A B-spline based Framework for Volumetric Object Modeling

Fady Massarwi, Gershon Elber

Computer Science Dept. Technion, Israel Institute of Technology, Haifa, Israel

Abstract

With the recent development of Iso-geometric Analysis (IGA) [**?**] and advanced manufacturing technologies employing heterogeneous materials, such as additive manufacturing (AM) of functionally graded material, there is a growing emerging need for a full volumetric representation of 3D objects, that prescribes the interior of the object in addition to its boundaries. In this paper, we propose a volumetric representation (V-rep) for geometric modeling that is based on trimmed B-spline trivariates and introduce its supporting volumetric modeling framework. The framework includes various volumetric model (V-model) construction methods from basic non-singular volumetric primitives to high level constructors, as well as Boolean operations' support for V-models. A V-model is decomposed into and defined by a complex of volumetric cells (V-cells), each of which can also represent a variety of additional varying fields over it, and hence over the entire V-model. With these capabilities, the proposed framework is able of supporting volumetric IGA needs as well as represent and manage heterogeneous materials for AM. Further, this framework is also a seamless extension to existing boundary representations (B-reps) common in all contemporary geometric modeling systems, and allows a simple migration of existing B-rep data, tools and algorithms. Examples of volumetric models constructed using the proposed framework are presented.

Keywords: Trivariate functions, Trimmed volumes, Volumetric Boolean operations, Domain Decomposition, Additive manufacturing, 3D printing, Iso-Geometric analysis.

1. Introduction

In geometric modeling (GM), 3D objects are mainly represented by their boundaries [**?**]. Typically, these boundaries are represented as a set of tensor product (trimmed) surfaces. These surfaces define the 2-manifold boundaries of the object and hence delineate its volume. Until recently, full volumetric representation of 3D objects has not been in high demand in the GM and engineering communities. However, with the development of advanced manufacturing technologies employing heterogeneous materials such as additive manufacturing [**?**] (AM, also known as 3D printing) using functionally graded materials [**?**], methods for representing materials in the entire volume of the object are in active demand and research. The object's description should include its geometry as well as other relevant internal volumetric data sets, like material's properties, such as stresses or conductivity fields, and also boundary conditions, like pressure. Hereafter, we refer to these non-geometric data sets and fields as *attributes*. These volumetric capabilities are simply lacking in contemporary GM systems.

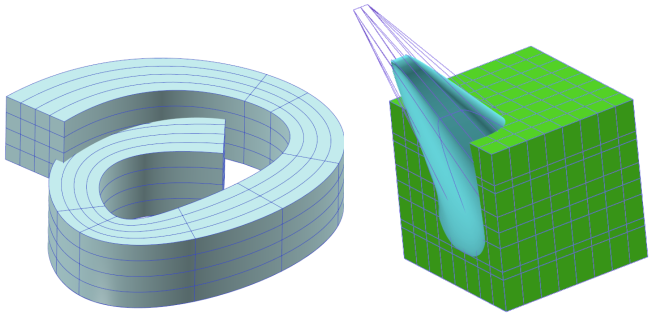
3D volumetric modeling is in demand also in micro-scale bone scaffold design, where patient's specific porous structures are designed, in order to replace unhealthy or diseased tissue [**?**], and in the design of porous material in general (i.e. topological optimizations [**?**]).

Traditional finite elements analysis (FEA) processes require the conversion of 3D B-rep data, such as mechanical parts, to a representation in which the physical simulation can be employed. Grids or meshes, based on piecewise linear approximating primitives like triangles, tetrahedra, quadrilaterals and hexahedra are frequently used representations for analysis [**?**]. The drawback of such a discrete representation is the lack of numerical stability on one hand, and accuracy on the other - the generated models are merely piecewise linear discrete approximations of the real freeform models [**?**]. [**?**] reports that the process of generating a discrete approximated mesh for analysis from a given 3D B-rep CAD model is the key, most time consuming, step in finite element analysis (FEA). It is estimated to consume about 80% of the overall design and analysis process in the automotive, aerospace and ship industry. Thus, employing a single geometric representation for both the design and the analysis, throughout the modeling cycle, has major potential advantages.

Isogeometric analysis (IGA) postulates the use of the same tensor product B-spline [**?**] representation used for representing the geometry in the physical analysis as well, striving for a tighter bond between GM and Analysis. The end user can work in the same B-spline representation in the design and analysis stages, avoiding both the need to generate a finite element intermediate mesh toward the analysis and the need for a complex feedback of the analysis results back to the B-spline based geometric model. However, it is necessary to have a B-spline based representation that is suitable for analysis. Over the last decade, several studies have been conducted to utilize IGA for different physical analysis problems, i.e. [**? ? ? ?**], showing improved accuracy and robustness over traditional finite element methods [**?**]. So far, these studies are, for the most part, designed

for the 2D case, (and to some extent trimmed surfaces), due to the lack of suitable geometrical representation and tools for handling volumetric objects.

In this work, we introduce a framework and techniques for representing and managing freeform volumetric objects, having a full representation of the boundary as well as the interior volume of the model, taking into account both accuracy and efficiency. In this work, we only deal with volumetric models (V-model) that are open sets and regular. That is, a V-model is a regular 3-manifold geometry and the boundary (closure) of the V-model is a closed regular 2-manifold. A regular 3-manifold (2-manifold) doesn't self-intersect, and have a vanishing Jacobian in no place. See Figure 1 for examples of a non-regular (singular) trivariate. As stated, in this work, we exclude such singular cases.



(*a*) (*b*)

Figure 1: (a) shows a non-regular (singular) self-intersecting B-spline trivariate whereas (b) shows a non-regular cuboid B-spline trivariate where three internal control points are moved up and outside the cuboid, resulting in a zone of negative Jacobian (in cyan). Note the boundary of this trivariate no longer consists of only the six boundary surfaces. See also [**?**].

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 also more advanced constructors such as ruled volumes and volumes of revolution.
3. Algorithms for Boolean operations over V-models, that are based on trimmed B-spline trivariate vector functions.
4. Geometric support tools for precise management and analysis.
5. Almost seamless conversion from B-rep geometry.

The rest of this document is organized as following. Section 2 discusses related work. In Section 3, we describe our V-model representation (V-rep). In Section 4, we present the volumetric Boolean operations between V-models, for creating arbitrarily complex V-models. In Section 5, we portray unique geometric tools that are supported in the framework and enable precise analysis over V-models. In Section 6, we present some results of constructions of V-models using the proposed V-rep modeling framework,and finally, in Section 7, we conclude and discuss future planned research.

2. Related Work

The most common geometric representation of 3D objects is the boundary representation (B-rep), where the object is delineated by its boundary surface(s) . There are many methods to model the boundary surface(s), and the two most common basic building blocks for B-reps are:

- Linear primitives such as triangles, quads or general polygons, and
- Trimmed spline surfaces, as a set of piecewise polynomial or rational functions, over some parametric spaces.

These methods are commonly used in contemporary GM systems for about half a century, with little change. Until recently, representing the inner volume of the object has been of little interest. However, in recent years, there has been a growing need for modeling the interior volume of geometric objects, for example in engineering, medicine and manufacturing. AM with graded materials is already a proven technology and IGA requires an appropriate representation for various prescribed and/or computed fields in the interior of an object.

The voxel based volume representation is common in medical applications and is simple to manipulate. However, voxel based approaches suffer from lack of accuracy and huge data sizes. For example, a CT scan from a typical device can generate a volume of 512 voxels in each dimension (a 2mm accuracy for an object of one meter wide), with each voxel represented in typically 16 bit. Such a volume requires around 256 MBytes of storage space. In contrast, the accuracy offered by current subtractive manufacturing (SM) technologies (i.e. CNC) is in the orders of microns and tens of microns, which is 2-3 orders of magnitude better, and AM is expected to follow these accuracies in the near future. There is little hope that voxel based representation can provide such accuracies, considering the amount of expected storage space.

There have been several relevant studies on spline based volumetric representations. Basic constructors of tensor product B-spline trivariates has been investigated in [**?**]. In [**?**], algorithms to derive the boundaries of tensor product trivariates with singularities (vanishing and/or varying-sign Jacobians) are proposed. Aigner et al.[**?**] proposed an algorithm for calculating a tensor product trivariate from boundary conditions and guiding curves. Their method is proposed to handle only swept volume structures. Liu et al. [**?**] uses Boolean operation of volumetric cylinders and cubes with hierarchical octrees to extract T-spline trivariates from boundary triangulated surfaces. However, [**?**] cannot handle more complex shapes, even as simple as cones and tetrahedron.

Kumar, et al. [**? ? ?**] provide a good summary on the various mathematical representations for models found in the literature. They proposed a framework for segmenting the object into cells by using constructive solid geometry (CSG) operations, where each cell describes both the geometry and material properties. However, their method uses a limited set of basic shapes (and CSG operations) like spheres and cubes, which makes their method too restrictive to handle complex general freeform objects.

Martin et al. [**?** **?**] modeled the attributes as separate trivariate volumes that are not coupled to the geometry, but share the same parametric domain. Their method supports only complete (non-trimmed) tensor product trivariates.

Biswas et al. [**?**] proposes a high level abstract model for the representation of heterogeneous objects, that are composed of geometry and continuously varying materials. Their extension toward heterogeneous materials is based on a distance function from interior or boundary geometric curves or points defining the material attributes, called features positions. Chen et al. [**?**] proposes a framework for representing and optimizing volumetric heterogeneous models. They represent the geometry of the model by implicit functions and the material distribution using a linear combination of B-spline basis functions. Both representations are defined over the same spatial domain, where only locations inside the model are considered.

None of the above studies handle trimmed volumes, and are not general enough to support the current modeling space of B-reps, common in all contemporary geometric modeling systems. To the best of our knowledge, there is no known general method for handling Boolean operations over volumes. Much like traditional 2-manifold B-rep modeling, where trimmed surfaces make the space of objects one can model much richer compared to tensor product surfaces, we similarly expect that the proposed trimmed volume representation will offer a much richer modeling space than tensor product B-spline trivariates. Modern geometric modeling systems offer a fairly small set of surface constructors, such as basic surface primitives (i.e. a cone), ruled surfaces and surfaces of revolution, as well as Boolean operations over these B-rep elements. We believe that B-rep models designed on a contemporary CAD system that is based the trimmed B-spline surface representation are also precisely representable as V-models in the proposed framework. More so, it is going to be a relatively simple task to perform these B-rep to V-rep conversions, as we will show, in this work, a parallel set of trivariate constructors as well as an ability to perform the Boolean operations directly in V-rep.

3. The volumetric modeling framework

The proposed framework supports methods and algorithms for constructing volumetric models (V-models) in a similar way to B-rep, allowing for migration with ease of B-rep algorithms, tools and data. Furthermore, the framework is designed to support anticipated AM and IGA needs, which means it should support different queries on V-rep models such as slicing (intersection with a plane), (point) inclusion, geometrical neighborhood information and contacts, local refinements and more. In Section 3.1, we describe the proposed data structures that represent a V-model, and in Section 3.2, we introduce methods and algorithms for constructing V-models.

3.1. The V-model representation

A V-model is a complex of several volumetric cells (V-cells), where each V-cell is a volumetric cell represented by a trimmed B-spline trivariate:

Definition 3.1. *A* B-spline trivariate *is a volumetric extension to parametric B-spline curves and surfaces, in a three dimensional parametric space [**?**]. A common representation of trivariates is by tensor product B-splines, as:*

$$F(u,v,w) = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} P_{i,j,k}B_{i,d_u}(u)B_{j,d_v}(v)B_{k,d_w}(w), \quad (1)$$

where F is defined over the 3D parametric domain $[U_{min}, U_{max})\times[V_{min}, V_{max})\times[W_{min}, W_{max})$, and where $P_{i,j,k} \in \mathbb{R}^q$, $q \geq 3$ are the control points of F and $B_{i,d}$ is the i'th univariate B-spline basis functions of degree d.

Note $P_{i,j,k} \in \mathbb{R}^q$, $q \geq 3$ where the first three coordinates always represent the geometry but optionally also additional attributes, such as a color or a stress tensor field, for $q \geq 3$. The scalar components of the different attribute fields can be easily encoded into the $q-3$ coefficients of the $P_{i,j,k}$'s, the control points of $F(u,v,w)$. An alternative, yet a bit more general approach, would be to define the attribute fields as additional trivariates alongside $F(u,v,w)$ while sharing the same parametric domain.

The B-spline trivariate is our basic building block that defines a volume. However, it is limited to a cuboid topology, and can't represent general volumetric shapes. For this, we define a *V-rep cell* (A V-cell) as a trimmed B-spline trivariate volumetric cell. In fact, a V-cell can also be contained in several trimmed B-spline trivariates, but then, all trivariates must contain all the volume in the V-cell. The trimming of a V-cell is prescribed by a set of trimming (bivariate) B-spline surfaces in the domain of the trivariate(s). Each such trimming surface is, in turn, possibly trimmed by trimming B-spline curves. Hence:

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.*

A V-cell defines a unique volumetric zone inside a V-model and, by definition, the V-cell is fully contained in all its tensor product trivariates. Then, the V-model is defined as following:

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.*

In addition to its geometry and attributes, the V-model holds topological information about its 3-/2-/1-/0-manifold elements. Such information is important for traversing and updating the inner structures of the V-model, for example, in domain decomposition [**?**] toward analysis, where boundary conditions are propagated between V-cells. Each V-cell can have several V-cell neighbors, where every pair of adjacent V-cells typically share one or more boundary trimming surfaces (a *V-surface*):

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.*

Hence, we define the *boundary* of a V-model as following:

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

The topological information structure we define can be seen as a 3-manifold extension to the known half-edge data structure for representing regular graphs [**?**]. Edges and faces are elevated a dimension here and extended to V-surfaces and V-cells, respectively, while we also manage the topology of the 2-manifold boundary B-rep as before. In a similar way to the half-edge semantics, where shared edges between faces are split into two half-edges, we split each V-surface into two *half-V-surfaces*:

Definition 3.6. *A* half-V-surface *is a boundary (trimmed) surface of a single V-cell. Each internal V-surface is split into two half-V-surfaces with opposite orientation (having negated normal directions at the same point), while a boundary V-surface is associated with only one half-V-surface. The half-V-surface holds references to:*

1. *Its V-cell and the B-spline trivariate this half-V-surface is a boundary/trimming surface of.*
2. *The other half-V-surface of the neighboring V-cell (if exists).*
3. *Trimming loops in the parametric space of the surface.*

Figure 2 shows an example of two V-cells sharing a V-surface, that is split into two half-V-surfaces. Each half-V-surface refers to its V-cell and also points to the other half-V-surface.

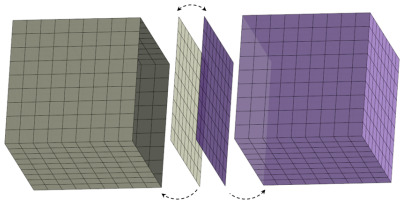


Figure 2: A shared V-surface boundary between two V-cells is split into two half-V-surfaces. Each half-V-surface references its V-cell and also references the adjacent half-V-surface.

The topology of boundary curves in 3-manifolds is more complex than the topology of edges defined in the original 2-manifold half-edge data structure. This complexity stems from the fact that herein each boundary curve can belong to more than two V-cells. In order to maintain the topological information through boundary curves we define another component in the V-model:

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 (see Definition 3.8 below).*

Adjacent V-cells can share common faces (V-surfaces), common curves (V-curves), and common points. To maintain topological information through curves' end points, we define our last topological component:

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.*

We now define a *gluing operation* that is used to construct the topology of a V-model from a given set of V-cells.

Definition 3.9. *A* gluing *operation on a set of V-cells, $\mathcal{V}_C^i$, creates a new V-model, V_M, that contains all the V-cells in $\mathcal{V}_C^i$ with proper topological adjacency information on its V-surfaces, V-curves and V-points.*

The gluing construction procedure of the topology is further discussed and explained in Section 3.1.1.

Figure 3 shows a V-model composed of three V-cells, A, B and C. This V-rep is generated by a union operation of two V-rep box models, O_1 and O_2. The B V-cell is the common volume between the two input V-models. V-cells A and C hold one trivariate each from the original V-models O_1 and O_2, while V-cell B holds both trivariates. Each V-cell has 6 trimming half-V-surfaces. There are two internal V-surfaces and 14 boundary ones. These V-surfaces are converted together to 18 half-V-surfaces. The V-surface between A and B is split into two half-V-surfaces and the same holds for the V-surface between B and C. The V-rep of this model also contains 16 V-points (in red), and 28 V-curves (in yellow). The V-curve labeled by 1 holds information about four V-surfaces (split into six half-V-surfaces) attached to it, and two V-points at its end points. While the V-curve labeled 2 holds information about two V-surfaces (split into two half-V-surfaces). The V-points of V-curve 1 are attached to five V-curves each, and those of V-curve 2 are attached to three V-curves each.

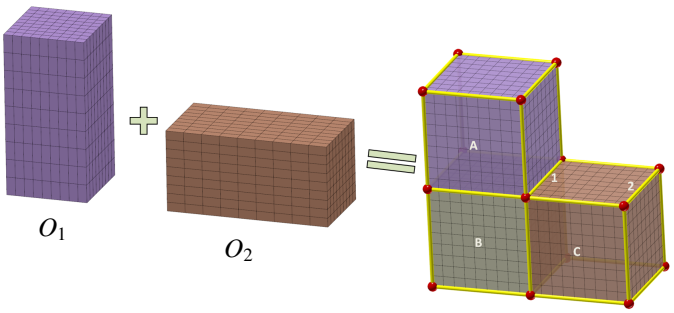


Figure 3: A V-model generated by uniting two box V-models, O_1 and O_2. The result V-model contains three V-cells, A, B and C. V-points are marked in red, and V-curves are displayed in yellow.

With the above data structures, topological queries can be implemented efficiently. For example, these queries include:

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?

There are several advantages in using the proposed V-rep:

1. The V-model is a complex of V-cells, where each V-cell can have a different independent geometry and set of attributes. Such a decomposition can make it convenient to define and control heterogeneity.
2. The V-model uses B-spline trivariates as basic volumetric building blocks, which makes it suitable, for example, for both AM and IGA, offering accuracy and light size.
3. The V-model maintains topological information between different parts of the model. Such information is useful, for example, for efficient local updates, and in domain decomposition analysis [**?**], where boundary conditions and constraints between V-cells, over V-surfaces and in inner structures are imposed.
4. The presented V-rep offers a seamless and precise migration path from B-rep (modeling tools, algorithms and data alike). This topic will be further discussed in Section 5.4.
5. The presented V-rep supports precise analysis, including: integration of trimmed surfaces using untrimming, contact and penetration depth analysis and precise point/curve projections. These topics will be further discussed in Section 5.

3.1.1. Managing the topology

As part of constructing the topological information of a V-model, neighboring V-cells are *glued* together by comparing V-surfaces, V-curves and V-points for similarity. Only the geometry (and no attribute values) is taken into account when these similarities are examined and we now briefly explain how these entities are compared:

- Two V-points are considered the same if they designate the same 3-space location in Euclidean space.
- Two V-curve are considered the same if their B-spline curves are the same. Following [**?**], two B-spline curves are the same if their control points are the same after reparametrization(s) (via compositions), if any, were eliminated and they were elevated to the same function space via degree raising and knot insertion.
  In this work, we only bring the curves to a common function space before any comparison of control points takes place. However, and because curves can be parameterized in reverse, the curves are also compared for similarity after one of the curves is reversed.
- Two half-V-surfaces are considered the same if their B-spline surfaces (and their trimming curves, compared as the above V-curves) are the same. Two B-spline surfaces are the same if their control points are the same after the elimination of any reparametrization (via a composition), if was one, and they were elevated to the same function space via degree raising and knot insertion.
  In this work, we only bring the surfaces to a common function space before any comparison of control points takes place. However, and because surface $S(u, v)$ can be parameterized by reversing either u or v or both and u and v can be flipped to yield the exact same traced surfaces, the surfaces are also compared following all these reversal/flip operations of one of the surfaces.

3.1.2. Managing the attributes

Attributes and attribute fields can be attached to trivariates and control points of trivariates as discussed in the beginning of Section 3.1. Further, attributes and attribute fields can also be attached to V-cells, V-surfaces, V-curves and V-points, possibly as boundary conditions. Attribute fields can be prescribed to boundaries of the V-model, i.e. as boundary conditions in IGA. Attributes can also be prescribed to boundaries of V-cells, for example, in domain decomposition analysis, as well as the interior of V-cells, setting interior desired properties at certain locations. Each V-cell can encapsulate a set of attributes with certain continuity requirements with its neighbors. Finally, independent constraints can also be prescribed in the interior of the V-model, and possibly manipulated in a similar way to [**?**].

Consider a Boolean operation between two V-models. All the intersecting zones between the two V-models are recreated as new V-cells that inherits the geometry and the attributes from *both models*. Consider, for example, the union of a cone O_1 and a cylinder O_2 (see Figure 4 (a)). The resulting V-model contains three categories of V-cells:

1. V-cells in O_1 that are not in O_2 (A in purple, in Figure 4 (b)). These V-cells inherit attribute information from O_1 only.
2. V-cells in O_2 that are not in O_1 (C in green, Figure 4 (b)). These V-cells inherit attribute information from O_2 only.
3. V-cells in the intersecting zone of O_1 and O_2 (B in cyan, Figure 4 (b)). These V-cells inherit attribute information from both O_1 and O_2.

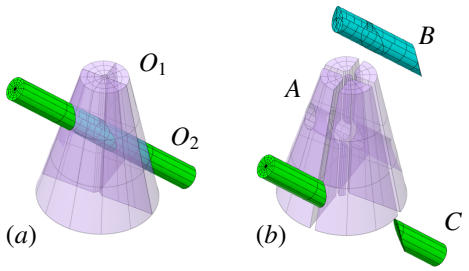


Figure 4: A union between a cone V-model and a cylinder V-model is shown in (a). The result contains several V-cells shown partially exploded in (b).

A V-cell can inherit the same attributes from different trivariates, during the Booleans, and hence, some blending scheme of these attributes must be defined. The V-model in Figure 5 shows one such simple blending scheme. This V-model is generated as the union of four V-models, each V-model has a different color attribute. The intersection volumes between these two V-models are represented as a separate, new, V-cell each (They are also translated up in the figure, in the Z axis, for clarity of the display). Each V-cell inherits two color attributes, from the two original models. The blending scheme used here simply sums up the the colors (clipped to maximally allowed intensity). This summation function introduces a C^{-1} discontinuity in the attributes along the V-surfaces shared between these V-cells.

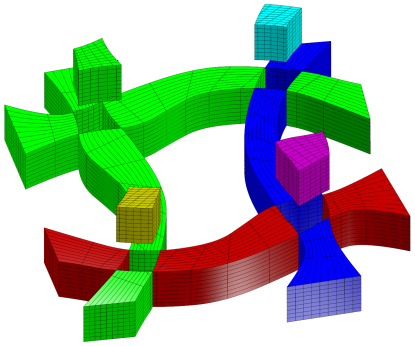


Figure 5: A Union between four V-models. The blending scheme used here simply sums the color attribute in the intersection volumes.

Many different blending schemes can be used. As an additional example, a different blending function, seeking a better continuity of the attribute fields, might consider the distance from the boundary V-surfaces of the V-cell, as in [**?**]. Let $V_M = V_M^1 \cup V_M^2$, be some V-model that is the result of a Boolean union operation between V-models V_M^1 and V_M^2. For simplicity, we assume each of V_M^1 and V_M^2 contains one V-cell only. Let p be a point in V_M, and let $d(p, O)$ denote the minimal distance from p to O, d is a C^0 continuous function. Let $A_1(p)$ and $A_2(p)$ be some C^0 attribute fields of models V_M^1 and V_M^2, respectively, at p, and finally, let $d_{i2}^{b1}(p) = d(p, \partial V_M^1 \cap V_M^2)$ denote the distance from the boundary of V_M^1, ∂V_M^1, that is inside V_M^2 from point p.

Then, the attribute value of model V_M, $A(p)$, at p can be calculated as the blend of $A_1(p)$ and $A_2(p)$ as follows (see Figure 6):

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

$A(p)$ (Equation (2)) ensures C^0 continuity of attribute values everywhere in V_M as it involves only C^0 continuous functions, except at singular locations. Point r, in Figure 6, where both $d_{i2}^{b1}(r)$ and $d_{i1}^{b2}(r)$ vanish simultaneously, is one such singular location. Further, having C^k continuous $A_1(p)$ and $A_2(p)$ and defining higher order versions of Equation (2), by employing $\left(d_{i1}^{b2}(p)\right)^k$ as blending function, $k > 1$, better continuity in the attribute fields can be gained at regular locations.

3.2. Basic construction methods of V-model

The presented V-rep framework supports several construction methods of V-models. The different methods are described in Sections 3.2.1, 3.2.2 and 3.2.3.

3.2.1. Trivariate B-splines

Any tensor product B-spline trivariate can be converted to a V-model having, typically, one V-cell, six boundary half-V-surfaces, twelve V-curves, and eight V-points, all in a cuboid topology. The trivariate can be explicitly prescribed, by providing its three, (u, v, w) orders, its control mesh and its three knot sequences. Alternatively, a trivariate can be prescribed using high level constructors, as described in Section 3.2.2 and 3.2.3.

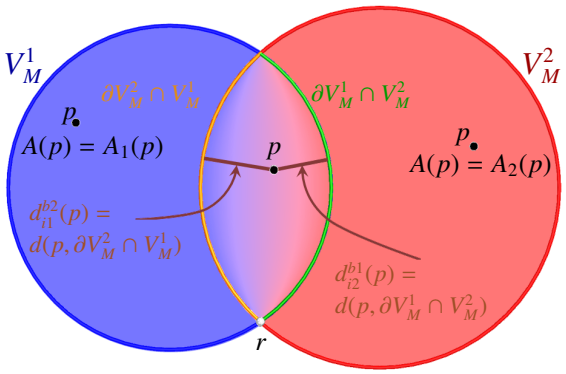


Figure 6: Attribute blending of two V-models V_M^1 and V_M^2. The red color of V_M^1 and the blue color of V_M^2 are blended using the blending scheme proposed in Equation (2).

3.2.2. High level V-rep constructors

Bivariate constructors such as extrusion, ruled surface, surface of revolution and sweep surface can be adopted to the trivariate case [**?**]. The presented framework supports the following constructors:

1. Extruded volume: Given a surface $S(u, v)$ and a vector V, an extruded trivariate $T(u, v, w)$ is defined as: $T(u, v, w) = S(u, v) + Vw$, $w \in [0, 1]$.
2. Ruled volume: Given two surfaces $S_1(u, v)$ and $S_2(u, v)$, a ruled trivariate $T(u, v, w)$ is defined as: $T(u, v, w) = S_1(u, v)(1 - w) + S_2(u, v)w$, $w \in [0, 1]$.
3. Volume of revolution: Given a surface $S(u, v)$, a trivariate of revolution $T(u, v, w)$ is constructed by rotating S around some axis.
4. Boolean Sum: Given six surfaces organized in a cuboid topology, a trivariate T is generated such that each such surface is a boundary of T. For more details see [**?**].
5. Volumetric sweep: A trivariate T is generated that interpolates or approximates a given ordered list of surfaces, $S_i(u, v)$, at different w_i parameters, $w_i \in [0, 1]$.

Figure 7 shows examples of trivariates generated using all the above constructors.

3.2.3. Non singular primitives

Primitive models such as spheres, cylinders, tori and cones can't be represented as a tensor product B-spline trivariate without introducing singularities, where the normal (and the Jacobian of the mapping) vanishes. Figure 8 shows examples of a B-spline trivariate cylinder and a trivariate sphere that are singular along their central axis.

For analysis, singularities in the parametric domain are not desired. We offer a set of different constructors of primitive V-models that are composed of non-singular V-cells. Each V-cell is non-singular in its domain. Note that at glued V-surface locations the primitives are only C^0 continuous. The cylinder, cone and the torus models are composed each of five V-cells, as displayed in Figure 9 (a)-(c), while the sphere model (Figure 9 (d)) is composed of seven V-cells: an internal cube V-cell and six rounded V-cells around the cube, and each one of these

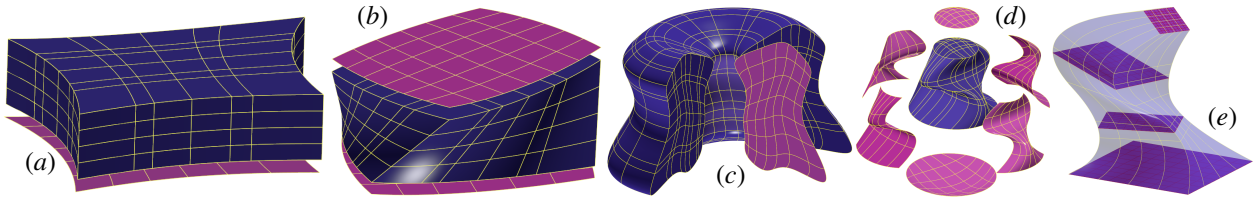


Figure 7: Constructors of trivariates: (a) A volume of extrusion (b) A ruled volume (c) Volume of revolution (d) Volumetric Boolean sum (e) A sweep volume. All these constructors yield a single B-spline trivariate.

V-cells is constructed as a ruled volume between a cube face and a sixth portion of a tensor product B-spline sphere surface, as described in [**?**]. Each V-cell's trivariate of these V-rep primitives is constructed using high level trivariate constructors, as discussed in Section 3.2.2. Then, these V-cells are glued together to form one V-model primitive.

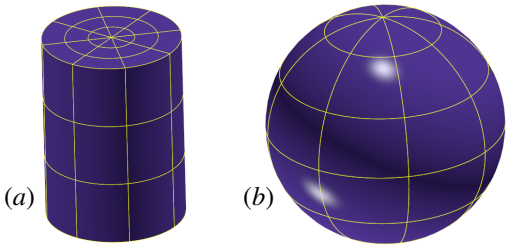


Figure 8: (a) A singular B-spline trivariate cylinder. (b) A singular B-spline trivariate sphere.

4. Boolean operations over V-models

The V-model describes both the boundary and interior structures of an object, as a complex of (mutually exclusive) V-cells. Boolean operations, such as union, intersection and subtraction, between V-models, should take into account the interior structure as well as the boundary surfaces. The Boolean operation is performed between the V-cells in the V-models, and the result is a set of new V-cells that are glued together into a new V-model.

Each V-cell is a closed 3-manifold that is enclosed in a 2-manifold boundary (V-surfaces). A Boolean operation algorithm on V-models can exploit algorithms of B-rep Boolean operations between the boundary V-surfaces of the V-cells, only to reconstruct the interior information as a second stage. Boolean operations between B-rep models is a well studied problem [**? ? ?**], and can generally be conducted as follows. Given two B-rep models M_1 and M_2, with their boundaries represented as sets of trimmed surfaces:

1. Compute all the intersection curves between the (trimmed) surfaces of M_1 and M_2. The intersection curves of the two surfaces are given in their parametric space.

   If there is no intersection between the surfaces of M_1 and M_2, then either they are disjoint or one contains the other. To delineate between these cases, an inclusion test is performed, for example, by tracing a ray from a point on one model in an arbitrary direction, and counting the number of intersections between the ray and the other model. The models are disjoint if the number of intersection is even, and contained otherwise.
2. For each (trimmed) surface S_i^k, $i = 1, 2$ in model M_i, split the trimming curves of S_i^k by intersecting them with the new intersection curves computed in Step 1, if any.

   If no intersection curves in (trimmed) surface S_i^k, move S_i^k to Step 5
3. For each intersecting (trimmed) surface from Step 2, use the new intersection curves computed in Step 1 and the splitted old trimming curves computed in Step 2, to create new trimming loops, based on the specific Boolean operation.
4. Classify each trimming loop (new and old) as in/outside according to the orientation and the specific Boolean operation. Loops that are not in the domain are purged.
5. Classify the rest of the non-intersecting (trimmed) surfaces from Step 2 as completely in/outside by boundary adjacency propagation from the intersecting (trimmed) surfaces.
6. Unite all the classified-as-inside (trimmed) surfaces and glue them topologically to define the final result. Note the classified-as-inside operation depends on the specific Boolean operation.

Before we present the Boolean operations over V-models, we need some common language. Let $\mathcal{S}_{V_C}(V_M)$ denote the entire set of V-cells of V-model V_M. Then, for each V-cell we have:

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 $\mathcal{S}_{\mathcal{TV}}(V_C)$ denote the set of all trivariates of V-cell V_C.*

Then, to build a V-cell, we have the following constructor:

Definition 4.2. *The* V-cell constructor, *$C_{V_C}(\mathcal{S}, \mathcal{T})$, creates a new V-cell having $\mathcal{S}$ as its set of trimming half-V-surfaces, and $\mathcal{T}$ as its set of trivariates.*

In other words, $C_{V_C}(\mathcal{S}, \mathcal{T})$ groups a given trimmed surfaces ($\mathcal{S}$) and trivariates ($\mathcal{T}$) into one V-cell. Figure 10 shows an example of constructing one of the two V-cells that are shared between the teapot's handle and body. The V-cell is also shown in Figure 14. The constructor of this V-cell, $C_{V_C}(\mathcal{S}, \mathcal{T})$, receives the two trivariates of the teapot's handle and the teapot's body that share it (as set $\mathcal{T}$), and its boundary V-surfaces that are trimmed surfaces resulting from the Boolean operation between the two trivariates' boundary surfaces (as set $\mathcal{S}$).

Being a well studied problem, we aim here to exploit the abilities to compute B-rep Boolean operations for freeforms as much as possible. As briefly explained at the beginning of

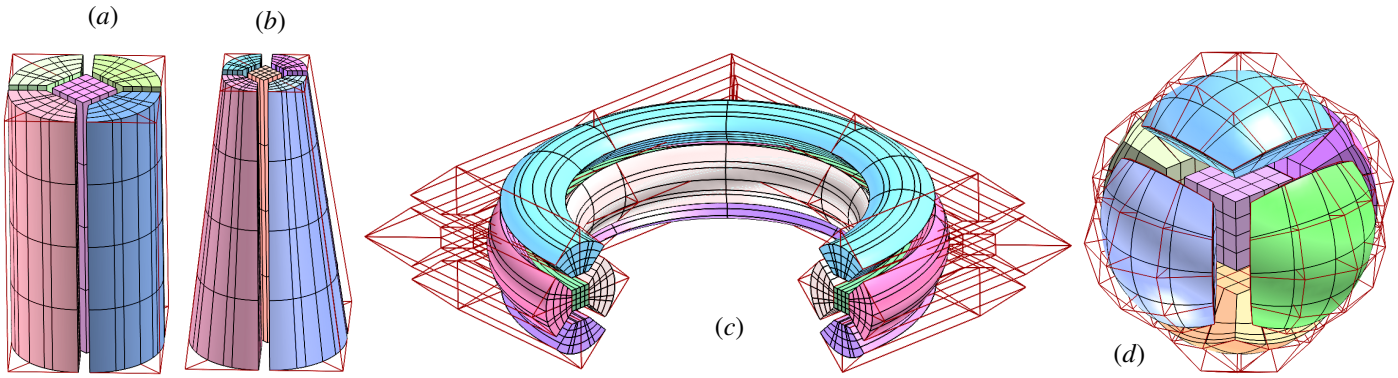


Figure 9: Non-singular primitive V-models, shown is an exploded view and exposing their inner structure, with the control meshes of the trivariates. (a) A cylinder composed of five trivariates constructed by extrusions. (b) A cone composed of five trivariates constructed as ruled volumes. (c) A torus composed of five trivariates constructed as volumes of revolution. (d) A sphere composed of seven trivariates constructed by ruled volumes. Each trivariate in these primitives is made of one V-cell, and all V-cells in a primitive are glued together along shared V-surface boundaries to form a single V-model primitive.

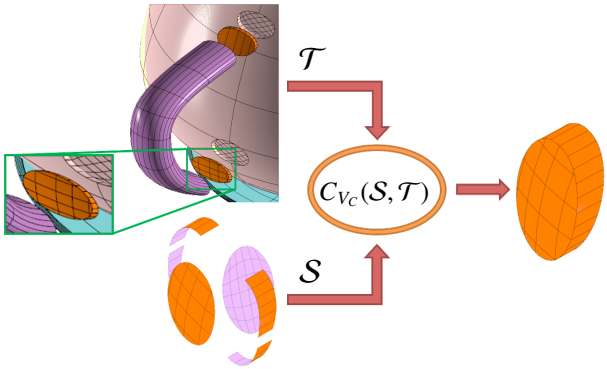


Figure 10: The construction of a V-cell of the bottom shared volume between the teapot's handle and body trivariates (marked and zoomed in, in green). See also Figure 14.

this section, B-rep Boolean operations over freeforms typically takes as input two sets of (trimmed) surfaces (and additional information, including topological), as the boundary of the two input objects and returns a set of (trimmed) surfaces as the result (and additional information, including topological).

Both the B-rep Booleans and the expected V-rep Booleans are in $\mathbb{R}^3$ and hence the result of a B-rep Booleans can be used to delineate the boundaries of the result V-rep model. Given two V-models, V_M^1 and V_M^2, and their respective boundaries, ∂V_M^1 and ∂V_M^2, their B-rep Boolean operation will be denoted by *BrepBoolOP*(∂V_M^1, ∂V_M^2, *BoolOP*), where *BoolOp* can be either INT (intersection) or SUBTR (Subtraction). Similarly *BrepBoolOP*(∂V_C^1, ∂V_C^2, *BoolOP*) will denote the B-rep Boolean operation over the boundaries of two V-cells.

Finally, in the ensuing discussion we assume V_M^1 and V_M^2 share a common boundary ($\partial V_M^1 \cap \partial V_M^2 \neq \emptyset$), for otherwise, either V_M^1 and V_M^2 are disjoint or one is enclosed by the other, cases that can easily handled via the application of inclusion tests. We also make the same assumption when dealing with V-cells and we are now ready to discuss Boolean operations over V-reps:

- **Intersection:** Since V_M^i, $i = 1, 2$ is a complex of V-cells, the result of a Boolean intersection operation between V_M^1 and V_M^2 is the union of V-cells that results from Boolean intersection operations over all possible pairs of V-cells in V_M^1 and V_M^2.

  The intersection of two V-cells, V_C^i and V_C^j, amounts to the computation of the boundary of the V-cell of the intersection, using the *BrepBoolOP* B-rep ability. However, we also carry on the relevant trivariates in this V-cell, which are the union of the trivariate sets in V_C^i and V_C^j. This, so we could manage the attributes in the new intersection V-cell (i. e. Section 3.1.2). See also Algorithm 1

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:

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

- **Subtraction:** The subtraction of V_M^2 from V_M^1 can also be done one V-cell at a time, subtract all V-cells $V_C^j \in V_M^2$ from every V-cell $V_C^i \in V_M^1$, $\forall i$. However, it is also possible to subtract V_M^2 once from $V_C^i \in V_M^1$, an alternative that is simpler and probably more robust. See also Algorithm 2
- **Union:** The union of V_M^2 and V_M^1 can exploit Algorithms 1 and 2. The union of V_M^2 and V_M^1 consists of all the V-cells in V_M^{1i2}, V_M^{1s2}, and V_M^{2s1}, or all the V-cells in the intersection of V_M^2 and V_M^1, the subtraction of V_M^2 from V_M^1, and the subtraction of V_M^1 from V_M^2. In a similar way to the refinement process in [**?**], the union operation divides the set of V-cells of V_M^1 and V_M^2 into a set of mutually exclusive V-cells. See also Algorithms 3.

5. Precise analysis of V-rep

Since IGA [**?**] introduced the use of spline-based analysis, efficient geometric modeling methods and tools were sought in

Algorithm 2 V-Subtract: Subtraction of two V-models

Input: V-models V_M^1, V_M^2;
Output: V_M^{1s2}, V-model of the subtracting V_M^2 from V_M^1;
Algorithm:

1: $\mathcal{V}_{Cells} := \emptyset$;
2: $\mathcal{B}_2 := \partial V_M^2$;
3: **for all** $V_C^i \in \mathcal{S}_{V_C}(V_M^1)$ **do**
4: $\quad \mathcal{B}_i$:=BrepBoolOP(∂V_C^i, $\mathcal{B}_2$, SUBTR);
5: $\quad \mathcal{V}_{Cells} := \mathcal{V}_{Cells} \cup \{C_{V_C}(\mathcal{B}_i, \mathcal{S}_{\mathcal{TV}}(V_C^i)\}$;
6: **end for**
7: V_M^{1s2} := Glue all V-cells in $\mathcal{V}_{Cells}$;

Algorithm 3 V-Union: Union of two V-models

Input: V-models V_M^1, V_M^2;
Output: V_M^{1u2}: V-model of the union of V_M^1 and V_M^2;
Algorithm:

1: V_M^{1i2} := V-Intersect(V_M^1, V_M^2);
2: V_M^{1s2} := V-Subtract(V_M^1, V_M^2);
3: V_M^{2s1} := V-Subtract(V_M^2, V_M^1);
4: $\mathcal{V}_{Cells} := \mathcal{S}_{V_C}(V_M^{1i2}) \cup \mathcal{S}_{V_C}(V_M^{1s2}) \cup \mathcal{S}_{V_C}(V_M^{2s1})$;
5: V_M^{1u2} := Glue all V-cells in $\mathcal{V}_{Cells}$;

order to perform the analysis more robustly and precisely. V-models are a complex of V-cells where each V-cell has a set of trimmed surfaces as its boundary (V-surfaces). Trimmed surfaces are a powerful tool for representing general freeform surfaces, and are typically defined as tensor product surfaces with a set of trimming curves. Because trimmed surfaces can posses complex arrangements of trimming curves, it is difficult to compute even their (reasonably tight) bounding box, not to say perform operations such as integration and contact analysis over them. Having surfaces as an integral part of the V-rep framework, tools to accurately handle (trimmed) surfaces and trivariates are a must. In Section 5.1, we present a method for accurate integration over trimmed surfaces using an operator we denote *untrimming*, a method that can also be extended to trimmed trivariates. In Section 5.2, a precise and robust method for computing the orthogonal projection of points and curves on surfaces (boundaries of V-models) is introduced. In Section 5.3, we present a tool for deriving the precise contact and penetration depth, toward precise analysis of contact related problems. Finally, in Section 5.4, we briefly discuss the possible migration of B-rep data to V-rep.

5.1. Untrimming of trimmed surfaces

Consider a trimmed surface $S_t(u, v)$ and let D_{S_t} be its trimming domain. While the direct integration over D_{S_t} is a challenging task, $S_t(u, v)$ can be precisely mapped to a set of tensor product patches, via the following process:

1. Tile D_{S_t} by mutually exclusive quads, Q^i, $i = 1, ..., k$, that can have freeform boundaries (including the trimming curves that delineate D_{S_t}), as B-spline curves.
2. Parameterize the interior of each quad Q^i using the four curves bounding the quad, for example using Boolean Sum, as $Q^i(r, t)$.
3. For each $Q^i(r, t) = (u^i(r, t), v^i(r, t))$, construct a tensor product surface $S_{Q^i} = S_t(Q^i(r, t)) = S_t(u^i(r, t), v^i(r, t))$, using surface-surface composition [**?**].

The set $\{S_{Q^i}\}, i = 1, ..., k$ precisely tiles and covers the original trimmed surface $S_t(u, v)$ but consists of only tensor product patches. Hence, a tight bounding box for S_t can be derived by computing the bounding box of the set $\{S_{Q^i}\}$. More importantly, the integration over the original trimmed surface $S_t(u, v)$ can be reduced to integration over the set of tensor product patches $\{S_{Q^i}\}, i = 1, ..., k$, in this process we denote *untrimming*. See Figure 11 for one example.

Note that S_t better be a Bézier surface as $Q^i(r, t)$ cannot cross knot lines (or otherwise $Q^i(r, t)$ must be divided along the knot line and likely to no longer be a topological quad and hence require further refinement into several smaller quads...). However, one can always subdivide trimmed surface $S_t(u, v)$ into Bézier patches before this untrimming process is applied.

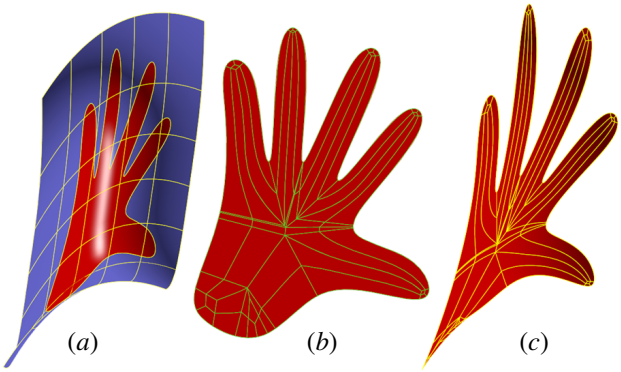


Figure 11: The conversion of a trimmed surface into a set of tensor product patches via *untrimming*. (a) A trimmed surface in the shape of a hand. (b) The domain of the trimmed surface is tiled with quads with freeform boundaries. (c) The composed tensor product tiles cover the original trimmed surfaces.

5.2. Points and curves projection

Let $C(t)$ be a parametric curve, and $S(u, v)$ be a C^1 regular B-spline surface. Orthogonally projecting $C(t)$ into $S(u, v)$ results in a curve(s) lying on S. If $C(t)$ is orthogonally projected to $S(u, v)$, then the normal of S at (u, v) pierces $C(t)$. Hence, we can algebraically prescribe the (univariate) solution of finding the orthogonal projection of $C(t)$ on $S(u, v)$ as:

$$\begin{aligned} \langle C(t) - S(u, v), \frac{\partial S(u, v)}{\partial u} \rangle &= 0, \\ \langle C(t) - S(u, v), \frac{\partial S(u, v)}{\partial v} \rangle &= 0, \end{aligned} \tag{3}$$

having two equations and three unknowns (u, v, t). The solution of these polynomial constraints are univariates in the (u, v, t) parametric space. One can clearly evaluate S at the (u, v) solution locations and project the solution to the Euclidean space. See Figure 12 for one example. The constraints in Equation (3) are simultaneously solved with the aid of solver [**?**], typically in a fraction of a second. Finally, one should note that the orthogonal projecting of a point P on surface $S(u, v)$ is a special

simpler case, where $C(t)$ is substituted by P in Equation (3), and Equation (3) is reduced to two constraints in two unknowns (u, v).

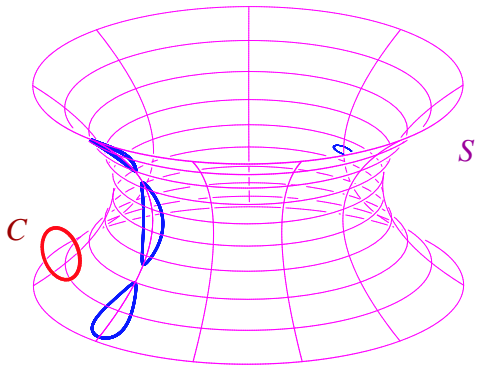


Figure 12: The orthogonal projection of a circular curve C (in red) on surface S, results in four disjoint curve segments as four solution loops (shown on the surface in blue).

5.3. Contact and maximal penetration depth analysis

In general, handling contacts in analysis is a major challenge. Herein, we seek to robustly handle contacts in IGAe [**?**]. The Analysis of contacts begins with the detection of the intersection between the two boundaries of the objects which are typically bounding surfaces. The intersection could be curve(s) or single points, in singular, first contact, cases. For two C^1 continuous regular surfaces $S_1(u, v)$ and $S_2(r, t)$, finding the extreme penetration of one surface into another (or the minimum distance between two disjoint surfaces) implies seeking the extreme values of $\|S_1(u, v) - S_2(r, t)\|$, which is divided into several cases:

1. Extrema can be found on the boundaries, in which case curve-curve and curve-surface extrema should be sought.
2. Extrema can occur at C^1 discontinuities that are isoparametric curves for tensor product surfaces, and hence reduces to curve-curve and curve-surface extrema as well.
3. The most complex case is of interior extrema, that reduces to solving four equations with four unknowns. By differentiating

   $$\|S_1(u, v) - S_2(r, t)\|^2 = \langle S_1(u, v) - S_2(r, t), S_1(u, v) - S_2(r, t) \rangle$$

   with respect to the four degrees of freedom (u, v, r, t), we get:

   $$\begin{aligned} \langle S_1(u, v) - S_2(r, t), \frac{\partial S_1(u, v)}{\partial u} \rangle &= 0, \\ \langle S_1(u, v) - S_2(r, t), \frac{\partial S_1(u, v)}{\partial v} \rangle &= 0, \\ \langle S_1(u, v) - S_2(r, t), \frac{\partial S_2(r, t)}{\partial r} \rangle &= 0, \\ \langle S_1(u, v) - S_2(r, t), \frac{\partial S_2(r, t)}{\partial t} \rangle &= 0. \end{aligned} \tag{4}$$

   The constraints in Equation (4) are, again, simultaneously solved to yield the extrema locations with the aid of solver [**?**], typically in a fraction of a second.

5.4. Migration from B-rep

The common paradigm of constructive solid geometry (aka CSG) is frequently used in B-rep modeling. A CSG tree of some B-rep freeform model can be mapped almost directly to a CSG tree of a V-rep model. We have shown that all primitives and all basic surface constructors can similarly create volumetric trivariates. Moreover, we have shown that Boolean operations over B-rep can also be extended to handle V-reps.

Given a B-rep model in some geometric modeling system, its history is known, and hence so its CSG tree. The equivalent construction of a V-rep model means mapping all leaves and all internal CSG operations from B-rep to V-rep. Since we showed that, for the most part, such mappings exist, we believe that the migration of B-rep data to the proposed V-rep should be fairly simple and almost transparent to the end user that will be performing the same operations as before. As one additional example, Figure 13 presents the conversion of a swept surface to V-rep, only to show one additional possible use of V-reps in constructing porous geometry, using surface-trivariate compositions.

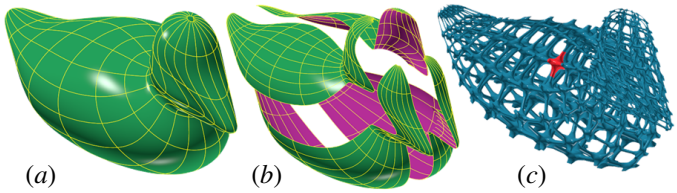


Figure 13: A B-spline swept surface in the shape of a duck in (a), is automatically converted to a V-rep using volumetric Boolean Sum by dividing the surface into four strips along the sweep and adding two cap surfaces at the beginning and end of the sweep, defining the six surfaces needed for the volumetric Boolean Sum. In (c), surface-trivariate composition is employed to tile the constructed trivariate duck with porous elements, as another possible application of V-reps, of construction of porous geometry. Note one tile is highlighted in red, in (c).

6. Examples and Results

The presented volumetric modeling framework is implemented and integrated into the IRIT solid modeler [**?**]. We use the IRIT solid modeler to perform the B-rep Boolean operations on the boundary surfaces and to generate trivariate by the constructors described in Section 3.2. The polynomial constraints described in Sections 5.2 and 5.3 are solved using the IRIT's multivariate polynomial solver [**? ? ?**].

To demonstrate the capabilities of the proposed V-rep modeling, several fairly complex V-models have been created (see Figures 14-17). Each example displays one V-model, in which each V-cell is colored differently. In order to better display the inner V-cells of the V-model, the V-models are also displayed in "exploded view", where the V-cells are translated in different directions.

In some of the examples, the trivariates in the V-cells were created using the primitive constructors that were presented in Section 3.2.3 while in other cases, B-spline trivariate constructors such as extruded volumes, ruled volumes, swept volumes, volumes of revolution and Boolean sum were used (See Section 3.2.2). Then, Boolean operations between these basic V-models were applied.

Table 1 presents some statistics on the V-models constructed using the above introduced abilities. The number of V-cells and the sizes of the trivariates that were used to construct the V-model are provided. Being the most demanding task, the running times of the Boolean operation algorithm are displayed as well, divided into the gluing time of the different V-cells and the total Boolean operation time (including the gluing). These models were constructed on Macbook-Pro i7 2.7Ghz machine, running Window 7 64 bit.

7. Conclusions and Future work

In this paper, we have introduced a possible framework for modeling volumetric objects . The framework presents a volumetric representation (V-rep) that efficiently and accurately supports the representation of the interior as well as the boundary of any general 3D object. Further, this V-rep is capable of precisely representing geometry as well as other attribute fields over the model. We have introduced a variety of construction methods for V-models, including non-singular volumetric primitive objects. An algorithm for Boolean operations between V-models is also presented that in all supports the same modeling space of that of freeform B-reps. Finally, we have introduced several examples of tools and algorithms for accurate processing and analysis of V-reps, as part of this volumetric framework.

The proposed framework is designed to enable smooth migration and transitions from B-rep to V-rep. We believe that any B-rep model constructed in a modern CAD system can be converted to a V-rep model. That said, the limitations over V-rep Booleans are similar to those in B-rep ones. Herein, we assumed valid (regular) input V-models and further, their intersections are not tangential (yielding non 3-manifold topology). Both restrictions can be removed if so desired, at extra efforts of supporting non 3-manifold topologies.

Proper methods that adhere to certain continuity requirements of the geometry and the different attribute fields across V-cells are needed and are under investigation. Blending method of attributes that guarantees more than C^0 continuity are to be considered. The presented gluing operation of V-cells can be improved to achieve a continuity higher than C^0, probably by having constraints imposed over near boundary control points. A tight link for analysis (and other applications) to update as well as query these non-geometrical attribute fields must be developed and offered as part of this framework.

The presented volumetric primitives in Section 3.2.3 are only C^0 continuous. The continuity across V-cells in the V-model primitives should be further investigated, devising a configuration that guarantees a C^k or a G^k, $k > 0$ continuity along the common V-surfaces' boundaries between these V-cells.

The result of Boolean operations between V-models might contain very small V-cells (i.e. see Figure 16 (b)), which are undesired in analysis, among others. Improving the V-rep algorithms to avoid or handle such small V-cells is an open problem.

In order to fully support a seamless migration from B-rep to V-rep, some additional development of advanced tools and operators for this volumetric modeling framework is required, operators such as volumetric fillets and blends.

In order to adapt different algorithms suitable for tensor product B-spline surfaces to trivariates in V-models, algorithms for untrimming of V-cells must be implemented. This reduces to the need to tile the untrimmed, valid, domain of the V-cell by a set of mutually exclusive cuboids, over which trivariates will be mapped using composition, much like the way we presented for trimmed surfaces in Section 5.1. However, we expect the untrimming to be only a computational tool whereas the real V-model will continue to consist as described in the proposed data structure (a complex of V-cells).

T-splines [**?**] (and similar alternative representations) were introduced as more compact alternative to tensor product B-splines and conceptually can similarly be extended to V-rep as trimmed trivariate T-splines. This possible extension should also be investigated.

The GM challenges we are facing today are completely different than a decade ago. The recent IGA developments and AM advancements are not the only foreseen changes. Specialized multi-material volumetric geometry, such as mixed compounds (i.e. rubber-graphite), and composite materials are additional GM challenges where geometric modeling software is behind industrial needs. Representations and algorithms to remedy this situation must be sought.

References

[] Aigner, M., Heinrich, C., Jüttler, B., Pilgerstorfer, E., Simeon, B., Vuong, A.V., 2009. Swept volume parameterization for isogeometric analysis. Springer.

[] Bartoň, M., Elber, G., Hanniel, I., 2011. Topologically guaranteed univariate solutions of underconstrained polynomial systems via no-loop and single-component tests. Computer-Aided Design 43, 1035–1044.

[] Beer, G., Marussig, B., Zechner, J., Dünser, C., Fries, T., 2014. Boundary element analysis with trimmed NURBS and a generalized IGA approach. CoRR abs/1406.3499. URL: `http://arxiv.org/abs/1406.3499`.

[] Bendsoe, M., Sigmund, O., 2013. Topology Optimization: Theory, Methods, and Applications. Springer Berlin Heidelberg. URL: `https://books.google.co.il/books?id=ZCjsCAAAQBAJ`.

[] Biswas, A., Fenves, S.J., Shapiro, V., Sriram, R., 2008. Representation of heterogeneous material properties in the core product model. Engineering with Computers 24, 43–58.

[] Cazier, D., Dufourd, J.F., 1997. Reliable boolean operations on polyhedral solids thanks to a 3d refinement .

[] Chen, J., Shapiro, V., 2008. Optimization of continuous heterogeneous models, in: Heterogeneous objects modelling and applications. Springer, pp. 193–213.

[] Cobb, J.E., 1988. Tiling the sphere with rational bézier patches, in: Report TR UUCS-88-009. University of Utah, USA.

[] Cohen, E., Riesenfeld, R., Elber, G., 2002. Geometric modeling with splines: An introduction. 2002. AK Peters Natick, MA, USA .

[] Cottrell, J.A., Hughes, T.J., Bazilevs, Y., 2009. Isogeometric analysis: toward integration of CAD and FEA. John Wiley & Sons.

[] DeRose, T., Goldman, R., Hagen, H., Mann, S., 1993. Functional composition via blossoming. ACM Transactions on Graphics 12, 113–135.

[] Elber, G., 2010. Irit 10 user's manual URL: `http://www.cs.technion.ac.il/~irit`.

[] Elber, G., Kim, M.S., 2001. Geometric constraint solver using multivariate rational spline functions, in: Proceedings of the sixth ACM symposium on Solid modeling and applications, ACM. pp. 1–10.

[] Elber, G., Kim, Y.J., Kim, M.S., 2012. Volumetric boolean sum. Computer Aided Geometric Design 29, 532 – 540. URL: `http://www.sciencedirect.com/science/article/pii/S0167839612000295`, doi:`http://dx.doi.org/10.1016/j.cagd.2012.03.003`. geometric Modeling and Processing 2012.

Model	# V-cells	Trivariate's orders range	Trivariate's sizes range	Gluing Time (Secs.)	Total Time (Secs.)
Teapot (Figure 14)	7	[2,2,2] - [4,2,2]	[4,2,2] - [13,13,7]	0.45	35.0
Solid (Figure 15)	7	[2,2,2] - [4,2,2]	[2,2,2] - [4,2,2]	0.08	2.9
Solid (Figure 16)	12	[2,2,2] - [4,2,2]	[2,2,2] - [4,2,2]	0.98	17.6
Solid (Figure 17)	19	[2,2,2] - [5,5,2]	[2,2,2] - [5,5,2]	0.24	6.2

Table 1: Statistics on the process of creating the V-models presented in Figures 14-17.

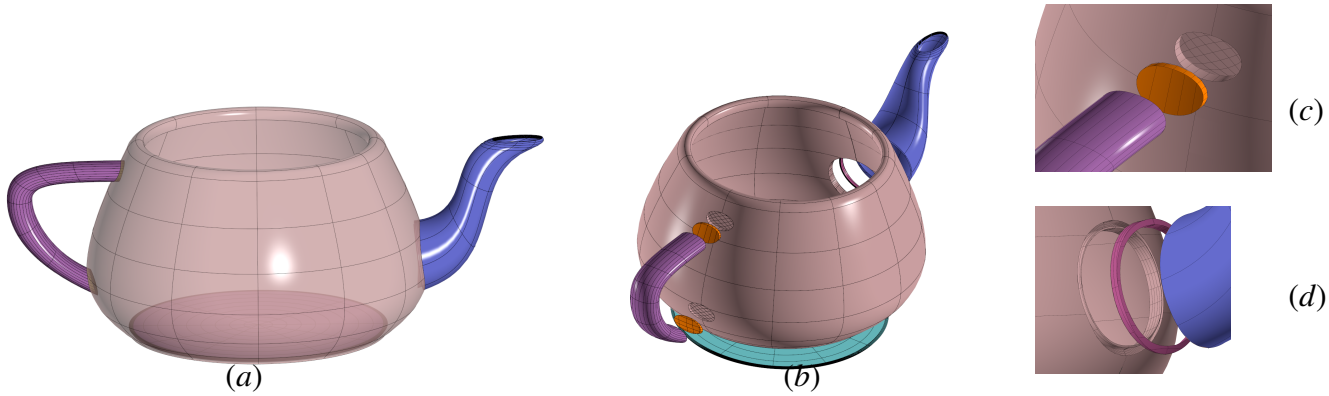


Figure 14: Utah teapot V-model. The V-model is created using three union operations and one subtraction. The Handle is a volumetric Boolean Sum version of the B-rep surface. The Body (and Spout) are ruled volumes between the original Body (Spout) surface and a small offset surface. The complete teapot V-model is shown in (a), consisting of 7 V-cells shown in (b) in an exploded view. A zoom in into the area near the handle is shown in (c), and near the spout in (d). Recall also Figure 10.

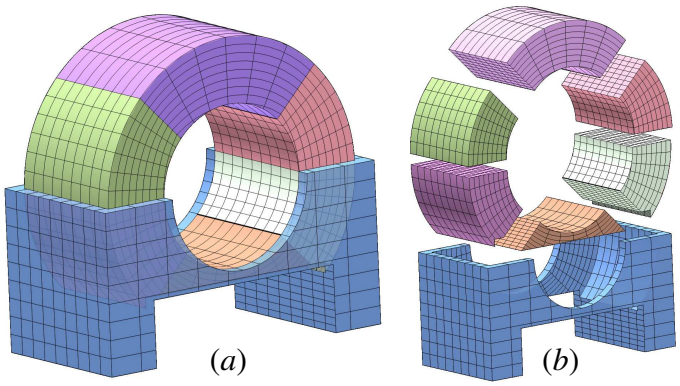


Figure 15: A simple solid V-model. The V-model is created using one union operations, and two subtractions. All the used primitives are non-singular. (a) The V-model consists of 7 V-cells (b) The V-model in an exploded view.

[] Evans, J.A., Bazilevs, Y., Babuska, I., Hughes, T.J.R., 2009. n-widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. ACM Transactions on Graphics 198, 1726–1741.

[] Hanniel, I., Elber, G., 2007. Subdivision termination criteria in subdivision multivariate solvers using dual hyperplanes representations. Computer-Aided Design 39, 369–378.

[] Hassani, B., Tavakkoli, S., Moghadam, N., 2011. Application of isogeometric analysis in structural shape optimization. Scientia Iranica 18, 846 – 852. URL: http://www.sciencedirect.com/science/article/pii/S1026309811001283, doi:http://dx.doi.org/10.1016/j.scient.2011.07.014.

[] Hughes, T.J., 2012. The finite element method: linear static and dynamic finite element analysis. Courier Corporation.

[] Joy, K., Duchaineau, M.A., 1999. Boundry determination for trivariate solids, in: Proceedings of the 1999 Pacific Graphics Conference, pp. 82–91.

[] Kumar, V., Burns, D., Dutta, D., Hoffmann, C., 1999. A framework for object modeling. Computer-Aided Design 31, 541–556.

[] Kumar, V., Dutta, D., 1997. An approach to modeling multi-material objects, in: Proceedings of the fourth ACM symposium on Solid modeling and applications, ACM. pp. 336–345.

[] Kumar, V., Dutta, D., 1998. An approach to modeling & representation of heterogeneous objects. Journal of Mechanical Design 120, 659–667.

[] Liu, L., Zhang, Y., Hughes, T.J., Scott, M.A., Sederberg, T.W., 2014. Volumetric t-spline construction using boolean operations. Engineering with Computers 30, 425–439.

[] Lorenzis, L.D., Wriggers, P., Hughes, T.J.R., 2014. Isogeometric contact: a review. GAMM Mitteilungen 37, 85–123.

[] Lu, J., Zheng, C., 2014. Dynamic cloth simulation by isogeometric analysis. Computer Methods in Applied Mechanics and Engineering 268, 475 – 493. URL: http://www.sciencedirect.com/science/article/pii/S0045782513002430, doi:http://dx.doi.org/10.1016/j.cma.2013.09.016.

[] Martin, T., Cohen, E., Kirby, R., 2009. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. Computer Aided Geometric Design 26, 648 – 664. URL: http://www.sciencedirect.com/science/article/pii/S0167839608000939, doi:http://dx.doi.org/10.1016/j.cagd.2008.09.008. solid and Physical Modeling 2008ACM Symposium on Solid and Physical Modeling and Applications.

[] Martin, W., Cohen, E., 2001. Representation and extraction of volumetric attributes using trivariate splines: A mathematical framework, in: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications, ACM, New York, NY, USA. pp. 234–240. URL: http://doi.acm.org/10.1145/376957.376984, doi:10.1145/376957.376984.

[] McGuire, M., 2000. The half-edge data structure. Website: http://www.flipcode. com/articles/article halfedgepf. shtml .

[] Paik, K.L., 1992. Trivariate B-splines. MSc. Department of Computer Science, University of Utah. URL: https://books.google.co.il/books?id=GCecNwAACAAJ.

[] Pekerman, D., Seong, J.K., Elber, G., Kim, M.S., 2005. Are two curves the same? Computer-Aided Design and Applications 2, 85–94. URL: http://dx.doi.org/10.1080/16864360.2005.10738356, doi:10.1080/16864360.2005.10738356, arXiv:http://dx.doi.org/10.1080/16864360.2005.10738356.

[] Printer, D., 2010. 3dprinter URL: http://www.3dprinter.net.

[] Satoh, T., Chiyokura, H., 1991. Boolean operations on sets using surface data, in: Proceedings of the First ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications, ACM, New York, NY,

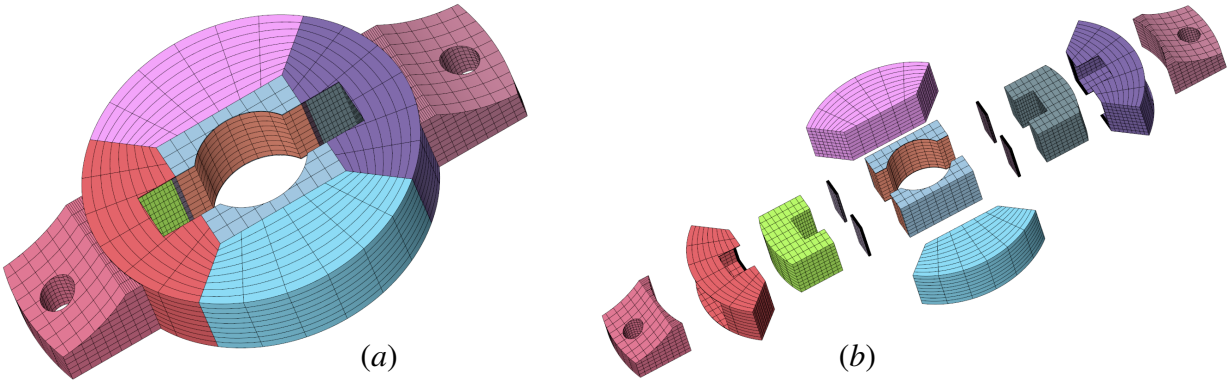


Figure 16: A solid V-model. The V-model is created using two union operations, five subtraction and one intersection. All the used primitives are non-singular. (a) The solid V-model consists of 12 V-cells. (b) The same V-model in an exploded view.

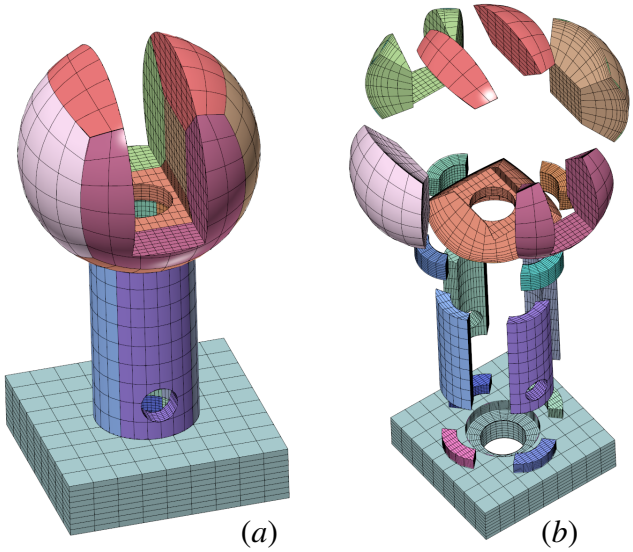


Figure 17: A solid V-model. The V-model is created using two union operations, and three subtractions. All the used primitives are non-singular. (a) The V-model consists of 19 V-cells. (b) The V-model in an exploded view.

USA. pp. 119–126. URL: http://doi.acm.org/10.1145/112515.112536, doi:10.1145/112515.112536.

[] Schroeder, C., Regli, W.C., Shokoufandeh, A., Sun, W., 2005. Computer-aided design of porous artifacts. Computer-Aided Design 37, 339–353.

[] Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A., 2003. T-splines and t-nurccs, in: ACM SIGGRAPH 2003 Papers, ACM, New York, NY, USA. pp. 477–484.

[] Seo, Y.D., Kim, H.J., Youn, S.K., 2010. Isogeometric topology optimization using trimmed spline surfaces. Computer Methods in Applied Mechanics and Engineering 199, 3270 – 3296. URL: http://www.sciencedirect.com/science/article/pii/S0045782510002008, doi:http://dx.doi.org/10.1016/j.cma.2010.06.033.

[] Suresh, S., 2001. Graded materials for resistance to contact deformation and damage. Science 292, 2447–2451.

[] Thomas, S.W., 1986. Set Operations on Sculptured Solids. Technical report, University of Utah, Department of Computer Science. URL: https://books.google.co.il/books?id=9IFzGwAACAAJ.

[] Toselli, A., Widlund, O.B., 2004. Domain Decomposition Methods :Algorithms and Theory. volume 34 of *Computational Mathematics*. Springer Verlag.