** // Iterate over all the cells.
    $(V, M, S) := FGM_{cell(i)}$;
    $\mathcal{O} := SSI(\mathcal{S}, Plane(z))$; // Surfaces-plane outline intersection.
    for $x = x_{min}; x \leq x_{max}; x = x + dx$ **do**
        $\mathcal{Y}$:= Intersection($Line$(X = x), $\mathcal{O}$); // A set of y-aligned intervals.
        for all $Y \in \mathcal{Y}$ **do**
            $(u, v, w) := FullSolve(V, x, \min_y(Y), z)$;
            for $y = \min_y(Y); y \leq \max_y(Y); y = y + dy$ **do**
                $(u, v, w) := NumericSolve(V, x, y, z, u, v, w)$;
                if $V(u, v, w) \neq (x, y, z)$ **then**
                    $(u, v, w) := FullSolve(V, x, y, z)$;
                end if
                $(x_i, y_i) = \left(floor\left(x_{res}\frac{x-x_{min}}{x_{max}-x_{min}}\right), floor\left(y_{res}\frac{y-y_{min}}{y_{max}-y_{min}}\right)\right)$;
                $M_g(x_i, y_i) := M(u, v, w)$;
            end for
        end for
    end for
end for
$M_{dither} := Dither(M_g)$;
$B := BitmapPerMaterial(M_{dither}, m)$;
Return B;

Graded Materials *results*

Benchmarking:

Optimization	Time [sec]	Full Solve [%]
none	22590	100
previous point tracing	71	0.2532
previous line tracing	18	0.0205
full tracing	15	0.0044



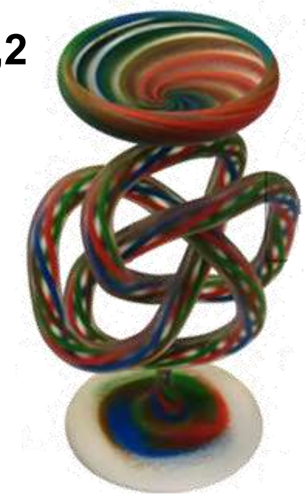
Graded Materials *results*

Benchmarking:

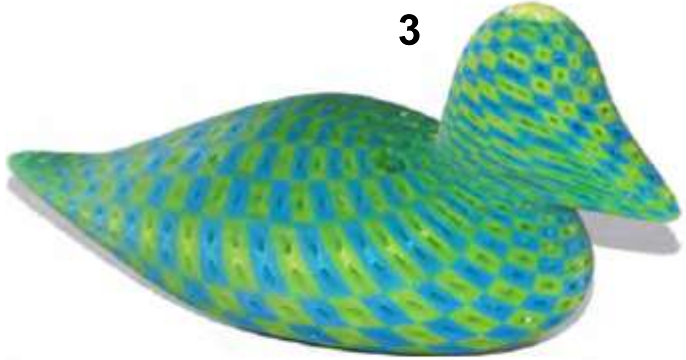
Table 2: *Statistics for the models in our fabricated examples.*

Figure	Bitmap Size	# Slices	# Voxels Total, and Inside Model [$\times 10^9$]	Average Time Per Slice (4 Threads) [sec]	Total Time (4 Threads) [hour]	Full Solve [%]
1, 2	(2944 x 1466)	1914	8.2, 0.45	5.75	3.05	0.1033
4	(2976 x 1477)	2110	9.2, 0.54	3.14	1.84	0.1008
3	(3040 x 1512)	2918	13.4, 2.69	10.4	8.5	0.0055

1,2



3



4

Graded Materials *conclusion*

1. Trimmed Volumetric Representations (V-reps) are effective for modeling and fabricating Functionally Graded Material (FGM) objects, offering advantages over traditional B-reps.

2. V-reps are well-suited for integration into advanced CAD tools, supporting both geometric and material representations in a unified framework.

3. The proposed slicing algorithm works with printers that use bitmap inputs but is adaptable to other formats, as long as material data can be discretized.

Graded Materials *conclusion*

Suggested improvements:

1. Exploring flood-fill sampling instead of raster-scan could improve the robustness and efficiency of pixel tracing and material assignment.

2. Using results from previous layers as initial conditions could enable a more efficient, possibly single-pass, global solver for the entire model.

TECHNION
Israel Institute
of Technology

Thank you for listening