www.cadanda.com Computer-Aided Design AND APPLICATIONS

A Hybrid Hierarchical Procedure for Composing Trivariate NURBS Solids

Oluwole T. Morgan[1], Ganesh Subbarayan[2] and David C. Anderson[3]

[1]Purdue University, omorgan@purdue.edu
[2]Purdue University, ganeshs@purdue.edu
[3]Purdue University, dave@purdue.edu

ABSTRACT

With the emergence of novel NURBS-based iso-geometric modeling methodologies, particularly the compositional approach of the Hierarchical Partition of Unity Field Composition (HPFC), there is a need to compose trivariate NURBS solids and quickly perform classifications of subdomains of the resulting composition. An algorithm and accompanying data structure utilizing a hierarchical subdivision of the solids using bounding constructs at different levels of refinements is presented. An overlap-box-sweep algorithm is introduced for computing the actual intersection points. The result of the composition is a hybrid representation containing the original trivariate NURBS solids, a list of oriented intersection segments, a hierarchy of decomposed patches, the partial triangulations and associated hierarchy of bounding constructs. For the purposes of this discussion, we assume the bounding regions of the trivariate NURBS solids can be trivially extracted and consider the boundaries of the solids as a collection of non-trimmed outer NURBS patches. Several examples illustrating the developed hybrid scheme are included.

Keywords: composition, isogeometric modeling, trivariate solids, intersection.
DOI: 10.3722/cadaps.2012.215-226

1 INTRODUCTION

With the emergence of novel NURBS-based iso-geometric engineering analysis methodologies, particularly the compositional approach of the Hierarchical Partition of Unity Field Composition [1,2], there is a need to compose trivariate NURBS solids and quickly perform classifications of subdomains of the resulting composition. An algorithm and accompanying hierarchical data structure has been developed to enable such compositions and provide support for the requirements of the analysis procedure that maintains independent parameterization of the composed solids. Consequently, there is a need for generating iso-parametric descriptions for geometric design and engineering analysis.

The Hierarchical Partition of Unity Field Composition is an evolution of the CSG-inspired analysis methodology that was introduced in [1] and allows for the composition of behavioral fields on primitive NURBS entities. This is done through the generation of smooth weight fields to allow for coupling of the primitive fields and ensuring convergence.

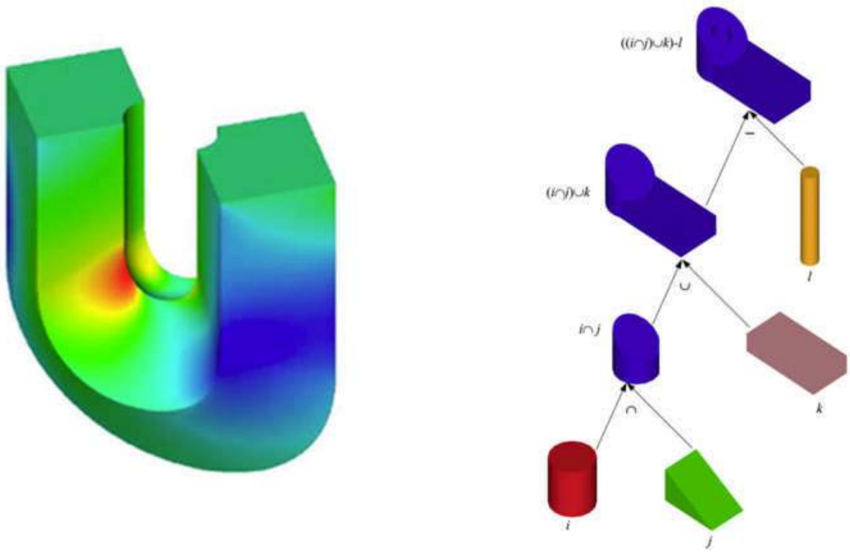

Fig. 1: (a) Isogeometric analysis and (b) the HPFC methodology for performing analysis using both the boolean operators in analysis with primitives represented as NURBS solids. From [1,2].

The reader is pointed to the referenced work [1] for the theoretical aspects that has motivated the development of the algorithm presented here.

A number of approaches exist for constructing trivariate NURBS solids. Although, ab initio NURBS solid construction from specific geometric operations are being developed [3], most are generated from alternate data representation such as tetrahedral meshes [4] and imaging data [5]. The generation methods provide the representations as composite trivariate NURBS solids that are finalized. In this form, any modification to the original models would indiscriminately require that the trivariate NURBS construction or generation procedure be re-initiated. This would be akin to a re-meshing operation with conventional finite element analysis and constitute a relatively large computational overhead. The HPFC computational methodology deals with topology and shape changing events while ensuring minimal re-discretization; an efficient approach would require minimal compositions and avoid complete boundary evaluations leading to static B-Rep models.

The above issues are illustrated with a simple example below. Consider Fig. 2 in which two spherical NURBS solids are composed. The composition of the two solids results in a representation that includes the original NURBS solids, decomposition of the boundary surfaces, a hierarchy of bounding constructs terminating with overlap boxes, localized triangulations and merged intersection line segments.

The hybrid hierarchical procedure involves analytic operations combined with linearized operations. The procedure consists of six levels of operations resulting in an equal number of multi-resolution approximations of the composed trivariate NURBS solids. The procedure includes the following steps:

1. Boundary extraction – extraction of the exterior and non-degenerate boundary surfaces.
2. Bezier decomposition – decomposition of the boundary surfaces based on the original parameterization into Bezier patches.
3. Overlap box construction – bounding boxes for the Bezier patches and overlap boxes
4. Selective tessellation – tessellation of patches within overlap boxes.
5. Kd-tree construction – construction of the kd-tree for fast determination of intersecting triangles.
6. Segment merging – merging of the intersection points computed from the intersections

between triangles and as well as neighboring overlap boxes.

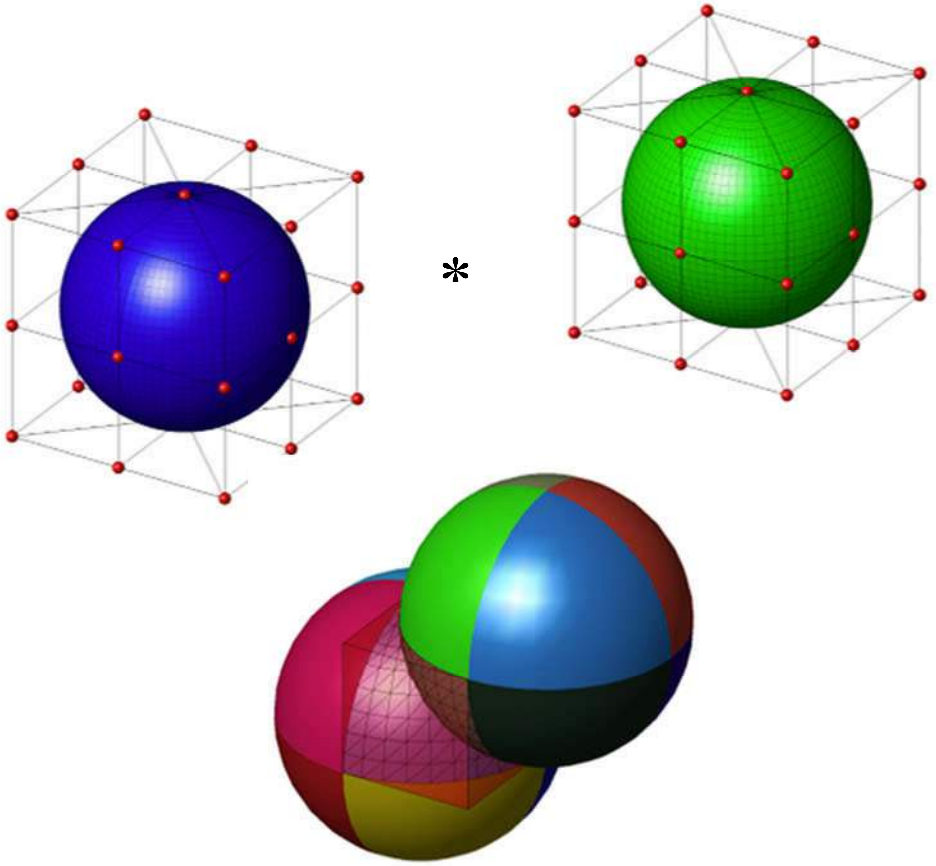

Fig. 2: Composition of two spheres: results in the two NURBS solids, the decomposed boundary surfaces, overlap boxes, partial triangulations and merged intersection lines.

Steps 1-3 are considered exact decomposition and the remaining three steps are based on a linearization.

A fast algorithm and the data structures for composing trivariate NURBS solids with minimal coupling between the two solids are described. The resulting hierarchical data structure enables classification of regions within the composed solid as either belonging to one of the trivariate solids or the intersection of both of the solids. Thereby it is possible to define CSG operations with trivariate solid NURBS as primitives and utilize the resulting model for other processing operations such as volumetric visualization. The algorithm is based on a decomposition of the boundaries of the trivariate NURBS solids using bounding constructs, selective triangulation of regions under the overlap boxes. The data structure stores multiple refinements of the Bezier patches of the bounding surfaces and the triangulation of the surfaces. The connectivity information between the bounding surfaces and the intersection curves provides sufficient information for classification of the composed model. The composition approach described can be extended to ensure closure on composed models; that is, given k trivariate NURBS solids, the algorithm and data structures can be resolved over $k-1$ pair-wise compositions.

Further utility of the resulting data structures (in addition to the domain classification) are the ability:

- to handle incremental modifications to the composition, e.g., analysis driven evolution of the primitives forming the composition.
- to provide starting point for further tessellation of the entire domain for applications such as analysis and visualization.
- to provide geometric structure for enriching the input solid model with additional CAD operations.
- to provide a central data structure for iso-geometric CAD-CAE integration.

2 RELATED WORK

2.1 Surface Intersection Computation

Surface intersection computation has been considered a challenging computational task in computer-aided design and its applications, particularly when the representation of the surfaces are exact representation using rational polynomials such as NURBS. Although, exact analytical results have been shown to exist [6, 7], the problem of degree explosion of the resulting intersection curves and robustness issues are still ongoing areas of exploration [8-11]. Utilizing polyhedral approximations have been known to simplify the intersection problem [12, 13].

2.2 Trimmed NURBS

Trimmed parametric surfaces are surfaces for which valid regions in the parameter space of the parametric surfaces are limited. Trimmed surfaces provide a convenient and efficient representation of surface patches that are not four-sided [16]. The trimmed parametric surface representation is also used in boundary representation of freeform solid models [17]. The boundary representation utilizes the trimmed parametric surfaces consisting of trimming curves and the parametric domains as geometric entities attached to its topological data structure.

Most CAD software and geometric modeling kernels are based on this organization of data through edge-based or vertex-based data structures [18]. The approach of generation of the trimmed surface representation of a boundary representation from Boolean CAD operations is termed boundary evaluation. The surface intersection computation procedure by Manocha et al. serves as an approach of boundary evaluation [9]. Ultimately, the boundary representation is used for both rendering and a basis for additional CAD operations.

Since, the trimming approach utilized is primarily a surface-modeling procedure, it becomes necessary to develop methods that are suitable for a more direct representation of solids. This work attempts to find such a means of providing a suitable representation for trivariate NURBS solids that is sufficient, minimal and provides additional utility for downstream CAD and analysis operations.

3 BOUNDARY EXTRACTION

The composition operation on trivariate NURBS solids is a prerequisite step for Boolean operations that provides sufficient geometric structure for rendering the actual Boolean operations. Obtaining the non-degenerate boundary surfaces from a trivariate NURBS representation is trivially achieved by obtaining the boundary control grid of the representation. This depends primarily on the structure of the trivariate NURBS representation and the expectation that there are no self-intersections.

For a trivariate NURBS solid,

$$P(\xi,\eta,\zeta)=\sum_i\sum_j\sum_k R_i^p(\xi),R_j^q(\eta)R_k^r(\zeta)P_{i,j,k} \tag{3.1}$$

$$R_i(u)=\frac{N_i(u)w_i}{\sum_j N_j(u)w_j} \tag{3.2}$$

with $\xi,\eta,\zeta\in\left[0\ 1\right]$. The trivial boundary is

$$P(\xi_{0,1},\eta_{0,1},\zeta_{0,1})=\left\{\begin{array}{c} P(\xi_0,\eta,\zeta)\\ P(\xi_1,\eta,\zeta)\\ P(\xi,\eta_0,\zeta)\\ P(\xi,\eta_1,\zeta)\\ P(\xi,\eta,\zeta_0)\\ P(\xi,\eta,\zeta_1)\end{array}\right\} \tag{3.3}$$

The non-degenerate exterior boundary for fully closed and partially closed solids is a subset of this collection.

In the case for which degeneracies exists, such as a trivariate representation that contains collapsed surfaces, the surfaces are simply not considered. The degenerate cases are those for which any of the six surfaces in Equation (3.3) collapse. An example would be a trivariate solid sphere defined as shown in Fig. 3.

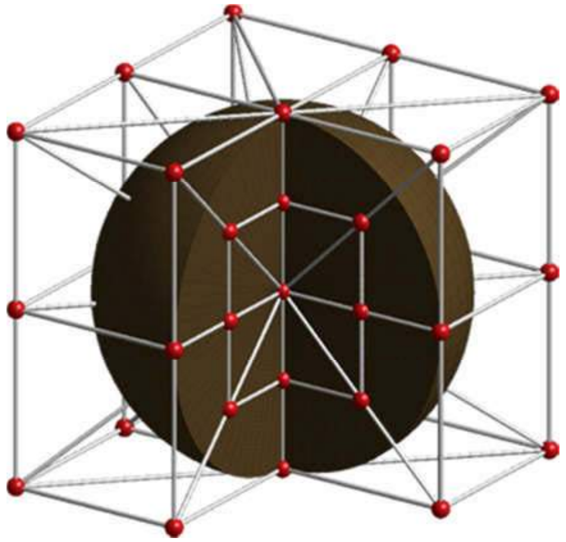

Fig. 3: A trivariate NURBS solid representation of a sphere with collapsed boundaries. The interior section displays the arrangement of the interior control structure.

4 HYBRID HIERARCHICAL REPRESENTATION

4.1 Level 1 – Bezier Decomposition

The next set of representations depends on a decomposition of the extracted boundary into Bezier patches. The decomposition of NURBS surfaces to Bezier patches is done initially using the current parameterization of the surfaces, i.e., the current knot spacing of the NURBS surface. This leads to a natural set of Bezier patches of the NURBS surface. For a NURBS surface with degree p in the u parameter direction and degree q in the v parameter direction, the Bezier patches have $(p+1)(q+1)$ control points, $\{\mathbf{P}_{k,l}\}^{i,j}$. The superscripts denote the indices in the knot intervals containing the Bezier patch, where the range of the indices is between 0 and ends with the maximum number of unique knots.

Each of the Bezier patches lead to two implicit bounding constructs – the convex hull of the control points, and the axis-aligned bounding box.

For the quick determination of intersections, only the bounding boxes are stored along side the Bezier patches in the hierarchical representation.

A secondary decomposition of the Bezier patches is possible by further refinement of the Bezier patches. Although not done explicitly, this refinement can obtained through a series of knot insertions in the and knot vectors of the primary NURBS surface. The refined Bezier patch is defined with the same number of control points and indexing scheme as the Level 1 patch.

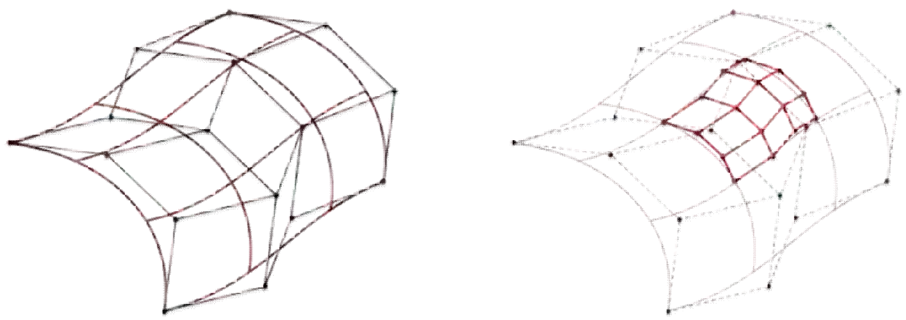

Fig. 4: Extraction of a refined Bezier curve through a series of knot refinements.

4.2 Level 2 – Overlap Boxes

Now, given two hierarchical representations derived from different NURBS solids, a Level 1 composition results in pairs of interacting bounding constructs. The overlap box resulting from the interacting bounding boxes of a pair of Bezier patches from the composed solids is the intersection of the bounding boxes.

A non-empty overlap box and positive intersection of the overlap box with both patches indicates possible interaction region and all further operations are confined to within this region. The portions of the Bezier patches that intersect the overlap box are further decomposed into overlap Bezier patches.

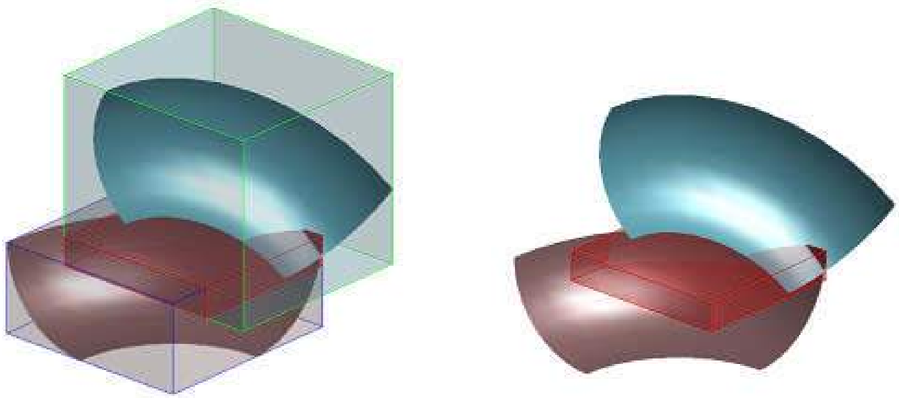

Fig. 5: Overlap box obtained as the intersection of the bounding boxes.

4.3 Level 3 - Tessellation

The subdivided Bezier patches contained within the overlap boxes are triangulated. The triangulation occurs in the parameter domain of the Bezier patches and follows an approach similar to [19]. For the decomposition in the parameter space, a grid of points is computed in the parameter space to match the degree of the original NURBS surface. Additionally, the corner points selected to ensure that the cord lengths of the edges of the quadrilateral partition in the model space are equal. The partitioning or subdivision is repeated until a flatness metric (Equation (4.1)) defined on the quadrilateral $\{\ ,\ ,\ ,\ \}$ satisfies a specified threshold value.

$$= \frac{((\ -\ )\times(\ -\ ).(\ -\ ))}{\|\ -\ \|} \quad (4.1)$$

Finally, the partitions are triangulated and bounding constructs for each of the triangles are stored. The triangles at this level are stored in a lightweight vertex-based data structure that only explicitly stores the vertices and faces. Each face corresponds to a triangle and stores references to a triad of vertices and the vertices store references back to each of the triangles. This structure is sufficient for incremental construction of a triangulation; although the faces are such that they can store additional lists of arbitrary data.

4.4 Level 4– Binary Space Partitioning Tree (Kd-tree)

The representation of the composed field contains at some levels, exact geometric representation as Bezier patches of the extracted boundary of the trivariate solids. Below this level in the hierarchy or tree, localized planar approximations of the surfaces around the regions where intersections with adjacent solids are likely stored. However, to perform the check for actual intersection of the planar approximations the, kd-tree (see Figure 6), is constructed over the vertices of the triangles for one of the triangulations in the pair associated with the overlap boxes in Level 2.

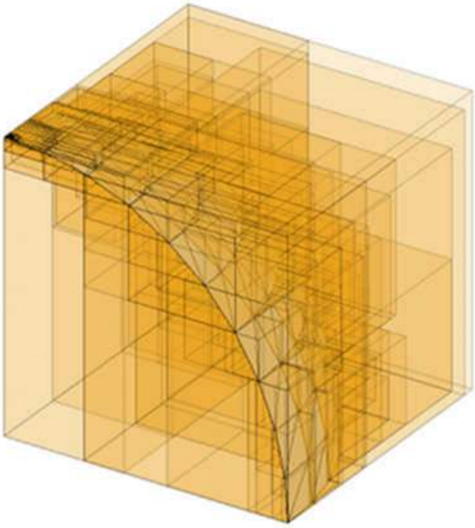

Fig. 6: Kd-tree construction over a triangulation.

4.5 Level 5– Intersection Curves

Associated with the composition is a set of intersection curves determined from the triangulations, both in the real space as well as the parametric domain of the extracted boundaries or localized Bezier segments of the overlap boxes. It is noted that the intersection curves are obtained from merged discrete line segments of the intersecting triangles. The procedure for computing the intersection segments is discussed in the next section.

5 INTERSECTION SEGMENT COMPUTATION AND MERGING

The intersect segment computation procedure initiates with localized triangulations with axis-aligned bounding constructs and a kd-tree defined over the vertices of one of the triangulations as show in Fig. **6**. In this case, we consider the kd-tree construction is over the vertices of the triangulation of the first element of the overlap pair, Level 2 elements. The overlap box is swept [20] and the triangles are queued and processed by testing their bounding boxes against the kd-tree. In the event of a successful hit, all triangles for which the vertex belongs are then tested against the triangle using the bounding boxes and then the triangles themselves. The intersection results in line segments associated with each triangle with a reference to the line segments maintained by the triangles.

The computed line segments also store the intersecting triangles and are merged into bi-direction lists during the intersection computation. Consequently, a list of triangles of the two patches is also merged. The combination of the bidirectional intersection segments, and the sweep allows closed loops to be formed quickly without further processing.

A second level of merging is done over the different overlap regions. The *hanging* intersection segments from each of the neighboring overlap regions are checked and merged appropriately. The merging of the line intersection is terminated when no hanging segments remain. The merge across different overlap regions coincide over bounding regions around intersection triangles since each of the triangulations might not be in perfect alignment. These triangles defined as the *merge* triangles are locally refined and the computed line intersections are merged.

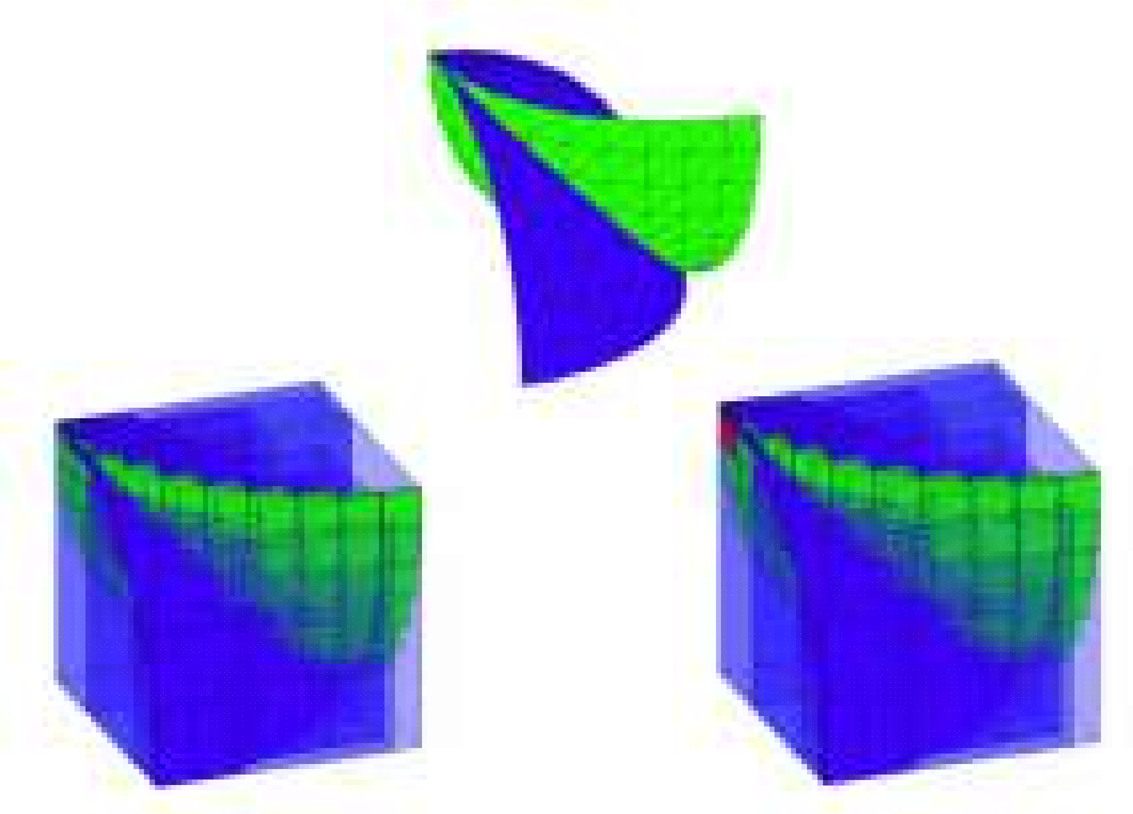

Fig. 7: Fast intersection computation using a kd-tree and an overlap box sweep algorithm.

Algorithm 1 Compose two Trivariate NURBS Solid

Input: $NURBS\_Solid_1, NURBS\_Solid_2$

Output: $\{Hi\_BezierPatches, Hi\_OverlapBoxes, Hi\_Triangulation, Hi\_KDTree, Hi\_XsectSegments\}$

Extract bezier patches and patch bounding boxes
Generate overlap boxes, $Hi\_OverlapBoxes$
for (All overlap boxes) **do**
  Subdivide bezier patches at overlap, $\bar{P}^1_{i,j}$, $\bar{P}^2_{i,j}$
  Tessellate overlap bezier patches, tri_1, tri_2
  for (Each tessellation pair from 1 and 2) **do**
    Construct Hi_KDTree over vertices of 1 and bounding boxes for triangles
    Sweep over tri_2 and search through kdtree
    Find triangles that intersect with tri_2 checking against bounding boxes
    Compute line segments, $Hi\_XsectSegments$
    Merge $Hi\_XsectSegments$ if triangles have been hit
  end for
  Merge $Hi\_XsectSegments$ list from different overlap boxes
end for

6 INTERSECTION CORRECTIONS

The intersection segments computed using the method described gives an approximate representation of the exact intersection curves. An obvious gap exists between the intersection line, the exact surfaces and the actual intersection curve. The gap can be quantified piecewise as the triangular area between a point of the intersection line segments, and the projection of the intersection points onto the exact Bezier patches. By computing the triple plane intersection of the plane containing the original intersection point and the projected points on the two Bezier patches $plane(\mathbf{p}^i_{x\sec t}, \mathbf{B}(u^i, v^i), \mathbf{G}(s^i, t^i))$; and the tangent planes at the project points $plane(\mathbf{B}(u^i, v^i), \mathbf{B}_u(u^i, v^i), \mathbf{B}_v(u^i, v^i))$, $plane(\mathbf{G}(s^i, t^i), \mathbf{G}_u(s^i, t^i), \mathbf{B}_v(s^i, t^i))$, the result $\mathbf{p}^i_{x\sec t}$ can be improved. The computation can be repeated until the distance between the two project points is within a specified tolerance. An efficiency improvement can be made by avoiding multiple projections and instead use an approximation to the Taylor series to determine the step size in the parametric space.

7 EXAMPLES AND DISCUSSION

Using a composition of two trivariate NURBS toroidal solids shown in Fig. 8, the result and final data structure is illustrated. The toroidal solids are defined as triquadratic NURBS with an extraction of a single double closed NURBS surface. Each of the extracted NURBS surfaces are decomposed into 16 bezier segments, with 17 overlap boxes, 26 set of subdivided Bezier segments and triangulations and two computed intersection loops. Figure 7 illustrates composition of two turbine blades.

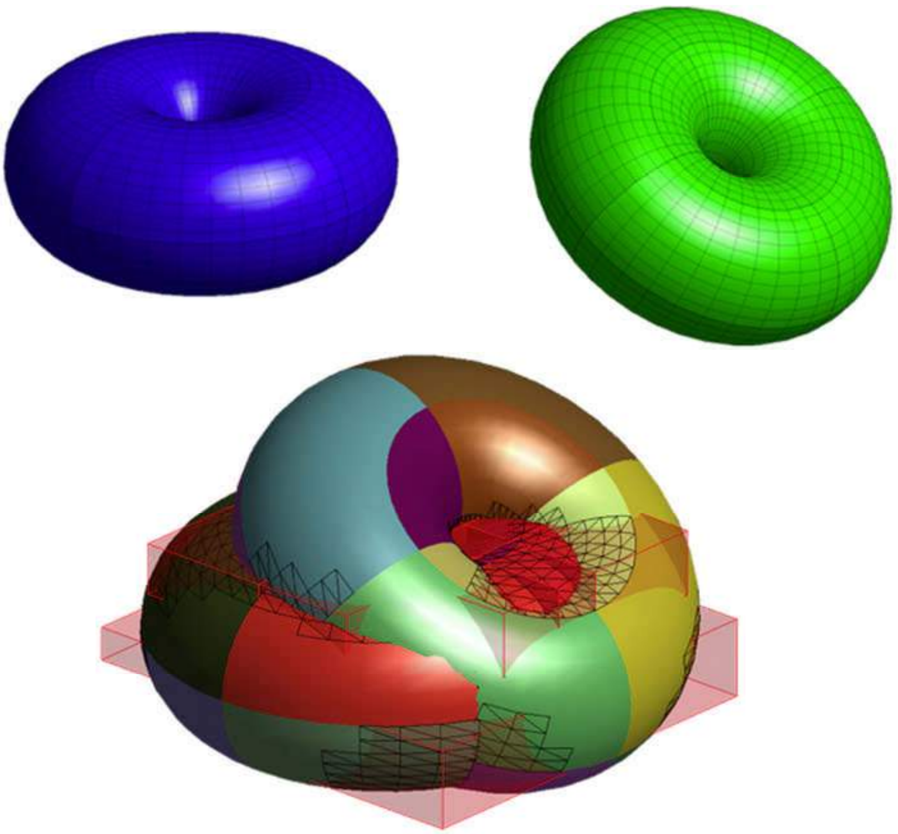

Fig. 8: Composition of two misaligned tori: results in the two NURBS solids, the decomposed boundary surfaces, overlap boxes, partial triangulations and merged intersection lines.

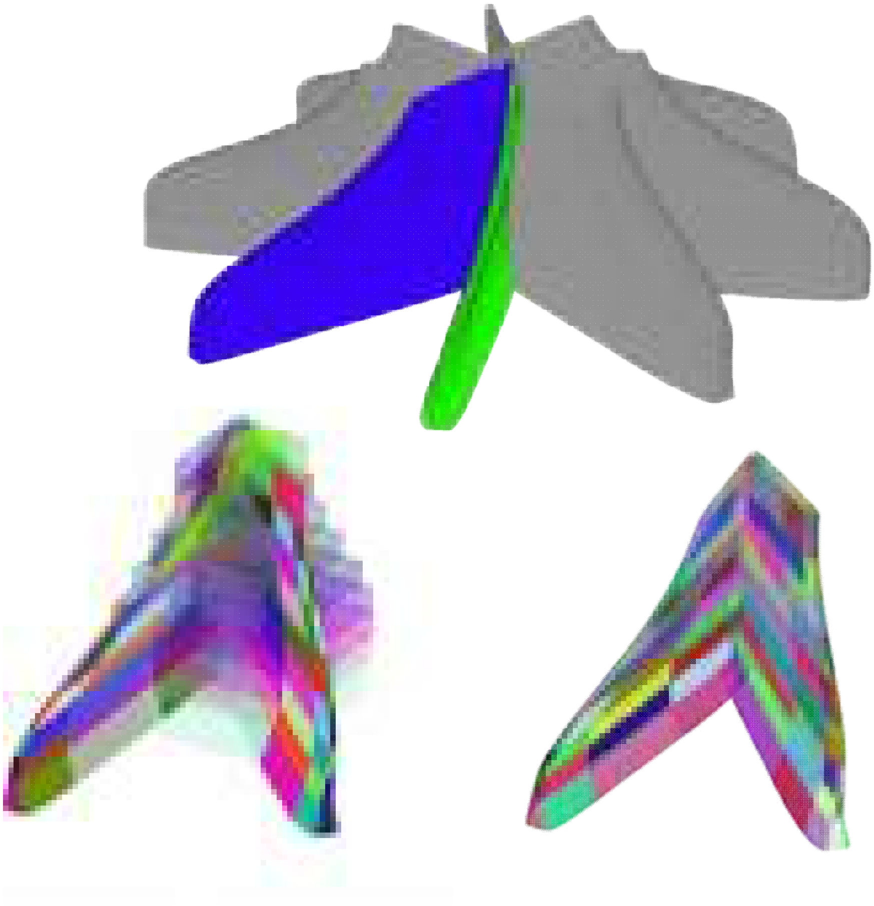

Fig. 9: Composing two turbine blades. The left image shows Bezier decomposition with bounding boxes defined over it.

8 CONCLUSION AND FUTURE WORK

A hybrid data structure for representing composed trivariate NURBS solids using exact and selective triangulation has been presented. This representation is a preprocessed output that can be utilized for performing quick classification for boolean operations. The need for such a representation is motivated by downstream operations for compositional analysis for which the boundary representation or trimmed representation is not available and would be inadequate without mesh generation.

A natural progression of this work would be carrying the presentation for use in analysis such as numerical integration over the composed domain or directly rendering of the Boolean output of the representation.

REFERENCES

[1] Rayasam, M.; Srinivasan, V.; Subbarayan, G.: CAD inspired hierarchical partition of unity constructions for NURBS-based, meshless design, analysis and optimization, International Journal for Numerical Methods in Engineering, 72, 2007, 1452–1489, doi:10.1002/nme.2046

[2] Hughes, T.; Cottrell, J.; Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer methods in applied mechanics and engineering, 194, 2005, 4135–4195, doi:10.1016/j.cma.2004.10.008
[3] Morgan, O. T.; Subbarayan, G.; Anderson, D. C.: Symbolic trivariate solid construction using MathML, under preparation.
[4] Martin, T.; Cohen, E.; Kirby, R.: Volumetric parameterization and trivariate B-spline fitting using harmonic functions, Computer Aided Geometric Design, 26, 2009, 648–664, doi:10.1016/j.cagd.2008.09.008
[5] Zhang, Y.; Bazilevs, Y.; Goswami, S.; Bajaj, C. L.; Hughes, T. J. R.: Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow, Computer methods in applied mechanics and engineering, 196, 2007, 2943–2959, doi:10.1016/j.cma.2007.02.009
[6] Sederberg, T. W.: Implicit and Parametric Curves for Computer Aided Geometric Design, PhD thesis, Purdue University, 1983.
[7] Katz S.; Sederberg, T. W.: Genus of the intersection curve of two rational surface patches, Computer Aided Geometric Design, 5(3), 1988, 253-258, doi:10.1016/0167-8396(88)90006-4
[8] Patrikalakis, N.: Surface-to-surface intersections, IEEE Computer Graphics and Applications, 13, 1993, 89-95. doi:10.1109/38.180122
[9] Krishnan S.; Manocha, D.: An efficient surface intersection algorithm based on lower-dimensional formulation, ACM Transactions on Graphics, 16, 1997, 74–106, doi:10.1145/237748.237751
[10] Dokken, T.; Juettler, B.; Skytt, V.: Challenges in surface-surface intersections, in Computational Methods for Algebraic Spline Surfaces, 11-26, 2005, Springer Berlin Heidelberg.
[11] Mukundan, H.; Ko, K. H.; Maekawa, T.; Sakkalis, T.; Patrikalakis, N. M.: Tracing surface intersections with validated ode system solver, in Proceedings of the ninth ACM symposium on Solid modeling and applications, SM '04, (Aire-la-Ville, Switzerland, Switzerland), 2004, 249-254.
[12] Turner, J.: Accurate solid modeling using polyhedral approximations, IEEE Computer Graphics and Applications, 8, May, 1988, 14-28. doi:10.1109/38.510
[13] Farouki, R. T.; Han, C. Y.; Hass, J.; Sederberg, T. W.: Topologically consistent trimmed surface approximations based on triangular patches, Computer Aided Geometric Design, 21, May 2004, 459-478.
[14] Wang, X.; Zhang, W.; Zhang, L.: Intersection of a ruled surface with a free-form surface, Numerical Algorithms, 46, 2007, 85–100, doi:10.1007/s11075-007-9118-y
[15] Puig-Pey, J.; Gálvez, A.; Iglesias, A.: A new differential approach for parametric-implicit surface intersection, Computational Science—ICCS 2003, 2003, 659–659.
[16] Casciola G.; Morigi, S.: The trimmed NURBS age, Advances in Computation: Theory and Practice; Recent Trends in Numerical Analysis, Ed. D. Trigiante, Nova Science, 1999.
[17] Casale, M.: Free-form Solid Modeling with trimmed surface patches, IEEE Computer Graphics and Applications, 1987, 33–43, doi:10.1109/MCG.1987.277025
[18] Weiler, K.: Edge-Based Data Structures for Solid Modeling in Curved-Surface Environments, IEEE Computer Graphics and Applications, 5, 1985, 21-40, doi:10.1109/MCG.1985.276271
[19] Peterson, J. W.: Tessellation of NURB Surfaces, Graphics Gems 4, Heckbert, P. S. ed., Academic Press, Boston, 1994, 286-320.
[20] Hertel, J. N. S.; Mantyla, M.; Mehlhorn, K.: Space sweep solves intersection of convex polyhedra, Acta Informatica, 21, December 1984, 501-519, doi:10.1007/BF00271644