

Maximal Disjoint Ball Decompositions for Shape Modeling and Analysis

Jiangce Chen^a, Horea T. Ilies^{a,*}

^a*Computational Design Laboratory (CDL), University of Connecticut, Storrs, CT 06269, USA*

Abstract

A large number of geometric representations have been proposed to address the needs of specific engineering applications. This, in turn, exacerbates the inherent challenges associated with system interoperability for downstream engineering applications.

In this paper, we define the Maximal Disjoint Ball Decomposition (MDBD) as the location of the largest d -dimensional closed balls recursively placed in the interior of a d -dimensional domain and show that the proposed decomposition can be used to provide an underlying common analysis framework for geometric models using different representation schemes. Importantly, our decomposition only relies on the ability of an existing geometric representation to compute distances, which must be supported by any valid geometric representation scheme, and does not require an explicit representation conversion. Moreover, MDBD is unique for a given domain up to rigid body transformation, reflection, as well as uniform scaling, and its formulation suggests appealing stability and robustness properties against small boundary modifications.

Furthermore, we show that MDBD can be used as a universal shape descriptor to perform shape similarity of models coming from various geometric representation schemes. A salient attribute of this decomposition is that it provides adequate support for key downstream applications for models coming from disparate geometric representations. For example, MDBD can be naturally used to carry out meshless solutions to boundary value problems; efficient collision detection; and 3D mesh generation of models that use any valid geometric representation scheme. Finally, our hierarchical formulation of the proposed Maximal Disjoint Ball Decomposition allows for a choice of model complexity at run-time to match the available computational resources.

Keywords: geometric representations, shape analysis, system interoperability, shape similarity.

1. Introduction

Numerous geometric representations have been proposed over the years to address the needs of specific engineering applications. For example, today's "gold standard" in CAD is the boundary representation with NURBS curves and surfaces, although there are other spline versions that are finding some success as analysis-suitable representations [1]; triangular meshes are the surface representation of choice in graphics-related applications; meshes are being used in solving boundary value problems with Finite Elements [2]; point clouds output by depth cameras are input into mesh reconstruction algorithms [3], which often involves user interaction - thus geometric processing methods operating directly on the point clouds have emerged as an alternative processing approach [4, 5, 6, 7]; cellular representations, such as octrees [8], voxels, and meshes [9, 10], are used to speed up or enable specific solution strategies for a variety of downstream applications; implicit geometric representations have found their niche in many applications involving some form of geometric fit, including fixture design and analysis [11], haptic-assisted assembly [12, 13, 14], and topology optimization [15, 16].

The wealth of geometric representations produced an even larger stream of algorithms designed to take advantage of specific representation schemes. On one hand, all current shape analysis frameworks assume a consistent representation of all models being analyzed, and this assumption requires all representation conversions to be successfully completed prior to undertaking

the analysis. On the other hand, converting between existing geometric representations is far from being a trivial or solved problem, and every such conversion generates some information loss that is in general not well understood. It is therefore not surprising that current shape analysis systems have limited ability to deal with models natively existing in different representations having different levels of informational content. In fact, the theoretical and practical challenges of interoperability prompted the development of alternative proposals aimed to control the interfaces between software systems rather than the format of the exchanged data [17].

1.1. Motivating Application Scenarios

The ability to measure the similarity between geometric information that exists in different geometric representations is essential for similarity search in geometric database systems [18] containing crowd-sourced geometric models of mechanical components. The current state of the art requires multiple representation conversions prior to measuring similarity with existing algorithms, subject to the limitations discussed above. Furthermore, consider the assembly of large and highly complex systems encountered, for example, in the context described in [19]. In such a case, the user faces the challenge of determining where a given component or subsystem needs to be spatially located to enable its correct assembly. In this context, an Augmented Reality (AR) framework can be used to fuse the information coming from different geometric representations, such as depth cameras and CAD systems, to provide correct assembly instructions to the user. Importantly, the associated computations may have to be performed on models that natively exist in distinct representations. The final example scenario presented here is that of the

*Corresponding author

Email addresses: jiangce.chen@uconn.edu (Jiangce Chen),
horea.ilies@uconn.edu (Horea T. Ilies)

traditional structural Finite Element Analysis, which requires geometric models to be converted to solid meshes in order to solve the corresponding boundary value problems. However, meshing continues to be an active area of research and remains a demanding geometric pre-processing step in FEA. To circumvent the mesh-related challenges, a variety of “meshfree” boundary value solution methods have been developed, but these operate only on one specific geometric representation, which are reviewed in [20].

It is clear that the representations that dominate a specific application domain may not adequately support other application domains without the use of representation conversion.

1.2. Proposed Framework

We aim to formulate a framework capable of unifying the geometric information that exists in multiple geometric representations without requiring an explicit representation conversion. To this end, we observe that the concept of *distance* is intrinsic to all valid geometric representations used in 3D geometric modeling, which suggests that distances should play an important role.

Furthermore, the circle is arguably the most studied shape in mathematics. For example, circle packings and their complex (the tangency patterns for circle packings are encoded as abstract simplicial 2-complexes K) in 2D, have been shown to bridge combinatorics and geometry [21]; the so-called Ford circles and spheres [22] reveal the approximation of real numbers by rationals; and sphere packing [23] is widely used in crystal chemistry. Our approach presented in this paper is inspired by the extensive research that has been carried out in the mathematics literature on circle packings and their complex K . Since our main focus in this paper is on 3D domains, we rely on a *unique* spherical decomposition of a domain Ω and on the associated complex K to develop a framework for shape analysis that can (1) interface with different geometric representations; (2) support geometric reasoning and standard geometric algorithms (similarity, segmentation, proximity/collision queries) and (3) support downstream applications including analysis and simulations.

Thus, we seek a geometric representation based on ball packings that is bijective [24], supports the extraction of compact shape descriptors that are invariant to rigid body transformations, and provides support for key downstream applications. In what follows, d -dimensional closed balls, where $d = 2, 3$, are discs in 2D and spheres in 3D that satisfy $\mathbf{y} \in E^d : d(\mathbf{x}, \mathbf{y}) \leq r$, where \mathbf{x} is the center and $r \in \mathbb{R}$ is the radius.

1.3. Background

Geometric Representation Schemes

The standard mathematical model for solids (informally, objects with boundaries both in 2D and 3D) is an *r-set* [24], and there are a variety of syntactically correct computer representations used to describe such an *r-set*. Representations schemes can be *complete* (i.e., unambiguous, that is, a representation corresponds to a single object) and *unique* (i.e., an object admits a single representation in the representation scheme), although some representations are neither. As discussed in [24], representations that are not complete or unique may still be adequate for specific applications. However, completeness becomes important when the modeling system using the particular representation scheme supports a wide variety of applications. The uniqueness of the shape representation implies the uniqueness of the shape descriptor within a given scheme and is clearly desired in shape similarity computations [25].

Shape Descriptors

Shape analysis is one of the key tasks in 3D modeling, and the associated techniques rely on shape descriptors that should be unique and invariant with respect to rigid body transformations along with possessing several other important properties [25]. A detailed review of existing shape descriptors is beyond the scope of this paper and can be found in [18, 6]. Here we comment on some of the attributes of the shape descriptors that are relevant to our objective. Importantly, the existing shape descriptors are defined for specific geometric representations, which implies that measuring the similarity of objects from distinct representations requires a conversion. At the same time, computing most shape descriptors from given representations requires significant computational resources [18], which could, in turn, impact their utility for real-time analysis of large data sets.

Various shape descriptors were designed for specific geometric representation schemes. For example, moments [26, 7] and spherical harmonics [27] are two global feature-based shape descriptors defined traditionally on voxelized geometric models. However, since high-order moments are sensitive to noise, only a small number of low order moments are used in practice, although these low-order moments tend to only capture global features of the shapes. Furthermore, the computational cost of moments increases exponentially with the size of the domain - a limitation shared by descriptors based on spherical harmonics.

The Heat Kernel Signature (HKS) [28] can capture local characteristics of a mesh or more recently of a point cloud model [6] and is robust against noise, which makes the HKS particularly appealing for measuring the similarity of point cloud models. These signatures are based on the heat diffusion properties of the model, and are stable against boundary noise. One key observation is that each geometric representation that the HKS operates on must have an appropriate definition of the discrete Laplacian, but there are infinitely many ways to define discrete Laplacians for meshes and point clouds. The fact that each such definition results in different convergence properties [29] implies that HKS-based signatures cannot be directly used to measure similarity between, say mesh and implicit models.

There are also many graph-based shape descriptors that capture local features of objects, such as the B-rep graphs [30] and Reeb graphs [31]. However, the B-rep graphs of similar models are not guaranteed to be similar [18], and the Reeb graph-based methods are not applicable to arbitrary shapes.

Spherical Decompositions

Spherical decompositions have been shown to be useful geometric constructions in several different application domains. In the context of analytic methods, overlapping spherical decompositions have been used to construct implicit representations of 3D domains. Specifically, a paradigm for efficient computation of analytic correlations based on a grid-free decomposition of potentially overlapping spheres was proposed in [14]. Within this framework, solids are represented as sublevel sets of summations of smooth radial kernels. By utilizing nonequispaced FFTs, one can unify convolution based applications, like holonomic collision constraints, shape complementary metrics, and morphological operations, within an analytic framework. A recent work investigating the quality of the approximations of a 2D domain by finite unions of balls has been presented in [32].

In addition, spherical decompositions have been used in meshing [33], where methods based on circle packings are used to generate high-quality surface elements for polygonal domains. In physics simulations, spherical decompositions offer direct

support to meshfree analysis methods, such as Method of Finite Spheres (MFS) [34, 35]. The MFS method, which operates directly on a decomposition of potentially overlapping spheres, uses compactly supported functions defined on a spherical cover of a domain. It has been shown that, even though MFS itself may be less efficient than the Finite Element Method (FEM) in solving the discrete system equations, it can overall be more efficient than FEM because it does not require a mesh of the geometry.

It has also been shown that non-equal spherical decompositions cover a domain more efficiently than uniform spherical decompositions because the former generally need fewer spheres to cover the domain than the latter [36]. Moreover, sphere inner trees (IST) have been proposed as an alternative data structure [37] for standard collision detection and proximity queries. Conceptually, IST are constructed at every level by placing the maximal sphere in the domain left after carving the volume occupied by the spheres placed at all previous steps. The focus in [37] is on keeping the computational cost of collision detection under control, rather than on a careful definition and investigation of the properties of the IST or on using IST for downstream applications. Arguably, the IST presented in [37] is not unique.

In [38], the authors proposed the so-called morphological shape decomposition applied to decompose a 2D object into a union of maximal discs. The morphological shape decomposition proposed in [38] is based on non-uniform circle packing and Minkowski operations, and is shown to be invariant to translation, rotation, and scaling. The morphological shape decomposition is similar with ISTs in the sense that it recursively seeks the maximal inscribable disk in the domain, and uses Minkowski operations to compute the decomposition as well as the corresponding approximation of the 2D object.

1.4. Contributions and Outline

In this paper, we define a new geometric representation, named MDBD, based on packing maximal disjoint closed balls inside a d-dimensional solid domain. We show that this representation is bijective [24], and that computing MDBD only requires a given representation to have the ability to compute distances. We further explore the mathematical properties of this representation, and illustrate its potential to shape analysis by establishing *universal shape descriptors* that operate directly on the proposed decomposition, and by using them to measure the similarity of models that natively exist in various representations. We also discuss the powerful support that MDBD offers for downstream applications, including various physical simulations. We note that MDBD is used in this paper not as a full-fledged geometric representation, but as a computational framework to unify the geometric information coming from multiple geometric representations without requiring an explicit representation conversion.

Observe that MDBD is a fundamentally different concept from that of the medial axis of the domain. The latter, of course, is the set of points of a domain that have at least two closest points on the boundary of the domain [39] or, in other words, the non-differentiable points of the distance function. On the other hand, computing MDBD does not require the computation of the medial axis. Instead, it is defined as the locations of the largest closed balls *recursively* placed in the interior of the domain. This difference is explored in more detail in sections 3 and 4.

Section 2 presents the formal definition of MDBD followed by a careful discussion of its key properties in section 3. Our implementation discussed in 4 is used to generate an ample number of examples illustrating the decomposition and its application to shape similarity in section 5. We conclude in section 6 with a

summary of the key contributions and some directions for future research.

2. Formulation

Our decomposition consists of a collection of carefully placed non-overlapping d-dimensional balls. For the rest of this paper, we are focusing on 3D domains, but the discussion applies to 2D domains as well.

Before we describe the concept formally, let's define a **disjoint spherical assembly** as a set of *non-overlapping* and *closed* d-dimensional balls $\{b_1, b_2, \dots, b_m, \dots\}$, such that $\text{int}b_i \cap \text{int}b_j = \emptyset, \forall i \neq j$, where $\text{int}X$ is the interior of a set X . In other words, two different balls in the same disjoint assembly can at most be tangent to each other. Let Ω be a compact regular semi-analytic subset of the Euclidean 3-space $\Omega \in \mathcal{P}(\mathbb{R}^3)$, also known as an *r-set* [24]. For a given domain Ω , the collection of all spherical assemblies contained in Ω is denoted by $\mathcal{A}(\Omega)$.

A **disjoint spherical decomposition** A_Ω of a solid domain Ω is a disjoint spherical assembly in $\mathcal{A}(\Omega)$ whose spheres satisfies

$$\bigcup_i b_i = \Omega. \quad (1)$$

This definition implies that a disjoint spherical decomposition A_Ω of a solid Ω is a complete cover¹ of Ω , and that, in general, A_Ω contains infinitely many balls. For 2D domains, the tangency patterns formed by tangent circles are encoded as abstract simplicial 2-complexes K [21], whereas for 3D domains the tangency patterns will be encoded as a simplicial 3-complex as discussed below.

It can be shown that domain Ω is completely described by the centers of all spheres b_i in a disjoint decomposition and by the associated sphere radii $r_i, i \in \mathbb{N} \setminus \{0\}$. Furthermore, we can order the radius list $\mathbf{r} = [r_1, r_2, r_3, \dots]$ and denote such an ordered list of radii by $R_o(A_\Omega)$. If $i < j$ we say that r_i is at a higher level than r_j in $R_o(A_\Omega)$.

2.1. Disjoint Spherical Assemblies as Totally Ordered Sets

Assume $A_\Omega, B_\Omega \in \mathcal{A}(\Omega)$, and let $R_o(A_\Omega)$ and $R_o(B_\Omega)$ be the corresponding ordered list of radii. Based on the above discussion, $R_o(A_\Omega)$ and $R_o(B_\Omega)$ are infinite sequences. If \mathbf{R}^∞ is the associated space of infinite sequences, then its dimensionality is *countably infinite*. Thus, addition and scalar multiplication are similar to those found in finite coordinate spaces and a standard basis consists of vectors \mathbf{e}^i whose i^{th} element is 1 with all other elements being 0. The space of infinite sequences, therefore, has a zero vector $\mathbf{0}$, which is also an infinite sequence. Furthermore, such a space has a number of interesting subspaces, such as a subspace that consists only of sequences that have a finite number of nonzero elements. Similarly, another subspace, which is useful in our context, consists of elements that have a finite number of elements above a given value. Such a subspace allows us to truncate the disjoint spherical decompositions as discussed in section 3.3.

We define a total order \geq on $\mathcal{A}(\Omega)$, and denote it by $(\mathcal{A}(\Omega), \geq)$, in terms of a lexicographic order of the two lists of radii [40]. The lexicographic order on \mathbf{R}^∞ is defined by $R_o(A_\Omega) \geq R_o(B_\Omega)$ if and

¹Observe that not all spherical assemblies in $\mathcal{A}(\Omega)$ provide a cover for Ω .

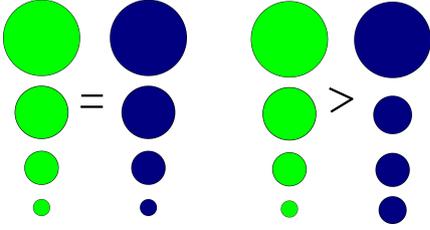


Figure 1: Ordering rule

only if one of the two conditions hold:

$$\begin{aligned} R_o(A_\Omega) - R_o(B_\Omega) &= \mathbf{0} \in \mathbf{R}^\infty \\ \exists k \in \mathbb{N} : R_o(A_\Omega)_i &= R_o(B_\Omega)_i \text{ for } i < k \text{ and } R_o(A_\Omega)_k > R_o(B_\Omega)_k \end{aligned} \quad (2)$$

where $R_o(X)_i$ is the i^{th} element of the ordered radius list of a set X .

For spherical decompositions $A_\Omega, B_\Omega \in \mathcal{A}(\Omega)$ with their corresponding radius lists $R_o(A_\Omega)$ and $R_o(B_\Omega)$, respectively, if $R_o(A_\Omega) \geq R_o(B_\Omega)$, we have $A_\Omega \geq B_\Omega$. Figure 1 shows examples of this ordering rule. The total order \geq satisfies the usual properties of antisymmetry, transitivity and reflexivity as shown in Appendix A.

2.2. Maximal Disjoint Ball Decomposition

Thus, a **disjoint spherical decomposition** A_Ω of a solid Ω is maximal if and only if

$$A_\Omega \geq B_\Omega, \quad \forall B_\Omega \in \mathcal{A}(\Omega). \quad (3)$$

We call an assembly $M_\Omega \in \mathcal{A}(\Omega)$ that satisfies (3) **the maximal disjoint ball decomposition** of Ω . It follows that M_Ω has the maximal radius values in its radius list $R_o(M_\Omega)$ according to the ordering (2).

2.3. Contact Graph

For a given circle packing of a domain, the tangency patterns of the circles are encoded as abstract simplicial 2-complexes K , and K triangulates an oriented topological surface [21]. This simplicial 2-complex K is also known as the contact graph of that particular circle packing and is always a planar graph. The circle centers along with the edges connecting the tangent circles of the packing form a planar embedding of the contact graph. The well known *Koebe-Andreev-Thurston* theorem states that the converse is also true. Specifically, the K-A-T theorem states that every triangulated planar graph G admits a *unique* circle packing in the plane whose contact graph is isomorphic to G [41], and its extension to general Riemann surfaces has been proven. These results allow one to use uniquely defined circle packings of domains to study the topology, geometry, and similarity of the corresponding planar domains.

Similarly, the contact graph of a ball decomposition A_Ω of Ω encodes the tangency pattern of the spheres in A_Ω , and we denote it by $G(A_\Omega)$. The set of vertices $V(A_\Omega)$ in the graph collects the center locations, while the set of edges $E(A_\Omega)$ stores the pairwise tangencies between spheres in the decomposition. An edge in $E(A_\Omega)$ formed by two vertices v_i and $v_j \in V(A_\Omega)$ is denoted by $v_i \sim v_j$; this implies that the corresponding spheres in the assembly b_i and $b_j \in A_\Omega$ are pairwise tangent. Consequently, the absence of an edge in $E(A_\Omega)$ connecting v_i and v_j indicates that $b_i \cap b_j = \emptyset$.

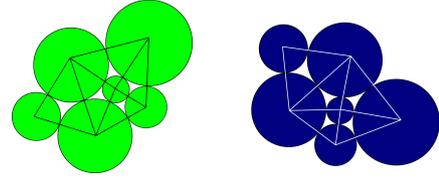


Figure 2: Two isomorphic graphs under a rigid body transformation.

Graph homomorphism can be seen as a generalization of graph colorings [42]. A *homomorphism* Φ from graph G to graph H is a map from the vertex set $V(G)$ to the vertex set $V(H)$ that takes edges to edges. In other words, if vertices u and v of G are connected by an edge, then $\Phi(u)$ and $\Phi(v)$ will also be connected by an edge in H . If the homomorphism map $\Phi : G \rightarrow H$ is bijective, and if Φ^{-1} is a homomorphism map from H to G , that is $\Phi^{-1} : H \rightarrow G$, then G and H are *isomorphic*.

The contact graph $G(M_\Omega)$ of the maximal decomposition $M_\Omega \in \mathcal{A}(\Omega)$ captures the topology of domain Ω [43]. Hence, we can use graph analysis tools [44] in order to perform topological and geometric analysis of the domain. One of the attractive properties of graph analysis is that graph topology is preserved under rigid body transformations, reflection, and uniform scaling. For example, the two contact graphs shown in Figure 2 are isomorphic under rigid body transformation. Thus, we conjecture that two solids Ω_1 and Ω_2 with isomorphic MDBD graphs *and* equal *normalized* radius lists are congruent up to rigid-body transformations, reflections and uniform scaling. These observations, together with the uniqueness properties of MDBD discussed in the next section, allows us to establish shape descriptors based on contact graph measures as discussed in section 5.

3. Properties

Following the notation from [24], a representation scheme s is a relation $s : \mathcal{M} \rightarrow \mathcal{R}$, where \mathcal{M} is an abstract modeling space whose elements are *r-sets*, and \mathcal{R} is the set of all syntactically correct representations r . The domain of s is denoted by D and the image of D under s by W . We repeat here three observations from [24]: (1) any representation $r \in W$ is valid, since it is both syntactically and semantically correct; (2) not all objects in \mathcal{M} are representable through s , i.e. there are elements of \mathcal{M} that are not in D ; and (3) syntactically correct representations r in \mathcal{R} may not be valid, i.e., W and \mathcal{R} may not be equal. Thus, a representation r in W is *complete* (i.e., *unambiguous*) if the set $s^{-1}(r)$ is a single element of D and is unique if a given element of D admits only r as a representation in W . The completeness and uniqueness of representations extend to representation schemes if all valid representations are complete and unique.

Representation schemes that are complete and unique are bijective maps and hence are highly desirable, but there are practical representations that are neither complete nor unique. In fact, many geometric representations are complete but not unique as detailed in [24]. However, for the range of applications that are considered in this work and described in section 1, both completeness and uniqueness are important.

3.1. Completeness of MDBD

The completeness of MDBD comes straight from its definition. Given a solid Ω and its corresponding MDBD $M_\Omega : \{b_1, b_2, \dots\}$, we have

$$\bigcup_i b_i = \Omega.$$

Thus, for a representation $r \in R$, $s^{-1}(r)$ is a single-element set $\{\Omega\} \subset D$. One direct consequence is that we can use M_Ω to fully reconstruct its original domain Ω . One way to implement this reconstruction is via morphological operations, as described in [14]. Given a maximal disjoint ball decomposition M_Ω of Ω , each $b_i \in M_\Omega$ could be considered as the translated and scaled instance of a base shape b_0 located at the origin and having a unit radius. Each 3D ball of M_Ω can be obtained from b_0 by a process that takes b_0 to a 4D cone followed by slicing the cone to generate sphere b_i with radius r_i as described in [14]. Consequently, Minkowski operations could be used with nonequiradius correlations to reconstruct the solid efficiently, at least in principle. We observe that in practical implementations one must use truncated versions of the disjoint decompositions, but the details of how one would go from such a decomposition to a reconstructed solid are outside the scope of this paper. Here it suffices to say that this process requires the computation of a cover² of set Ω .

3.2. Uniqueness

Since our representation scheme s is complete, there is only one solid $\Omega \in D$ mapped by s into our maximal disjoint ball decomposition $M_\Omega \in \mathcal{A}(\Omega)$. At the same time, the uniqueness of the maximal decomposition clearly depends on the uniqueness of the maxima of the distance function of the domain, since some domains admit multiple, and sometimes infinitely many maxima of their distance function. Thus, from a computational standpoint, finding the maximal assembly would require exhaustive comparisons between all possible decompositions of Ω , which is clearly impractical. Therefore we distinguish between cases that have finitely many or infinitely many maxima of the distance function. The latter case is explored below, and our implemented strategy is described in 4.2.

To illustrate our approach to handle domains that admit infinitely many maxima of the distance function, consider the simple rectangular domain shown in Figure 3(a), which also shows for illustrative purposes the medial axis of this rectangular domain. Observe that the domain is highly symmetric and that the segment of the medial axis connecting the two branch points of the medial axis contains points where the distance function has the same maximum value. Clearly, one can place a maximal disc at any one of these infinitely many maxima during the first step of the decomposition. Figures 3(b-d) illustrate several levels of the resulting decompositions for three distinct choices of maximal balls placed at the first level of the decomposition hierarchy, namely b_1 , c_1 , and d_1 . Even though the ball radii are equal for b_1 , c_1 , and d_1 , the balls placed at the next level will have smaller radii for the decomposition shown in Figure 3(b) compared to those shown in 3(c & d). For this domain, one can easily prove that the second level maximal balls will be largest if we add either ball c_1 or d_1 first. Consequently, according to the MDBD definition presented in section 2, the resulting decompositions will have the largest list of radii ordered lexicographically and hence the two decompositions will be maximal with, in this case, equal radius lists.

This discussion suggests that, by placing maximal balls at the branch points of domains that admit infinitely many maxima of the distance functions, we can obtain multiple MDBDs. However, a closer inspection reveals that the two contact graphs corresponding to Figures 3(c & d) are mirror images of each other and are, in fact, isomorphic. Since the corresponding radii lists

are equal, the two decompositions are actually the same up to isomorphism.

We therefore conjecture that, given a solid Ω and two disjoint spherical decompositions A_Ω and B_Ω in $\mathcal{A}(\Omega)$, if $R_o(A_\Omega) = R_o(B_\Omega)$, then their corresponding contact graphs $G(A_\Omega)$ and $G(B_\Omega)$ are isomorphic. In other words, our representation scheme MDBD is unique up to isomorphism.

The rectangular domain shown in Figure 3 leads to a broadly applicable insight. Specifically, for a domain that admits infinitely many maxima, the branch points of the medial axis favor maximal decompositions. This observation has practical applicability in finding MDBDs because it limits the search for MDBD to finitely many decompositions induced by finitely many branch points, as illustrated in Figure 4.

Another interesting observation is that the number of isomorphic maximal decompositions (i.e., decompositions with isomorphic contact graphs) seems to be related to the hierarchical symmetry of the domain. While a detailed investigation of this aspect is outside the scope of this paper, it is meaningful to observe what happens when we destroy the domain symmetry by adding a small feature to the domain, as illustrated in Figure 5. During the first step, maximal balls b_1 and c_1 are placed at the two branch points of the medial axis. However, because of the change in the symmetry of the original rectangular domain caused by the subtraction of the small feature, the decomposition shown in Figure 5(c) will quickly emerge as the MDBD of this domain.

3.3. Level of Detail

The proposed decomposition allows for a choice of model complexity at run-time to match the available computational resources. The balls having larger radii capture the larger geometric features of the domain, while the smaller balls reveal smaller features of the domain, as illustrated in Figure 6 as well as in the examples presented in Section 5. The proposed hierarchical decomposition together with the lexicographically ordered list of radii naturally lends itself to establishing ϵ -approximations of the domains. This, in turn, could be used in various applications of engineering interest, such as defeating [46] of CAD models in preparation for analysis. Furthermore, various similarity metrics can be defined on MDBD, see for example [44], to provide support for multiscale, global, or local feature matching.

By definition, an MDBD of a domain contains infinitely many spheres, which supports a theoretically complete reconstruction of the domain. However, practical algorithms must truncate the hierarchy of the decomposition, and the small spheres that are eliminated near the boundary do contain some of the domain's geometric information. Consequently, without some *a priori* knowledge of the geometry of the domain, a complete reconstruction from a truncated decomposition may not be possible, although the level of approximation provided by a truncated decomposition is quantifiable, at least in principle [47]. However, it is interesting to observe that polyhedral domains would only require a relatively small number of balls to perform a complete domain reconstruction. These observations suggest that the spherical decompositions could be used under certain assumptions as compact encoders of geometric information, although this avenue of research is outside the scope of this paper.

3.4. Stability

The concepts of MDBD and medial axis are similar in the sense that they both depend on characteristic points of the distance function. Specifically, MDBD finds the global maxima of the distance function of a domain, while the medial axis relies

²A *cover* of a set X is defined to be a subcollection of sets whose union covers X [45].

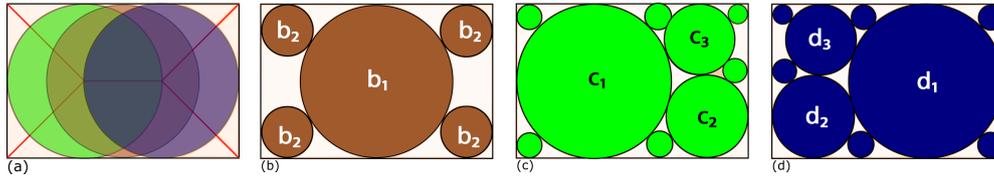


Figure 3: (a) A rectangular domain with infinitely many maxima of the distance function. (b) By placing maximal ball b_1 first, the next level balls have a smaller radius than those obtained in (c) and (d); For those cases, the maximal balls c_1 and d_1 are placed at the branch points of the medial axis, which results into decompositions with isomorphic contact graphs.

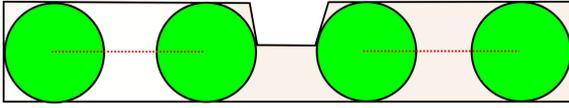


Figure 4: A domain with infinitely many maxima of the distance function is processed at this level by adding maximal balls at the branch points of the domain's medial axis.

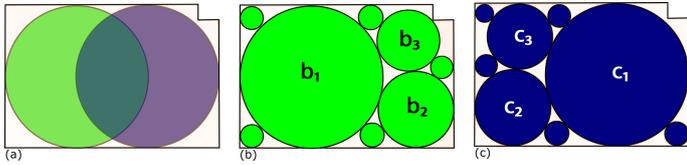


Figure 5: A change in the symmetry of the domain from Figure 4 favors one of the two isomorphic decompositions. For decompositions shown in (b) and (c), $c_1 = b_1$, and $c_2 = b_2$, but $c_3 > b_3$.

on its ridges. So, it should not be surprising that their stability properties are related as well, although there are some significant differences that we point out next.

Medial axis is notoriously unstable against small boundary perturbations [39] and it is well known that small changes in the boundary of the domain induce large changes to the structure of the medial axis. Its stability properties have been not only studied, but practical stable alternatives to the “Blum medial axis” have been proposed. For example, the λ -medial axis has been proposed in [48] as a subset of the domain’s “Blum medial axis” that captures the homotopy of the domain as long as λ is smaller than the “weak feature size” of the domain defined in terms of the minimum distance between the complement of the domain Ω^c and the critical points of Ω . The authors show that the λ -medial axis remains stable under small perturbations.

The fact that MDBD depends on the maxima of the distance function rather than on all of its non-differentiable points has important implications for the stability of MDBD. Specifically, the maximal values of the distance function of a domain with multiple (finitely or infinitely many) maxima are stable with respect to small boundary perturbations. Figure 7 shows a polygonal domain for which the effects of boundary perturbations remain localized. However, the geometry and topology of the set of the maximal points of the distance function for domains with infinitely many maxima can undergo significant changes even for small boundary perturbations as shown in Figure 8. This is so because even the smallest change in the boundary could collapse the infinitely many maxima into finitely many.

The solid modeling literature has dealt with issues raised by the uncertainty and the limited accuracy of the data. For example, instead of assuming that solids have exact (infinitesimally thin) boundaries, the ϵ -topological formulation [47, 49] assumes that the solids have finite size boundaries ϵ . In this formulation, the classical set topological notions, like interior i , closure k , and

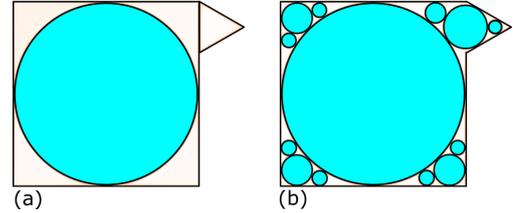


Figure 6: The MDBD of a square domain with a small triangular feature. (a) The largest circle accounts for general shape of the solid but cannot capture the small features that are being captured by balls introduced at subsequent levels of the hierarchy.

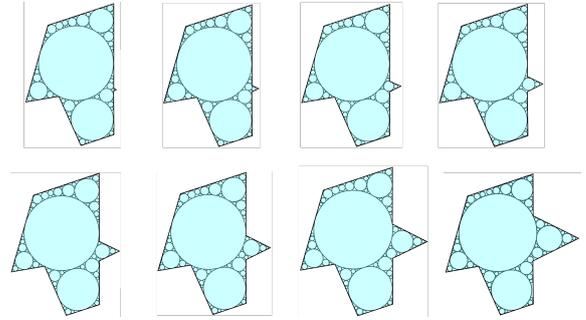


Figure 7: Boundary noise can have local effect on MDBD for domains with finitely many maxima.

boundary ∂ , become the equivalent ϵ -topological operations that tolerate imprecision. In turn, this formulation can be used to revise the classical solid modeling framework to include the notion of ϵ -solidity. Such notions of ϵ -boundary and ϵ -neighborhood, or, alternatively, other constructs, such as the Skeletal Density Function (SDF) [13], could be used to redefine versions of our maximal decomposition that are stable against boundary noise. Nevertheless, such reformulations are outside the scope of this paper.

4. Algorithms

4.1. Distance Field Computation

The distance field of a domain is a scalar field whose value at each point is the shortest distance between the point and the domain boundary. It is one of the key computational tasks in geometric modeling and has received significant attention over the last several decades. A good review of the principal methods that have been used to compute distance fields is provided in [50].

Traditional standard voxel-based algorithms for unsigned distance computation have a complexity of $O(mn^3)$, where m is the number of geometric primitive elements (points, segments, and triangles) constructing the model and $n \times n \times n$ is the number of voxels in the space. Rather than computing the distance to

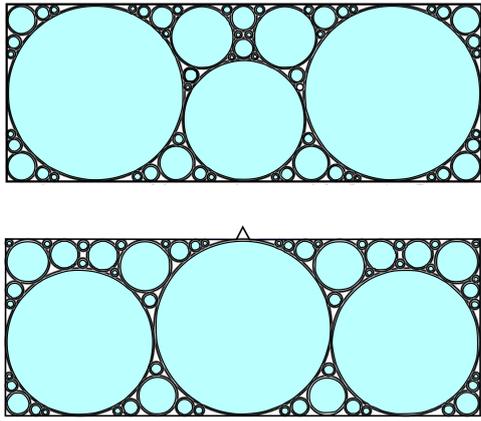


Figure 8: MBBD of shapes with infinitely many maxima could be sensitive to boundary noise.

the domain boundary for individual voxels, *HAVOC3D* [51] calculates the distance field for each primitive element of a solid with piecewise-linear boundary discretized into parallel slices and computes the rasterized Voronoi diagram using the OpenGL, the rendering pipeline, and the depth-buffer. *HAVOC3D*'s computational complexity is $O(mn)$. Some applications, including ours, require a signed rather than an unsigned distance field to distinguish between points that are interior and exterior to the domain. This distinction can be made by a point membership classification (PMC) test that outputs *in*, *on* and *out* points and generally carries a marked computational cost. This is one reason why recent work on distance field computation has focused on massive parallelization enhancements [52, 53], as well as on machine learning algorithms [54] to speed up the computations.

Most of the existing algorithms operate only on one specific geometric representation, which is typically a mesh. However, in our context, the distance field computations must handle multiple geometric representations. Consequently, in our preliminary implementation, we traded efficiency for generality and used *HAVOC3D* to compute the unsigned distance function, followed by a membership test that handles meshes, point clouds, and CAD geometry, as described below. Note, however, that one can easily use any other algorithm to compute the signed distance field (SDF).

In our implementation, *HAVOC3D* computes the rasterized unsigned distance function one layer at a time, followed by a PMC test to label points that are outside of the domain. A relatively straightforward approach to perform the PMC test is to first voxelize the domain, and there are a number of efficient voxelization methods implemented for objects bounded by surface meshes, such as [55, 56].

However, our voxelization must handle multiple representations within the same formulation. This is why we first approximate the contour of the solid in each slice with a piecewise linear closed planar curve. For boundary mesh models we intersect the plane of the slice with the boundary of the solid to obtain the intersection curve in that plane. For a point cloud model, each contour is computed by first projecting the boundary points that are in the neighborhood of the plane of the slice onto the slice, followed by constructing a piecewise linear approximation of the planar boundary. Once the contour is computed, we use the OpenGL tessellator to distinguish between points that are interior and exterior to the domain. The process of updating the distance function after adding balls to the decomposition is explained in Appendix B.

4.2. Shapes With Multiple Maxima

Enumerating and comparing the infinitely many spheres of the infinitely many spherical decompositions $\mathcal{A}(\Omega)$ of a set Ω is clearly impossible. A more practical approach is to iteratively place the maximal spheres in the domain, and subtract the corresponding volume from the domain (see Appendix B) as more maximal spheres are being added.

Thus, if there are finitely many locations for the maximal sphere at any given step, *and* the corresponding maximal spheres overlap, we explore each resulting branch of the hierarchy and compare lexicographically the corresponding radius lists as described in section 2 up to a preimposed *depth of explored levels* DL . If after DL steps the radius lists are equal, we randomly select one and continue the process. On the other hand, if there are infinitely many maxima of the distance function at that level of the hierarchy, we place maximal spheres at the finitely many branch points as mentioned in section 3. If these spheres overlap, we explore individual branches as described above; otherwise, we continue the process until we reach the minimum sphere size.

4.3. Prototype Implementation

We ran our algorithms on meshed and point cloud models to illustrate the capability to compute MBBD for diverse representations. Figure 9 illustrates the MBBD of several meshed models with triangulated boundaries, while Figure 10 shows a comparison of the MBBDs obtained from two different representations of the same object, namely mesh model and point cloud model. The point cloud models have been generated by triangle subdivision. Observe the differences in Figure 10 between the decompositions obtained from the two geometric representations: the former having a continuous boundary, while the latter having a boundary represented by discrete points. These differences are due to the differences in the corresponding distance functions, which converge to a common distance function as the sampling density of the point cloud increases. However, the similarity of the resulting MBBDs is not only visually apparent, but it is also shown computationally, as explained in Section 5.

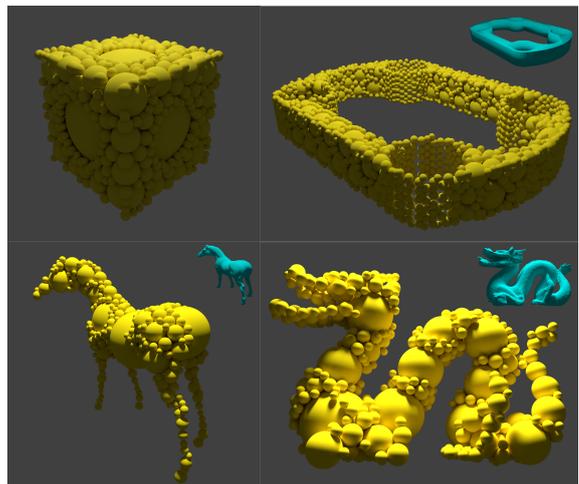


Figure 9: MBBD of mesh models.

5. Applications

5.1. Universal Shape Descriptors

Since MBBD is invariant to rigid body transformations, its attributes can be used to establish universal shape descriptors based

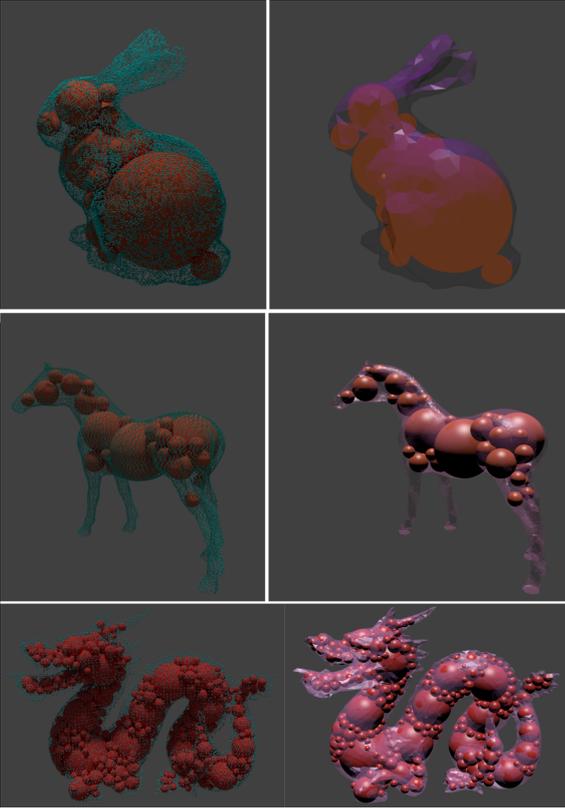


Figure 10: Comparing MDBD for mesh and point cloud representations of the same objects. Left column corresponds to point clouds, while the right column corresponds to mesh models.

on its radius list and on its associated contact graph. In principle, the radius list could be directly used as a shape descriptor to distinguish between different types of shapes.

However, the contact graph, particularly if augmented with weights based on sphere radii or connectivity information, captures much more geometric and topological information than the radius list itself. Consequently, one can build powerful shape descriptors based on these contact graphs. Several advanced deep learning methods have been developed recently for this purpose, such as [57, 58, 59], and, in principle, one can use any other graph-based shape signature in conjunction with MDBD. Nevertheless, in this paper, we aim to show that the maximal disjoint ball decomposition can be used effectively to establish similarity between models using distinct geometric representations. Consequently, we describe next a relatively simple similarity metric based on random walks on graphs [44], which have been widely used in graph kernels as a measure for graph similarity: short walks reveal local characteristics of the graphs, while long walks capture global properties.

Let $M_\Omega \in \mathcal{A}(\Omega)$ be the maximal disjoint ball decomposition of a domain Ω and let's consider the first n spheres of this maximal decomposition together with the associated ordered list of n radii $\mathbf{r}_n = \{r_1, \dots, r_n\}$, such that $r_1 \geq r_2 \dots \geq r_n$. Moreover, $G(M_\Omega) = \{V(M_\Omega), E(M_\Omega)\}$ is the associated contact graph, and we perform random walks on $G(M_\Omega)$.

Let the column vector $\mathbf{p}^t \in \mathbb{R}^n$ denote the probability distribution at time t . We denote by $p^t(v_i)$ the probability of being at vertex v_i at time t , and we assign the initial distribution over $V(M_\Omega)$ according to:

$$p^0(v_i) = \frac{r_i}{r_1}$$

and normalize the values. Let $\mathbf{W} \in \mathbb{R}^n \times \mathbb{R}^n$ be the weighted adjacency matrix of $G(M_\Omega)$. The element w_{ij} of \mathbf{W} denotes the probability to reach vertex v_i from v_j in one step:

$$w_{ij} = \frac{1}{\deg(v_j)},$$

where $\deg(v_j)$ is the degree of vertex v_j or, equivalently, the number of incident edges at v_j . The diagonal elements $w_{ii} = 0$.

The scores are iteratively redistributed during the random walk process according to

$$\mathbf{p}^t = \mathbf{W}\mathbf{p}^{t-1}.$$

followed by a normalization.

5.2. An Example Similarity Measure

To compare the similarity between two given models, we need a similarity metric to measure the similarity "distance" between the two models. We designed a universal similarity metric whose value relates to the degree of similarity between models so that the smaller the similarity "distance," the higher the geometric similarity between the two models.

Let $P_k^t(\Omega)$, $t = \{1, \dots, m\}$ be the score summation over the first k balls at step t of the random walk process on $G(M_\Omega)$, which is defined as

$$P_k^t(\Omega) = \lambda \sum_{i=1}^k p^t(v_i) \quad \lambda \in (0, 1),$$

where λ is used to decrease the weights of long walks. We denote the vector of all individual summations by $\mathbf{P}_k^m(\Omega) = [P_k^1(\Omega), \dots, P_k^m(\Omega)]$.

Thus, we form the feature vector for the similarity metric of a domain as the combination of its ordered radius list and vector $\mathbf{P}_k^m(\Omega)$, which contains information about the random walk process. Given a maximal decomposition M_Ω with n spheres, a number $k < n$, and m steps of the random walk process, the feature vector of domain Ω is defined as

$$\mathbf{f}(\Omega) = \{[r_1, \dots, r_k], \mathbf{P}_k^m(\Omega)\}. \quad (4)$$

In practice we only need a relatively small number of spheres $k < n$ in our feature vector.

Given two solids Ω_1 and Ω_2 , their corresponding feature vectors are defined by equation (4). Thus, we define a similarity distance as

$$d(\Omega_1, \Omega_2) = \|R_{ok}(M_{\Omega_1}) - R_{ok}(M_{\Omega_2})\|_2 + \alpha \|\mathbf{P}_k^m(\Omega_1) - \mathbf{P}_k^m(\Omega_2)\|_2,$$

where $R_{ok}(M_\Omega)$ is the ordered list of k radii for the maximal decomposition M_Ω , $\|\cdot\|_2$ is the usual ℓ_2 -norm in Euclidean spaces, and α is a weight that balances the contributions of the radius list and contact graph.

We tested this similarity measure on models obtained from the Shape COSEG Dataset [60, 61, 62]. Figure 11 shows 4 different query models, as well as 10 models from each class - some of which showing only subtle differences with our query models. Importantly, the database that we used contains two types of representations, namely meshes and point clouds. In our experiment we selected query models represented as point clouds, and, for each of the 4×10 models, we computed the first $n = 500$

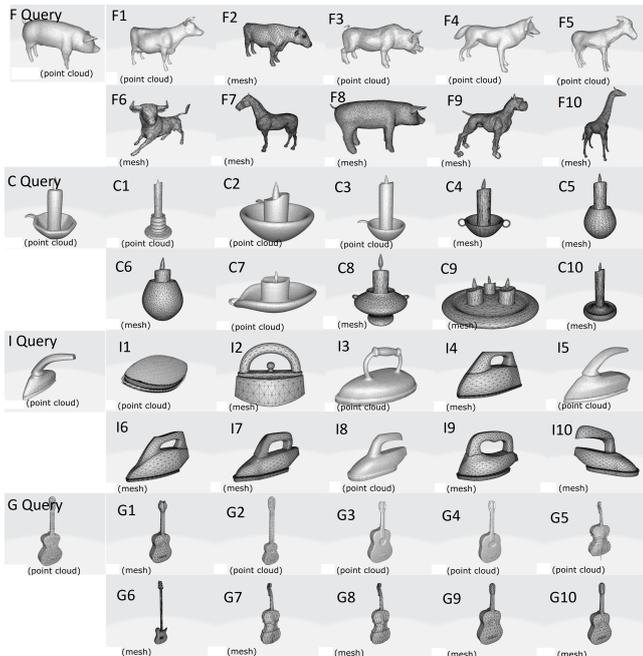


Figure 11: The dataset: F corresponds to the four-leg animal, C to candelabra models, I to iron models, and G to guitar models.

spheres of each model’s maximal disjoint spherical decomposition. For the feature extraction, we choose $k = m = 50$, $\lambda = 0.9$, $\alpha = 1$, and therefore each feature vector has 100 elements.

Figure 12 shows the similarity distances defined above between each query model and the 40 models from Figure 11. The smaller the value, the higher the similarity between models. Next to each graph in Figure 12 we show the models that our method measured to be the most similar with our query models from the group of models illustrated in Figure 11. Results do agree with human intuition, even though the similarity is computed between models having disparate geometric representations. In other words, even a relatively simple shape similarity metric based on MDBD, as defined above, correctly captures the similarity between models having only subtle differences and using distinct geometric representations.

To explore the stability of the similarity metric against boundary noises, which is tightly linked to the stability of MDBD, we add random noise of 0.5% and 1% to the four query models by using the Random Noise Displacement function in MeshLab. The size of the added random displacement is bounded by a user-defined value, which is the percentage of the diameter of the sphere that totally covers the specific model. As we can see from Figure 13, even though the noise produces changes to the values of the similarity metric, the general distribution of the metric across models remains essentially the same, which suggests that even with this simple metric the MDBD-based similarity computation is robust in practice against small boundary perturbations.

5.3. Shape Classification

Finally, a two-layer feed-forward network with sigmoid hidden neurons and softmax output neurons is trained to perform shape classification within Matlab’s Neural Net Clustering environment.

We use three large sets of 3D models from the Shape COSEG Dataset as our data for classification, which includes 200 models of “tele-aliens”, 300 models of “vases,” and 400 models of

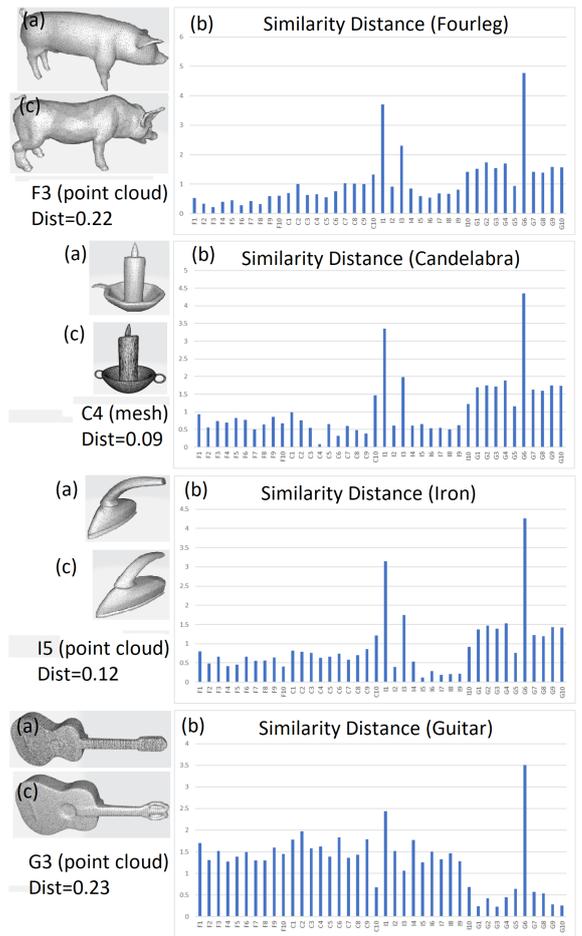


Figure 12: Panels (a) contain the point cloud query model; (b) the similarity distances between the query model and the rest of dataset; and (c) the model most similar with the query model in each panel (a).

“chairs.” All models are mesh models with triangulated boundaries. The 900 samples are randomly divided into 3 groups: 70% for training, 15% for validation, and 15% for testing. For each 3D model, an MDBD is computed for the first 800 spheres, which is then used to extract 130 features as the input to the neural networks, including the first 20 radii in the radius lists ($n = 20$), the score values of the first 10 ($k = 10$) vertices in 10 time steps ($m = 10$) of random walks. We input into the network: the radii $[r_1, \dots, r_n]$, the matrix of vectors $\mathbf{p}^t, t = \{1, \dots, m\}$ of size $k \times m$, and the m summation vectors $\mathbf{P}_k^m(\Omega)$ over the first k balls.

Figure 14 shows the confusion matrix of the neural network. Despite the simple structure of the neural network and of the signature that has been used, the overall accuracy is 95.4%, which is comparable to other published results performing shape similarity of geometric models.

5.4. Support for Downstream Applications

The data presented above shows that the proposed maximal disjoint spherical decomposition can be effectively and successfully used for shape analysis across geometric representations. However, MDBD offers strong support for other key downstream applications.

Collision detection is at the core of all physics engines in virtual reality environments, and various domain decompositions, including spherical decompositions, have been commonly employed to speed up the collision and penetration computations.

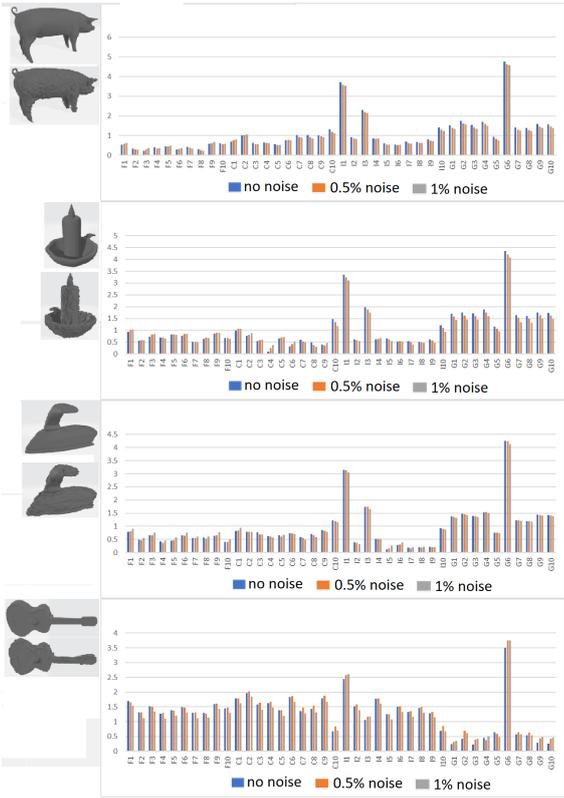


Figure 13: Models with no noise and models with 1% random boundary noise; The general distribution of the bar charts remains essentially the same.

The truncated MDBD of models using distinct geometric representations could be used directly in efficient collision detection algorithms - for example, in a manner similar to [37]. Furthermore, the hierarchical structure of MDBD is ideally suited for adjusting the model complexity at run-time based on the available computational resources in a manner similar to [14].

As mentioned in Section 1, spherical decompositions have been used in meshing [33, 63] to generate high-quality surface or volumetric meshes. For example, [33] describes a method based on circle packing to produce high quality quadrilateral surface elements. Furthermore, the BubbleMesh method proposed in [63] tightly and carefully packs spheres of equal radii inside the domain to be meshed followed by a constrained Delaunay triangulation of the sphere centers. One can easily envision designing 2D and 3D *adaptive* meshing algorithms based on truncated MDBD coupled with sphere subdivision.

On the other hand, meshing is in general time-consuming and involves a careful, often manual handling of mesh distortions. Consequently, a variety of meshless/meshfree finite element methods have been developed over the last 3 decades. One such method that is particularly promising is the method of finite spheres (MFS) [2, 35], which operates directly on a domain discretization with overlapping spheres, and altogether avoids the meshing generation. In [35] it is argued that despite the fact that the method of finite spheres is slower than the standard FEM, overall the computational cost is comparable once the pre-processing cost of meshing is taken into account. Once again, one can easily envision algorithms that would convert the MDBD of a domain into a complete cover of the domain, which would allow one to perform FEM based on the method of finite spheres directly on MDBD.

Finally, the larger spheres in MDBD capture large geometric

Training Confusion Matrix				Validation Confusion Matrix					
Output Class	1	190 30.2%	0 0.0%	10 1.6%	95.0%	48 35.6%	1 0.7%	1 0.7%	96.0%
	2	1 0.2%	140 22.2%	0 0.0%	99.3%	0 0.0%	27 20.0%	0 0.0%	100%
	3	16 2.5%	0 0.0%	273 43.3%	94.5%	5 3.7%	0 0.0%	53 39.3%	91.4%
		91.8%	100%	96.5%	95.7%	90.6%	96.4%	98.1%	94.8%
		8.2%	0.0%	3.5%	4.3%	9.4%	3.6%	1.9%	5.2%
		Target Class				Target Class			

Test Confusion Matrix				All Confusion Matrix					
Output Class	1	38 28.1%	3 2.2%	2 1.5%	88.4%	276 30.7%	4 0.4%	13 1.4%	94.2%
	2	0 0.0%	29 21.5%	0 0.0%	100%	1 0.1%	196 21.8%	0 0.0%	99.5%
	3	2 1.5%	0 0.0%	61 45.2%	96.8%	23 2.6%	0 0.0%	387 43.0%	94.4%
		95.0%	90.6%	96.8%	94.8%	92.0%	98.0%	96.8%	95.4%
		5.0%	9.4%	3.2%	5.2%	8.0%	2.0%	3.2%	4.6%
		Target Class				Target Class			

Figure 14: Confusion matrices: class 1 corresponds to “vases,” class 2 to “tele-aliens,” and class 3 to “chairs.”

features of the domain as well as its topology, while the smaller spheres tend to capture the small geometric features. At the same time, our algorithms end up placing spheres at the branch points of the medial axis. Consequently, by appropriately truncating small spheres and “branches” of MDBD, our proposed decomposition is ideally suited to perform model defeaturing and segmentation, perhaps in conjunction with graph clustering, such as [64].

6. Conclusions and Future Work

In this paper, we propose the *maximal disjoint ball decomposition* as an underlying common analysis framework for models using different geometric representation schemes. We showed that MDBD is unique up to rigid-body transformations, reflection, and uniform scaling, as well as complete, so it satisfies the requirements of a full-fledged geometric representation. Because MDBD is defined in terms of the maxima of the distance function of a domain, its stability properties are closely linked with the stability of the maxima of the distance function, as discussed in Section 3. We argued in that section that these global maxima of the distance function are in general more stable to boundary perturbations than the medial axis itself. Moreover, we argued that one could in principle use the ϵ -topological operations to redefine a version a stable version of MDBD, whose robustness properties could be proven mathematically. However, our examples show that MDBD, as defined here, is robust in practice.

One important attribute of MDBD is that it only relies on the ability of an existing geometric representation to compute distances, which must be supported by any valid geometric representation scheme, and does not require an explicit representation conversion. This is why MDBD can be used to define universal shape descriptors to support shape analysis and geometric reasoning tasks for models that use distinct geometric representation schemes, with important practical advantages for shape analysis as well as downstream applications. To this end, we showed how even a relatively simple MDBD-based similarity metric can be effectively and successfully used for shape analysis, and that it is robust in practice against small boundary perturbations.

We reviewed in section 2.3 some of the known results connecting the circle packings of 2D domains with the correspond-

ing contact graphs. It is worth noting that the *Koebe-Andreev-Thurston* theorem applies to *any* triangulated planar graph, including the one produced by our maximal ball decomposition for a planar domain. However, the mathematical literature lacks an extension of the *K-A-T* theorem to 3D or a formal treatment of the algebraic topological properties of these contact graphs. Inspired by the known 2D results, our exposition relies on two conjectures that we made for both 2D and 3D cases. Even though we do not currently have a proof for these conjectures, our numerous computational experiments show that they do hold in practice.

The work presented in this paper could be extended along several different directions. First, redefining MDBD by using concepts similar to those proposed in [48] and [47] would address the stability issues for those domains whose distance function possesses infinitely many maxima. At the same time, more powerful shape signatures and deep learning methods are bound to further improve on the already sensible performance exhibited by the examples presented in this paper. Finally, it would be worth exploring the support that MDBD offers to downstream applications as discussed in Section 5.

Acknowledgements

This work was supported in part by the National Science Foundation grants CMMI-1462759, IIS-1526249, and CMMI-1635103. We would like to thank the anonymous reviewers for their constructive feedback. The responsibility for any errors and omissions lies solely with the authors.

Appendix A. Totally ordered set

Let $A_\Omega, B_\Omega, C_\Omega$ be arbitrary elements in $\mathcal{A}(\Omega)$. We show that \leq is a partial order since it satisfies the following three properties [40]:

1. Reflexivity: $A_\Omega \leq A_\Omega$
2. Transitivity: If $A_\Omega \leq B_\Omega$ and $B_\Omega \leq C_\Omega$, then $A_\Omega \leq C_\Omega$
3. Antisymmetry: If $A_\Omega \leq B_\Omega$ and $B_\Omega \leq A_\Omega$, then $A_\Omega = B_\Omega$

Reflexivity:

Since $R_o(A_\Omega) = R_o(A_\Omega)$, we have $A_\Omega = A_\Omega$, so immediately, $A_\Omega \leq A_\Omega$.

Transitivity:

The transitivity property follows directly from the properties of the lexicographic order described in Section 2. Specifically, if $A_\Omega \leq B_\Omega$ and $B_\Omega \leq C_\Omega$, then $R_o(A_\Omega) \leq R_o(B_\Omega) \leq R_o(C_\Omega)$ so $R_o(A_\Omega) \leq R_o(C_\Omega)$, and therefore $A_\Omega \leq C_\Omega$.

Antisymmetry:

By definition, if $A_\Omega \leq B_\Omega$ then either $R_o(A_\Omega) < R_o(B_\Omega)$ or $R_o(A_\Omega) = R_o(B_\Omega)$. On the other hand, if $B_\Omega \leq A_\Omega$, then either $R_o(B_\Omega) < R_o(A_\Omega)$ or $R_o(B_\Omega) = R_o(A_\Omega)$. Since both sets of conditions have to be true, $R_o(A_\Omega) = R_o(B_\Omega)$, and, therefore, $A_\Omega = B_\Omega$.

Appendix B. Updating the Distance Function During Decomposition

The signed distance field needs to be updated for every new sphere being added to the decomposition. Specifically, if SDF_{Ω_i} is the signed distance field of a closed regular set Ω_i , and SDF_b is the signed distance field of sphere $b \in \Omega_i$, then the solid obtained by subtracting sphere b from Ω_i is

$$\Omega_{i+1} = \mathbf{r}(\Omega_i \cap b^c) \quad (\text{B.1})$$

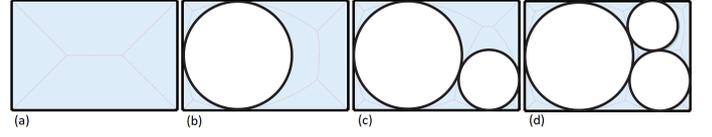


Figure B.15: Iteratively updating the domain shown in (a) for the first 3 iterations. Note that we compute the maxima of the signed distance function at each step and **not** the medial axis.

where $\mathbf{r}X$ is the regularization of a set X [24] and b^c is the complement of b . This is illustrated in Figure B.15 where the original rectangular domain Ω is iteratively updated as maximal spheres are being introduced at every step. We also show in Figures B.15(a-d) the ridges of the corresponding distance functions of the updated domains, although note that we only compute the *maxima* of the signed distance function and **not** the medial axis.

We choose the signs of these functions so that: (a) SDF_{Ω_i} is positive in the interior $i\Omega_i$ of Ω_i , and negative in the complement of Ω_i ; and (b) SDF_b is negative in the interior ib of sphere b and positive outside.

Therefore, the signed distance field of Ω_{i+1} is

$$SDF_{\Omega_{i+1}} = \min(SDF_{\Omega_i}, SDF_b) \quad (\text{B.2})$$

In our implementation of the signed distance function computation described in section 4.1, we slice the model in its original representation to compute the rasterized unsigned distance function, one layer at a time, followed by a PMC test to convert the unsigned into a signed distance function in each slice. We use equations (B.1) and (B.2) to update the SDF in each slice during the iterative decomposition directly in the graphics buffer. However, one can use any other SDF algorithm to compute SDF_{Ω_i} as discussed in Section 4.1.

References

- [1] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. Bordas, T. J. Hughes, T. W. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, *Computer Methods in Applied Mechanics and Engineering* 254 (2013) 197–221.
- [2] K.-J. Bathe, *Finite element method*, Wiley encyclopedia of computer science and engineering (2007) 1–12.
- [3] T. Schöps, T. Sattler, M. Pollefeys, Surfelmeshing: Online surfel-based mesh reconstruction, *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [4] N. Mitra, I. Kokkinos, P. Guerrero, N. Thuerey, V. Kim, L. Guibas, *CreativeAI: Deep learning for graphics SIGGRAPH 2019* (2019).
- [5] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, J. M. Solomon, Dynamic graph cnn for learning on point clouds, *ACM Transactions on Graphics (TOG)* 38 (5) (2019) 146.
- [6] R. M. Williams, H. T. Ilieş, Practical shape analysis and segmentation methods for point cloud models, *Computer Aided Geometric Design* 67 (2018) 97–120.
- [7] X. Zhao, H. T. Ilieş, Learned 3D shape descriptors for classifying 3D point cloud models, *Computer-Aided Design and Applications* 14 (4) (2017) 507–515.
- [8] D. Meagher, Geometric modeling using octree encoding, *Computer graphics and image processing* 19 (2) (1982) 129–147.
- [9] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, in: *ACM siggraph computer graphics*, Vol. 21, ACM, 1987, pp. 163–169.
- [10] A. Kaufman, Voxels as a computational representation of geometry, *The computational representation of geometry. SIGGRAPH 94* (1994) 45.
- [11] S. Nelaturi, A. Rangarajan, C. Fritz, T. Kurtoglu, Automated fixture configuration for rapid manufacturing planning, *Computer-Aided Design* 46 (2014) 160–169.

- [12] M. Behandish, H. T. Ilieş, Peg-in-hole revisited: A generic force model for haptic assembly, *Journal of Computing and Information Science in Engineering* 15 (4) (2015) 041004.
- [13] M. Behandish, H. T. Ilieş, Haptic assembly using skeletal densities and Fourier transforms, *Journal of Computing and Information Science in Engineering* 16 (2) (2016) 021002.
- [14] M. Behandish, H. T. Ilieş, Analytic methods for geometric modeling via spherical decomposition, *Computer-Aided Design* 70 (2016) 100–115.
- [15] J. Chen, V. Shapiro, K. Suresh, I. Tsukanov, Shape optimization with topological changes and parametric control, *International journal for numerical methods in engineering* 71 (3) (2007) 313–346.
- [16] M. Y. Wang, H. Zong, Q. Ma, Y. Tian, M. Zhou, Cellular level set in B-splines (CLIBS): A method for modeling and topology optimization of cellular structures, *Computer Methods in Applied Mechanics and Engineering* 349 (2019) 378–404.
- [17] C. Hoffmann, V. Shapiro, V. Srinivasan, Geometric interoperability via queries, *Computer-Aided Design* 46 (2014) 148–159.
- [18] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three-dimensional shape searching: state-of-the-art review and future trends, *Computer-Aided Design* 37 (5) (2005) 509–530.
- [19] Minerva : A mixed reality system for real-time construction problem resolution tool set demonstration [online] (November 2019).
- [20] A. Huerta, T. Belytschko, S. Fernández-Méndez, T. Rabczuk, X. Zhuang, M. Arroyo, Meshfree methods, *Encyclopedia of Computational Mechanics Second Edition* (2018) 1–38.
- [21] K. Stephenson, Circle packing: a mathematical tale, *Notices of the AMS* 50 (11) (2003) 1376–1388.
- [22] S. Northshield, Ford circles and spheres, arXiv preprint arXiv:1503.00813.
- [23] M. O’Keeffe, M. Eddaoudi, H. Li, T. Reineke, O. M. Yaghi, Frameworks for extended solids: geometrical design principles, *Journal of Solid State Chemistry* 152 (1) (2000) 3–20.
- [24] A. G. Requicha, Representations for rigid solids: Theory, methods, and systems, *ACM Computing Surveys (CSUR)* 12 (4) (1980) 437–464.
- [25] M. Yang, K. Kpalma, J. Ronsin, A survey of shape feature extraction techniques, *Pattern recognition* 15 (7) (2008) 43–90.
- [26] M. Elad, A. Tal, S. Ar, Content based retrieval of VRML objects—an iterative and interactive approach, in: *Multimedia 2001*, Springer, 2002, pp. 107–118.
- [27] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: *Symposium on geometry processing*, Vol. 6, 2003, pp. 156–164.
- [28] J. Sun, M. Ovsjanikov, L. Guibas, A concise and provably informative multi-scale signature based on heat diffusion, *Computer Graphics Forum* 28 (5) (2009) 1383–1392.
- [29] M. Wardetzky, S. Mathur, F. Kälberer, E. Grinspun, Discrete laplace operators: no free lunch, in: *Symposium on Geometry processing*, Aire-la-Ville, Switzerland, 2007, pp. 33–37.
- [30] M. El-Mehalawi, R. A. Miller, A database system of mechanical components based on geometric and topological similarity, part ii: indexing, retrieval, matching, and similarity assessment, *Computer-Aided Design* 35 (1) (2003) 95–105.
- [31] V. Barra, S. Biasotti, 3D shape retrieval using kernels on extended Reeb graphs, *Pattern Recognition* 46 (11) (2013) 2985–2999.
- [32] T. Nguyen, Finite unions of balls with inner and outer margins, Ph.D. thesis, Université Grenoble Alpes, (2018).
- [33] M. Bern, D. Eppstein, Quadrilateral meshing by circle packing, *International Journal of Computational Geometry & Applications* 10 (04) (2000) 347–360.
- [34] S. De, K. Bathe, The method of finite spheres, *Computational Mechanics* 25 (4) (2000) 329–345.
- [35] B. Lai, K.-J. Bathe, The method of finite spheres in three-dimensional linear static analysis, *Computers & Structures* 173 (2016) 161–173.
- [36] C. L. Bajaj, R. Chowdhury, V. Siddahanavalli, f^2 Dock: Fast Fourier protein-protein docking, *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 8 (1) (2011) 45–58.
- [37] R. Weller, G. Zachmann, Inner sphere trees for proximity and penetration queries, in: *Proceedings of Robotics: Science and Systems*, Vol. 2, Seattle, USA, 2009.
- [38] I. Pitas, A. N. Venetsanopoulos, Morphological shape decomposition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (1) (1990) 38–45.
- [39] D. Attali, J.-D. Boissonnat, H. Edelsbrunner, Stability and computation of medial axes—a state-of-the-art report, in: *Mathematical foundations of scientific visualization, computer graphics, and massive data exploration*, Springer, 2009, pp. 109–125.
- [40] E. Harzheim, *Ordered sets*, Vol. 7, Springer Science & Business Media, 2006.
- [41] S. Felsner, G. Rote, On primal-dual circle representations, in: J. Fineman, M. Mitzenmacher (Eds.), *Proceedings of the 2nd Symposium on Simplicity in Algorithms (SOSA 2019)*, San Diego, Vol. 69, 2018, pp. 8:1–8:18.
- [42] M. Chudnovsky, S. Huang, P. Rzazewski, S. Spirkl, M. Zhong, Complexity of c_k -coloring in hereditary classes of graphs, in: M. A. Bender, O. Svensson, G. Herman (Eds.), *27th Annual European Symposium on Algorithms (ESA 2019)*, Vol. 144, 2019, pp. 31:1–31:15.
- [43] K. Stephenson, *Introduction to circle packing: The theory of discrete analytic functions*, Cambridge University Press, 2005.
- [44] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, K. M. Borgwardt, Graph kernels, *Journal of Machine Learning Research* 11 (Apr) (2010) 1201–1242.
- [45] J. Munkres, *Topology*, Pearson Education, 2014.
- [46] A. F. Zubair, M. S. A. Mansor, Automatic feature recognition of regular features for symmetrical and non-symmetrical cylinder part using volume decomposition method, *Engineering with Computers* 34 (4) (2018) 843–863.
- [47] J. Qi, V. Shapiro, ε -Topological formulation of tolerant solid modeling, *Computer-Aided Design* 38 (4) (2006) 367–377.
- [48] F. Chazal, A. Lieutier, The “ λ -medial axis”, *Graphical Models* 67 (4) (2005) 304–331.
- [49] J. Qi, V. Shapiro, Epsilon-regular sets and intervals, in: *International Conference on Shape Modeling and Applications 2005 (SMI’05)*, IEEE, 2005, pp. 308–317.
- [50] M. W. Jones, J. A. Baerentzen, M. Sramek, 3D distance fields: A survey of techniques and applications, *IEEE Transactions on visualization and Computer Graphics* 12 (4) (2006) 581–599.
- [51] K. E. Hoff III, J. Keyser, M. Lin, D. Manocha, T. Culver, Fast computation of generalized Voronoi diagrams using graphics hardware, in: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 277–286.
- [52] H. Yu, J. Xie, K.-L. Ma, H. Kolla, J. H. Chen, Scalable parallel distance field construction for large-scale applications, *IEEE transactions on visualization and computer graphics* 21 (10) (2015) 1187–1200.
- [53] A. Roosing, O. Strickson, N. Nikiforakis, Fast distance fields for fluid dynamics mesh generation on graphics hardware, *Communications in Computational Physics* 26 (3) (2019) 654–680. doi:<https://doi.org/10.4208/cicp.0A-2018-013>.
- [54] J. J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove, DeepSDF: Learning continuous signed distance functions for shape representation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174.
- [55] E. Eisemann, X. Décoret, Single-pass GPU solid voxelization for real-time applications, in: *Proceedings of graphics interface 2008*, Canadian Information Processing Society, 2008, pp. 73–80.
- [56] M. Schwarz, H.-P. Seidel, Fast parallel surface and solid voxelization on GPUs, *ACM transactions on graphics (TOG)* 29 (6) (2010) 179.
- [57] M. Niepert, M. Ahmed, K. Kutzkov, Learning convolutional neural networks for graphs, in: *International conference on machine learning*, 2016, pp. 2014–2023.
- [58] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: *Advances in neural information processing systems*, 2016, pp. 3844–3852.
- [59] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, *IEEE Transactions on Neural Networks* 20 (1) (2009) 61–80.
- [60] Y. Wang, S. Asafi, O. Van Kaick, H. Zhang, D. Cohen-Or, B. Chen, Active co-analysis of a set of shapes, *ACM Transactions on Graphics (TOG)* 31 (6) (2012) 165.
- [61] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, D. Cohen-Or, Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering, *ACM Trans. on Graphics (Proc. SIGGRAPH Asia)* 30 (6) (2011) 126:1–126:10.
- [62] O. Van Kaick, A. Tagliasacchi, O. Sidi, H. Zhang, D. Cohen-Or, L. Wolf, G. Hamarneh, Prior knowledge for part correspondence, *Computer Graphics Forum (Proc. Eurographics)* 30 (2) (2011) 553–562.
- [63] K. Shimada, D. C. Gossard, Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing, in: *Proceedings of the third ACM symposium on Solid modeling and applications*, 1995, pp. 409–419.
- [64] Y. Zhou, H. Cheng, J. X. Yu, Graph clustering based on structural/attribute similarities, *Proceedings of the VLDB Endowment* 2 (1) (2009) 718–729.