Indexing 3D Scenes Using the Interaction Bisector Surface

XI ZHAO, HE WANG, and TAKU KOMURA
University of Edinburgh

The spatial relationship between different objects plays an important role in defining the context of scenes. Most previous 3D classification and retrieval methods take into account either the individual geometry of the objects or simple relationships between them such as the contacts or adjacencies. In this article we propose a new method for the classification and retrieval of 3D objects based on the Interaction Bisector Surface (IBS), a subset of the Voronoi diagram defined between objects. The IBS is a sophisticated representation that describes topological relationships such as whether an object is wrapped in, linked to, or tangled with others, as well as geometric relationships such as the distance between objects. We propose a hierarchical framework to index scenes by examining both the topological structure and the geometric attributes of the IBS. The topology-based indexing can compare spatial relations without being severely affected by local geometric details of the object. Geometric attributes can also be applied in comparing the precise way in which the objects are interacting with one another. Experimental results show that our method is effective at relationship classification and content-based relationship retrieval.

Categories and Subject Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Geometric algorithms, languages, and systems*

General Terms: Algorithms, Design, Experimentation, Theory

Additional Key Words and Phrases: Spatial relationships, classification, context-based retrieval

ACM Reference Format:

Xi Zhao, He Wang, and Taku Komura. Indexing 3D scenes using the interaction bisector surface. ACM Trans. Graph. 33, 3, Article 22 (May 2014), 14 pages.
DOI: http://dx.doi.org/10.1145/2574860

1. INTRODUCTION

Understanding contexts is important for applications such as managing 3D animated scenes and video surveillance. In such applications, individual geometry and movement of objects does not provide enough information and should be complemented by description of their interactions. For example, contexts such as "a boy wearing a cap" or "a book on a bookshelf" are defined by the fact that the upper half of the boy's head is covered by the inner area of the hat or the book is surrounded by other books and the bookshelf. Such contexts need to be described using a representation based on spatial relationships between different objects.

The importance of context is well recognized in the area of computer vision and image comprehension. Contextual data encoded by the adjacency information of individual objects in the image has been widely applied in shape matching [Belongie et al. 2002], annotation [Rabinovich et al. 2007], object detection [Giannarou and Stathaki 2007], and indexing [Harchaoui and Bach 2007]. In Harchaoui and Bach [2007], scene graphs are produced by connecting adjacent objects by an edge and conducting graph matching for scene comparison. The innovation of this approach is that it does not index images based on only individual object features, but also the spatial relations of multiple objects.

It is not an easy task to directly extend such an approach for 3D scenes where complex spatial relationships are present. Fisher and his colleagues encode the spatial context of 3D scenes using contacts between objects [Fisher et al. 2011] and the adjacency information is represented by relative vectors [Fisher and Hanrahan 2010]. Although such approaches can successfully classify static scenes where objects are correlated only by simple adjacencies or support, they may not be enough for encoding more complex relations such as enclosures, links, and tangles, or those that involve articulated models such as human bodies or deformable objects such as ropes and clothes. Therefore, a more descriptive representation that can evaluate the complex nature of interactions is needed for successfully indexing such spatial relationships.

In this article, we propose using the Interaction Bisector Surface (IBS), which is a subset of the Voronoi diagram, for the representation of the spatial context of the scene. The Voronoi diagram has been applied in indexing and recognizing the relationships of proteins in the area of biology [Kim et al. 2006]. In a similar manner to the Voronoi diagram, the IBS is the collection of points that are equidistant from at least two objects in the scene. The IBS can describe the topological and geometric nature of a spatial relationship. By computing a topological feature set called the Betti numbers of the IBS, we can detect relationships such as enclosures and windings, which characterize scenes such as a house surrounded by fences, a lady with a handbag hanging on her arm, or an object contained in a box. The geometric nature of the relationships can be analysed using the shape of the IBS, the direction of its normal vectors, and the distance between the objects and the IBS. The computation of the IBS makes minimal assumptions about the forms of data input, which can be polygon meshes, skeletons, or point-clouds, making it applicable to a wide range of existing data. In this article, we aim to analyse spatial relationships only based on the topological and geometric features, thus avoiding object labels as used in Fisher and Hanrahan [2010] and Fisher et al [2011].

Using the IBS as the interaction descriptor, we present the following three applications.

This work is supported EPSRC Standard Grant (EP/H012338/1), EU FP7/ TOMSY and China Scholarship Council.
Authors' addresses: X. Zhao, H. Wang (corresponding author), and T. Komura, School of Informatics, The University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, UK; email: tkomura@inf.ed.ac.uk.

DOI: http://dx.doi.org/10.1145/2574860

Interaction Classification. The topological and geometric features of the IBS can be used for the classification of different spatial relationships.

Automatic Construction of Scene Hierarchies. Using scenes that are composed of multiple objects such as room data, we show that the IBS can be applied in composing a hierarchy that describes the scene. Given an input scene, we group individual objects or object groups iteratively using a closeness metric based on the IBS.

Content-Based Relationship Retrieval. Our distance function can be used for finding similar relationships in the database, based purely on the relationship information.

Contributions.

—We provide a rich representation of relationships between objects in a scene, which can encode not only the geometric but also the topological nature of the spatial relationships;
—We present an automated mechanism to build hierarchical structures for scenes based on the spatial relationships of the objects;
—We show similarity metrics for object-object relations and an approach for conducting context-based relationship retrieval.

The rest of the article is organized as follows. After reviewing related work in Section 2, we explain how to compute the IBS in Section 3, and its topology and geometry features in Section 4. Then we propose an algorithm for building hierarchical structures for 3D scenes in Section 5. Based on the hierarchy, we explain how to represent and measure the similarity of spatial contexts of objects in Section 6. Next, we show the experimental results in Section 7 and finally discuss the methodology and draw conclusions in Section 8.

2. RELATED WORK

We will first review work about 3D analysis and synthesis, which is a relatively new topic in the area of computer graphics. As the medial axis is quite relevant to the IBS, we also review works about medial axis computation, and discuss the difference between the medial axis and the IBS.

Analysis of 3D Objects and Scenes. Recently, research into retrieval and synthesis of 3D objects and scenes has been growing due to the large amount of datasets available from, for example, Google Warehouse. Among such works, we are mainly interested in methods that use the spatial relationships between multiple components in the data to describe the entire object or scene.

Several methods to analyse the structure of man-made objects have been recently proposed [Wang et al. 2011; Kalogerakis et al. 2012; van Kaick et al. 2013a; Zheng et al. 2013a]. Wang et al. [2011] compute hierarchical pyramids of single objects based on symmetry and contact information. Kalogerakis et al. [2012] produce a probabilistic model of the shape structure from examples. Van Kaick et al. [2013a] use a co-hierarchical analysis to learn models' structure. Zheng et al. [2013a] build a graph structure from an object based on the spatial relationships of its components. As these methods are focused on single objects, the spatial structure of the objects is mainly based on the contact information, and the spatial relationships between separate parts are either ignored or only described by simple features such as relative vectors. Some recent works that aim to achieve shape matching [van Kaick et al. 2013b; Zheng et al. 2013b] propose new features based on pairwise points to encode the spatial context of shapes. These works also show that the spatial relationship between different parts of a 3D shape is important for shape understanding.

Structure analysis is also applied for scenes composed of multiple objects. Fisher et al. [2011] propose to construct scene graphs based on contextual groups and contact information between objects. The scenes are then compared by a kernel-based graph matching algorithm, which has been applied in image analysis [Harchaoui and Bach 2007]. The spatial relationships are highly abstracted by simple binary information of contacts. Yu et al. [2011] encode the relationships between furniture using metrics such as distance, orientation, and ergonomic measures. The objects are grouped into hierarchies to learn the furniture arrangements. The complex interactions are manually labelled by the users in these studies due to the difficulty of automatically learning them by simple measures. Fisher et al. [2012] learn contexts from examples by using Bayesian networks and mixture models. The relationships between adjacent objects are represented by relative vectors, and are compared using bipartite matching. Paraboschi et al. [2007] use the distance from the barycenter, height distance, and geodesic distance as a metric and compose a graph Laplacian to encode the relationship of adjacent objects. Tang et al. [2012] similarly encode the interactions of multiple characters by applying Delaunay tetrahedralization to the joints composing the character skeletons and computing the Hamming distance between them. These methods require the objects to be manually labelled in order to reinforce the simple representations used to describe the relationships. In many situations, however, the objects may not be tagged or tagged in an inconsistent manner.

In our case, we compare scenes which may be composed of unlabelled, dense mesh structures that may interact with one another in a complex manner. Such relationships are difficult to represent using simple relative vectors or distances. We cope with this problem by using a more expressive representation that takes into account the relationships of the entire surfaces of the objects composing the scenes.

Medial Axis and Shape Recognition. Here, we briefly review the medial axis computation and application, and how the medial axis is related to our work.

The medial axis of a 3D object is the set of points within the object that have more than one closest points on the boundary of the object. It has a long history of being used for the recognition of 2D and 3D shapes [Sebastian et al. 2001; Chang and Kimia 2011]. The success of the medial axis for shape recognition lies in the fact that it produces a discrete graph structure that abstracts the shape. As a result, the shape recognition problem can be converted into a graph matching problem, for which various efficient techniques have been proposed.

The previous research for computing the medial axis can be classified into two main categories: the continuous method and the discrete method. The continuous method [Culver et al. 1999; Sherbrooke et al. 1995] aims to compute an accurate medial axis for polyhedrons, while the discrete method [Amenta et al. 2001] approximates the medial axis by sample points on the boundary of the shape.

One issue with using the medial axis for pattern recognition is its instability, as small perturbations of the boundary of the shape can introduce large changes to its medial axis. Many methods, including Sud et al. [2007] and Imai [1996], are proposed to get the stable subset of the medial axis which is not sensitive to small perturbations. Our method does not suffer from such instability as we only use the bisector surface that is defined between two groups of polygons. We shall discuss more about this in Section 3.

The medial axis can be computed inside an object as well as at the external area of the object. The bisector surface, which is a subset of the medial axis, has been applied in representing protein

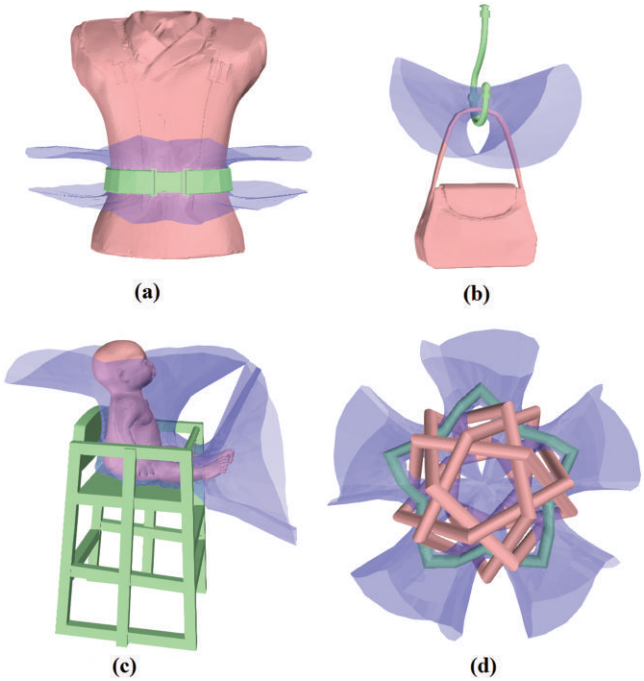

Fig. 1. Examples of the interaction bisector surface (the blue surface) for two parts of the 3D scene (shown as red and green): (a) belt on uniform; (b) bag on hook; (c) baby on chair; (d) a pentagon tangled with five other pentagons.

interactions [Kim et al. 2006]. Instead of dealing with specific structures like proteins, we define a more general metric that can compare the spatial relationship between interacting parts by measuring the features of the IBS. In our research, we use the IBS as a descriptor of the spatial relationships between objects in the scene.

3. INTERACTION BISECTOR SURFACE

Here we define the IBS and then describe how it is computed.

3.1 Definition

Given N point sets $S_1, S_2, \ldots S_N$ in the 3D space where $S_i = \{p_1^i, p_2^i, \ldots, p_{n_i}^i\}$, an Interaction Bisector Surface (IBS) divides the space into N regions with following properties.

—Points from the same point set lie in exactly one region.

—If a point $q \notin \{S_1 \cup S_2 \cup \ldots S_N\}$ lies in the same region as S_i, then the Hausdorff distance (in the Euclidean space) between set $\{q\}$ and S_i will be shorter than the Hausdorff distance between set $\{q\}$ and S_j, where S_j is any other point set.

The IBS is the set of points equidistant from two sets of points sampled on different objects. It is an approximation of the Voronoi diagram for objects in the scene. Examples of the IBS for different scenes are shown as blue surfaces in Figure 1. It can be either open or closed. Although the IBS can reach infinity when it is open (the same as the Voronoi diagram), we truncate it by a bounding sphere (details are given in Section 3.2). Despite the possibility that the IBS can produce a complicated polyhedral complex, it tends to form smooth shapes with stable topology when computed from objects in daily life such as those presented in the article.

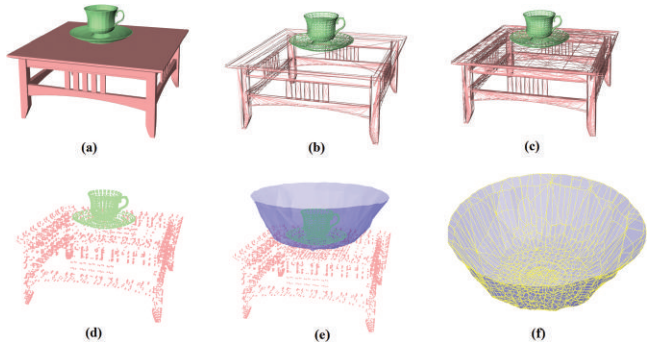

Fig. 2. Steps for computing IBS: Given a segmented scene (a) (in this example the scene has two segments: a cup and a table), which is composed of polygon meshes (b), we first subdivide the mesh to triangles of similar size (c) and then take the center points of each triangle (d). IBS consists of the Voronoi surfaces produced by two samples from different objects (e) and is also represented by a polygon mesh (f). The table and cup models are from the Stanford Scene Database [Fisher et al. 2012].

3.2 IBS Computation

Here we give details about how we compute the IBS for a given scene. We start by sampling points on the surfaces of the scene models uniformly, and then compute the Voronoi diagram for all these samples. The Quickhull algorithm [Barber et al. 1996] was used in this process. The result of the Quickhull algorithm is a simplicial complex consisting of polygons called *ridges*. Every ridge is equidistant to the two sample points which produce it. Hence there is a correspondence between ridges and the sample points. Assuming that the scene data is presegmented into objects, which is usually the case in scene data, we only select ridges that correspond to sample points from two different objects for computing the IBS. These steps are shown in Figure 2.

As the IBS by definition could reach infinity, we trim it by adding a bounding sphere to the scene data to compute the Voronoi diagram. In practice, the bounding sphere is found in the following way. We first find the minimum bounding box of the scene, and use the center of the bounding box as the center of the bounding sphere. The diameter of the sphere is set to 1.5 times the diagonal of the bounding box.

Special attention is needed if two objects are very close to each other, as there is a chance that the IBS will penetrate the objects due to the inadequate sampling density. In this case, we iteratively refine the IBS by the following process: if penetrations are found between the IBS and any object, we sample more on the parts of the object where the penetrations happen and recompute the entire IBS. Figure 3 shows the IBS between a table and a coffee cup. In this example, there are no more penetrations after four iterations.

Although the topological structure of the medial axis can be sensitive to subtle geometric changes of the relevant surfaces, the IBS is rather robust against such changes as it is computed between two objects. The instability of the medial axis is due to the "fluctuating spikes" [Attali et al. 2009], which are produced by concave dips on the surfaces (the grey branches in Figure 4). As the IBS is only produced between separate objects, such spikes are not included in its structure and therefore are less likely to be affected by subtle geometric changes of the object. More examples of the IBS of 3D object pairs are shown in Figure 1 and Figure 5.

Given a scene, we only need to compute the IBS for the whole scene once, and it already contains the spatial relation of every pair

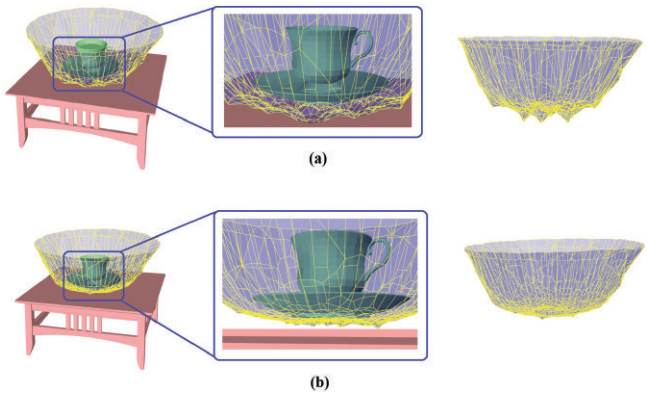

Fig. 3. (a) Penetrations between the models and the IBS, which are caused by the inadequate sampling on objects; (b) after 4 iterations refinement there is no penetration any more, and the shape of the IBS becomes smoother (the big gap between the cup and the table is for visualization purposes).

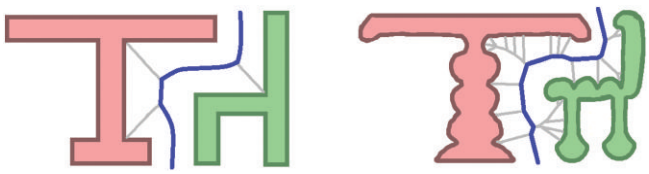

Fig. 4. The IBS (the blue line) is the stable part of the medial axis (the blue line and the grey lines). It does not fluctuate under subtle geometric changes.

of objects. We denote the subset of the IBS between object i and object j as $IBS(i, j)$. Furthermore, a subset of the IBS between two groups of objects, g_x and g_y, can be represented by $IBS(g_x, g_y) = \bigcup IBS(i, j)$ where $i \in g_x$ and $j \in g_y$.

4. IBS FEATURES

In this section, we give details about how to compute the topological and geometric features of the IBS.

4.1 Topological Features of the IBS

Topological descriptions of relationships are succinct and robust against small geometric variations. Consider a ball in a box. The description "in" here is irrelevant to the ball position or orientation as long as it is inside the box. Thus, capturing the topological nature of the interaction between two objects is crucial in relationship understanding. A good indicator of the topological nature is the Betti numbers of the IBS. We will first briefly give the definition of Betti numbers and then demonstrate how they can be applied as a feature to classify complex interactions.

The Betti number is a concept in algebraic topology. Formally, the k-th Betti number refers to the number of independent k-dimensional surfaces [Massey 1991]. We make use of the second (denoted as b_1) and third (denoted as b_2) Betti numbers in this research. They represent the number of 2D or "circular" holes (b_1), and the number of 3D holes or "voids" (b_2). Intuitively speaking, b_1 represents the number of "cuts" needed to transform a shape into a flat sheet. For example, objects that are laterally surrounded by others, such as a house surrounded by fences (see Figure 5(c)), form an IBS of a cylindrical shape, resulting in $b_1 = 1$. For objects tangled with other objects, such as toilet paper (see Figure 5(d)), a partial torus is generated, resulting in $b_1 = 2$. Moreover, b_1 can be even larger under complex interactions whose IBS involves a lot of loops (see Figure 5(e)). And b_2 represents the number of closed surfaces. In our scenario, it counts how many objects are wrapped by other objects (see Figure 5(b)). The Betti numbers can be easily computed from the mesh data by the incremental algorithm [Delfinado and Edelsbrunner 1995].

4.2 Geometric Features of the IBS

Although the Betti numbers can distinguish the qualitative difference of interactions, they cannot distinguish subtle differences. For example, the IBS of two boxes laterally adjacent to each other has exactly the same Betti numbers as that of an apple in a bowl. To address this problem, we evaluate the following geometric attributes of the IBS:

(1) geometric shape,
(2) distribution of the direction vectors, and
(3) distribution of distance between the IBS and the objects.

These features are computed at points sampled on the IBS. As different parts of the IBS are not equally descriptive of the relationship, we use an importance-based sampling scheme that is described in Appendix A. In brief, more points are sampled where the IBS is in close proximity with the objects defining it.

Geometric Shape of the IBS. The geometric shape of the IBS is useful for comparing the nature of the interactions. For example, when flat planes of two objects are simply parallel to each other, the IBS will become planar, but it will form a bowl shape when one object is surrounded by another object.

Various shape descriptors can be considered for the IBS. One possibility is to use the curvature profile; however, the curvature data can be unstable as the IBS may include ridges with sharp turns. This occurs when the mapping of the closest point between the IBS and the object becomes discontinuous due to the concavity of the object. Also, the IBS may be either an open or closed surface.

Taking into account these characteristics, we use the Point Feature Histogram (PFH) descriptor [Rusu et al. 2008a]; PFH is a histogram of the relative rotation between each pair of normals in the whole point-cloud. It describes the local geometrical properties by generalizing the mean curvature at every point. It provides an overall pose- and density-invariant feature which is robust to noise. PFH is applied for 3D point-cloud classification [Rusu et al. 2008a] and registration [Rusu et al. 2008b].

More specifically, for each sample point on a given IBS, we compute a 125-bin histogram of the relative rotation angles between the normal vector at the sample point and those of the other sample points. We produce a set of histograms for the whole IBS. Then we follow a method proposed in Alexandre [2012]. We compute the centroid and the standard deviation for each dimension of the histogram set, and use the resulting 250-dimension vector as the final feature of the IBS. More details for computing the PFH feature are described in Appendix B.

Direction. The normal vectors of sample points on an IBS contain the direction information about the spatial relationship. For example, if all the normal vectors of the IBS samples are pointing upwards, one of the objects forming the IBS is above the other.

The direction of the normal vector of each IBS sample is defined so that it points toward the reference object. In our definition, spatial relations are unidirectional. The relationship of A with respect to B is different from B with respect to A. Because of this, we first need

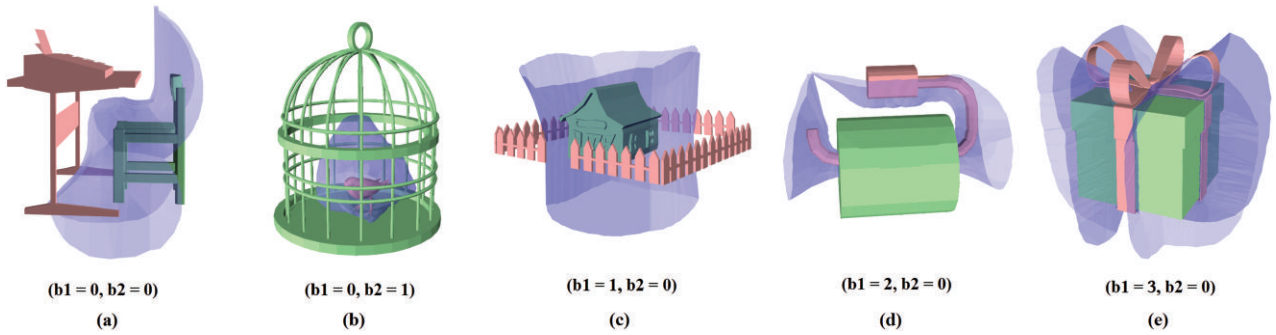

Fig. 5. The IBS (in blue) of two object scenes (a) table and chair; (b) bird in cage; (c) house surrounded by fences; (d) toilet paper on holder; (e) gift box and ribbon, and their Betti numbers. The 3D models in (a) and (c) are from the Princeton Shape Benchmark [Shilane et al. 2004].

to specify the reference object, and then use the normal direction that is defined on the side of the reference object.

The direction feature of the IBS is computed as follows. To reduce the dimensionality of the feature while maintaining the ability to tell the difference between relations such as "above" and "below", we use the angle between the normal vector and $+z$ direction (upwards direction), denoted here by θ, to compute the direction feature. We compute θ for each sample on an IBS, and produce a uniform histogram with 10 bins in the range of 0 to π. The number of samples that fall into each bin is counted and normalized against the total number of samples.

Distance between the Object Surface and IBS. The distribution of the distance between the IBS and the object surface is descriptive about the relations of the two objects. The larger the distance, the less likely that the two objects are closely related. We produce a uniform histogram with 10 bins whose range is between 0 to $0.5 \times d$, where d is the diagonal distance of the bounding box of the two objects. We compute the distance for each sample on the IBS, and accumulate the number of sample points that fall into each bin. The histogram is normalized by the total number of samples and is used as another geometric feature.

5. AUTOMATIC HIERARCHICAL SCENE ANALYSIS

In this section, we propose a method to automatically build a hierarchy out of a scene by making use of the IBS data. The method is an adapted version of the Hierarchical Agglomerative Clustering (HAC) algorithm [Hastie et al. 2009]. The resulting scene structure is used later for content-based relationship retrieval.

We first give the motivation, then a metric to measure inter-object and inter-group relations and finally an algorithm for constructing a hierarchy based on spatial relations.

5.1 Motivation

The idea to represent scenes by graph structures has been applied in content-based scene retrieval [Fisher and Hanrahan 2010; Fisher et al. 2011] and synthesis [Yu et al. 2011; Fisher et al. 2012]. In their works, the relationships between objects in a 3D scene are either generated from the information embedded manually at the design stage, or computed based on contact. Examples of this type of scene graph are shown in Figure 6(b) and (d).

The major difference between our method and previous works is that we adopt a multiresolution structure that encodes not only the spatial relations of the individual objects but also those between the object groups, which are more descriptive about the scene, especially when the number of scene components is large.

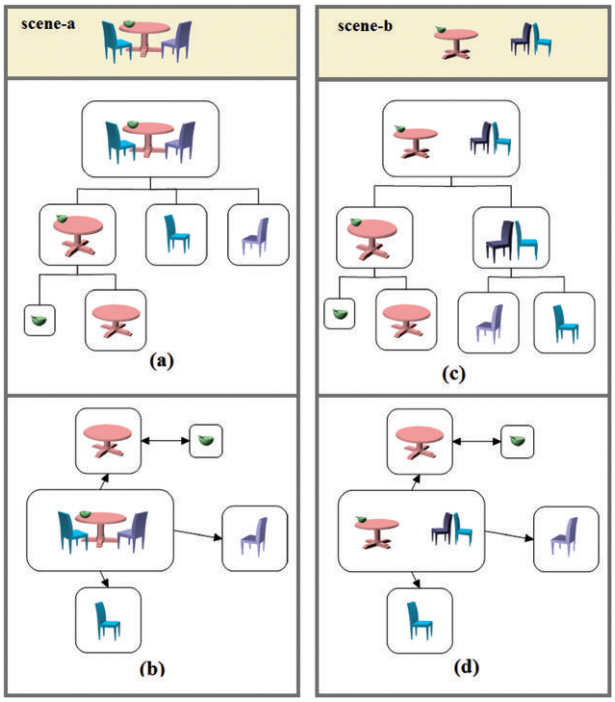

Fig. 6. Scene structures of two example scenes: scene (a) and scene (b). (a) and (c) show the hierarchical structures produced by our method; (b) and (d) show the scene graphs produced by the method in Fisher et al. [2011]. The 3D models shown in this figure are from the Stanford Scene Database [Fisher et al. 2012].

Let us first describe the advantage of considering the inter-group relationship with an example. For the sake of simplicity, we shall call an object group a *community*. A community containing only an object and its immediately surrounding objects is called the *local community* of the object. A larger community containing other objects further away in the scene is called the *extended community* of the object. In scene-b of Figure 6, the status of the bowl can be described through its local community (the table and the bowl) first, and then further described by the relation between the bowl's local community and other communities (the two chairs) in

the room. This description is far easier to recognize than using the raw, low-level relationships of all the individual objects as shown in Figure 6(c). The reason behind this is that humans tend to recognize a scene at the group level when observing it from a global perspective [Goldstein 2010], by aggregating objects based on proximity, continuation, uniformity, etc. Our multiresolution representation is also more descriptive than the raw graph used in previous work. This can be seen through scene-a and scene-b in Figure 6; the two are the same under the raw graph representation (Figure 6(b) and (d)) while the objects are grouped based on the spatial relationships and distinguished in our multiresolution representation (Figure 6(a) and (c)).

The terms "local" and "extended" community are only used for description purposes, and we do not arbitrarily classify neighbours into such categories. The inter-community relationships are produced by first grouping individual objects into communities of closer objects and then recursively grouping them into larger communities. The details of this procedure are described in Section 5.2. This structure naturally forms different abstraction levels of the scene. Given a reference object, the inter-community relationships on each level reflect the relationships between the reference object and the scene at different abstraction levels.

5.2 Closeness Measure and Hierarchy Construction

To formally define the hierarchy and the relationships between one object and its environment, we define a measure called *closeness* between communities that can contain only an object or a set of objects. Given a scene $\mathcal{S}$ with n communities, $G = \{g_1, g_1, \ldots, g_n\}$, the closeness measure between any two communities, g_x and g_y, is defined as

$$R_c(g_x, g_y) = R_{\textbf{ratio}}(g_x, g_y) + R_{\textbf{ratio}}(g_y, g_x), \tag{1}$$

$$R_{\textbf{ratio}}(g_x, g_y) = \frac{W(IBS(g_x, g_y))}{W(IBS(g_x, G \setminus g_x))}$$

$$IBS(g_x, g_y) = \bigcup_{i \in g_x, j \in g_y} IBS(i, j), \tag{2}$$

where $IBS(i, j)$ represents the IBS subset shared by object i and j. The function W computes the weighting of the IBS region. Note that simply computing the area of $IBS(i, j)$ does not give a good measure of the importance as mentioned in Section 4.2. In practice, we use $W(IBS(i, j)) = n$, where n is the number of sample points (that is described in Section 4.2) on the IBS shared between object i and j instead of computing its actual area. This is to weigh more the parts where the two communities are closely interacting with each other.

$R_{\textbf{ratio}}(g_x, g_y)$ is the *commitment* of g_x towards g_y; $R_{\textbf{ratio}}(g_x, g_y)$ is larger if g_x shares a large amount of the IBS with g_y than with other communities. It also means g_x commits more to g_y than to any other communities. Note that $R_{\textbf{ratio}}(g_x, g_y)$ is not necessarily symmetric. Essentially, R_c measures the relation between two communities under the context of the whole scene.

With R_c as a distance function, we present an adopted HAC algorithm to build a hierarchical structure of a scene. This hierarchy is built iteratively in a bottom-up fashion. Starting from individual objects (leaf nodes of the tree), we measure the R_c between nodes and group them into nodes that represent bigger communities. A merge can combine more than two nodes. This process is repeated until the whole scene is merged into one big single node. The details of the approach can be found in Algorithm 1. Figure 6(a) and (c) show simple examples.

ALGORITHM 1: Automatic Hierarchy Construction

Data: A scene S, grouping threshold τ, $0 \leq \tau \leq 1$
Result: A hierarchy H
Compute and sample IBS ;
The first level of grouping $G^0 = \{g_1, g_2, ...g_m\}$;
Initialize the current level $G = G_0$;
Initialize $H = \emptyset$;
Define next level $G' = \{g'_1, g'_2, ...g'_n\}$;
while $size(G) > 1$ **do**
  $H \leftarrow H \cup G$;
  n = size(G);
  compute matrix $\mathbf{M1}_{n*n}$: $\mathbf{M1}_{i,j} \leftarrow R_c(i, j)$ (Equation 1) ;
  compute $\mathbf{M2}_{n*n}$:
  $\mathbf{M2}_{i,j} \leftarrow \begin{cases} 1 & \text{if } g_i \text{ and } g_j \text{ have contact(s)} \\ 0 & \text{otherwise} \end{cases}$
  for $0 \leq i, j \leq n$ **do**
    if M2 $\neq$ **0 then**
      $\mathbf{M3}_{i,j} \leftarrow \mathbf{M1}_{i,j} * \mathbf{M2}_{i,j}$;
    else
      $\mathbf{M3}_{i,j} \leftarrow \mathbf{M1}_{i,j}$;
    end
    if $\mathbf{M3}_{i,j} > \tau$ **then**
      if $\exists g', g' \subset G', g_i \subset g'$ *or* $g_j \subset g'$ **then**
        if $g_j \not\subset g'$, $g' \leftarrow g' \cup g_j$, **else** $g' \leftarrow g' \cup g_i$;
      else
        build $g' \leftarrow g_i \cup g_j$ and $G' \leftarrow G' \cup g'$;
      end
    end
  end
  $G \leftarrow G'$;
end
$H \leftarrow H \cup G$

6. SIMILARITY METRICS BASED ON IBS

In this section, we describe how we make use of the features of the IBS and the scene structure to compute the similarity of interactions. We first explain the similarity measure of relationships between two objects. We then describe the similarity measure of relationships between reference objects and their immediate neighbors in the local community. Finally, we explain the similarity measure of relationships between objects and their extended communities. Note that these measures are only used for content-based retrieval explained in Section 7.3. For classification, a different equation based on radial basis function is used, which is explained in Section 7.1.

6.1 Similarity Measure for Relationships between Two Objects

Given an IBS, we can compute its feature $\boldsymbol{f} = \{f^b, f^{\text{PFH}}, f^{\text{dir}}, f^{\text{dis}}\}$. The four items are Betti numbers, PFH, direction, and distance, respectively, as explained in Section 4. To compare two IBS features $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$, we first use a simple Kronecker delta kernel as a measure for the topological features.

$$\delta\left(f_1^{\text{b}}, f_2^{\text{b}}\right) = \begin{cases} 1 & \text{if } f_1^{\text{b}} = f_2^{\text{b}} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Next we define a measure for the geometric features of the IBS that uses the L_1 distance of the PFH, direction, and distance features.

$$d_{\mathbf{geo}}(\boldsymbol{f}_1, \boldsymbol{f}_2) = a \cdot L_1\big(f_1^{\mathrm{PFH}}, f_2^{\mathrm{PFH}}\big) + b \cdot L_1\big(f_1^{\mathrm{dir}}, f_2^{\mathrm{dir}}\big) + c \cdot L_1\big(f_1^{\mathrm{dis}}, f_2^{\mathrm{dis}}\big), \quad (4)$$

where $a + b + c = 1$ ($0 \le a, b, c \le 1$). As the three features are in different ranges, we apply the *inverse variance weighting scheme* [Hartung et al. 2011] to the L_1 distance of three features. We set $a = 0.1$, $b = 0.4$, and $c = 0.5$ in our experiments.

We combine the topology and geometry measures of the IBS and compute the final similarity between two IBS by

$$s_{\mathbf{sr}}(\boldsymbol{f}_1, \boldsymbol{f}_2) = \delta\big(f_1^{\mathrm{b}}, f_2^{\mathrm{b}}\big)^w \cdot (1 - d_{\mathbf{geo}}(\boldsymbol{f}_1, \boldsymbol{f}_2)), \quad (5)$$

where w is a "switch" for using topological features. From the experiment in Section 7.1 we can see that Betti number is quite useful for complex interactions like tangles or enclosures, while it can contradict geometric features for data that contains penetrations. The metric function for measuring the similarity between two sets of IBS features should be defined based on the nature of the dataset and the purpose of retrieval. If the data mainly contains complex relations, w should be 1 so that the Betti number is used as a filter for different interaction types with respect to its topology; if the data mainly contains simple relations that have Betti numbers $b_1 = 0, b_2 = 0$, w should be set to 0 to speed up the computation and avoid the influence of possible penetrations.

6.2 Similarity Measure for Local Communities

We now describe how we can compare two objects with respect to their local communities. We define a profile of object o_i in a local community $g = \{o_1, o_2, \ldots o_m\}$ by

$$\boldsymbol{f}_{\mathbf{local}_i} = \bigcup_{1 \le j \le m, j \ne i} \boldsymbol{f}_{i,j}, \quad (6)$$

where $\boldsymbol{f}_{i,j}$ is the feature computed from the IBS between o_i and o_j. Therefore, $\boldsymbol{f}_{\mathbf{local}_i}$ is the set of IBS features between o_i and all the other objects in g. We call $\boldsymbol{f}_{\mathbf{local}_i}$ the *local profile* of o_i. Given two objects o_i, o_i' from different communities g and g', their local profiles $\boldsymbol{f}_{\mathbf{local}_i}$ and $\boldsymbol{f}_{\mathbf{local}_{i'}}$ are first computed. Then we can compute the similarity between o_i and o_i' under the contexts of their local communities. We define a similarity measure $s_{\mathbf{local}}$, normalized in a way similar to the graph kernel normalization [Fisher et al. 2011]

$$s_{\mathbf{local}}(i, i') = \frac{K(i, i')}{max(K(i, i), K(i', i'))}, \quad (7)$$

$$K(i, i') = \sum_{\boldsymbol{f}_1 \in \boldsymbol{f}_{\mathbf{local}_i}} \sum_{\boldsymbol{f}_2 \in \boldsymbol{f}_{\mathbf{local}_{i'}}} s_{\mathbf{sr}}(\boldsymbol{f}_1, \boldsymbol{f}_2), \quad (8)$$

where $s_{\mathbf{sr}}$ is defined in Eq. (5).

6.3 Similarity Metric for Extended Neighborhood

After defining $s_{\mathbf{local}}$, we are ready to combine it with the hierarchical structure and define a profile for an object o_i at every level of the hierarchy. For a scene $\mathcal{S}$ and its hierarchy, we assume that the leaf nodes are at level 1. Let l_d denote the nodes at d-th level so that l_d is a set of communities $\{g_1^d, g_2^d, \ldots, g_m^d\}$. Assume an object $o_i \in g_x^d$, $1 \le x \le m$, a profile of o_i at the d-th level is defined as

$$\boldsymbol{f}_{\mathbf{ext}_i}^d = \bigcup_{1 \le y \le m, y \ne x} \boldsymbol{f}_{g_x^d, g_y^d}, \quad (9)$$

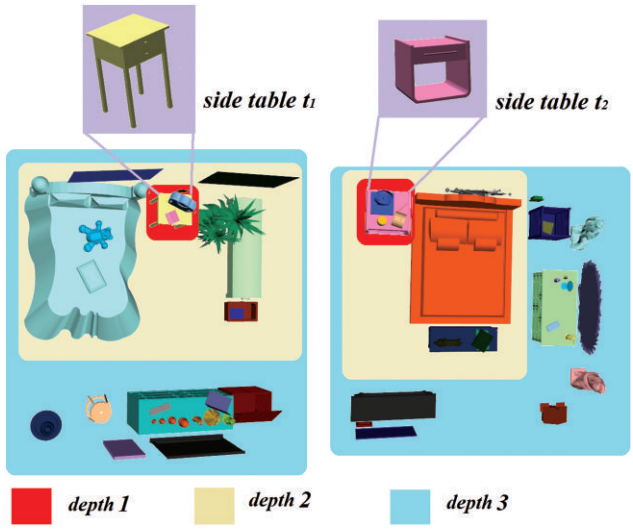

Fig. 7. An example of hierarchical comparison. The two side tables are the "center" objects we want to compare. The red, yellow, and light blue regions contain the side table's neighbors in level 1 (bottom level), level 2, and level 3 of the scene structures respectively. The 3D models are from the Stanford Scene Database [Fisher et al. 2012].

where $\boldsymbol{f}_{g_x^d, g_y^d}$ is the IBS feature set computed from $IBS(g_x^d, g_y^d)$, which is the IBS subset shared by community g_x^d and g_y^d. Given two objects o_i and o_i', we can compute their profile $\boldsymbol{f}_{\mathbf{ext}_i}^d$ and $\boldsymbol{f}_{\mathbf{ext}_{i'}}^d$. Then the similarity between o_i and o_i' at level d can be computed by

$$s_{\mathbf{ext}_d}(i, i') = \frac{K_e(i, i')}{max(K_e(i, i), K_e(i', i'))}, \quad (10)$$

$$K_e(i, i') = \sum_{\boldsymbol{f}_1 \in \boldsymbol{f}_{\mathbf{ext}_i}^d} \sum_{\boldsymbol{f}_2 \in \boldsymbol{f}_{\mathbf{ext}_{i'}}^d} s_{\mathbf{sr}}(\boldsymbol{f}_1, \boldsymbol{f}_2). \quad (11)$$

Finally, given a search-depth parameter $d_{\mathbf{depth}}$, we can find the similarity between object o_i and o_i' by accumulating their similarities from level 1 to $d_{\mathbf{depth}}$

$$s_{\mathbf{all}}(i, i') = \sum_{d=1}^{d_{\mathbf{depth}}} \gamma^{d-1} s_{\mathbf{ext}_d}(i, i'), \quad (12)$$

where γ is set to 0.5 taken to the power of d at each level. This is the contextual similarity for two objects in the database up to a given level.

A detailed example can be found in Figure 7. Assume that we want to compare side table t_1 and side table t_2 in two scenes. If $d_{\mathbf{depth}} = 1$, then only $s_{\mathbf{local}}(t_1, t_2)$ is calculated. The only objects involved are the objects on top of t_1 and t_2. If $d_{\mathbf{depth}} = 2$, then $s_{\mathbf{ext}_1} = s_{\mathbf{local}}(t_1, t_2)$ and $s_{\mathbf{ext}_2}(t_1, t_2)$ is calculated based on the IBS subset between the red areas and other areas within level 2. Finally, $s_{\mathbf{all}}(t_1, t_2) = s_{\mathbf{ext}_1} + 0.5 \times s_{\mathbf{ext}_2}(t_1, t_2)$.

Our idea to take account of the extended communities for comparing the status of an object in a scene resembles the part-in-whole queries in Shapira et al. [2010]. In Shapira et al. [2010], a hierarchical structure of objects is constructed, and when one part of the object is compared with other parts of another object, how the part locates with respect to the other parts in the hierarchy is taken into account and the similarity is computed based on the maximum flow

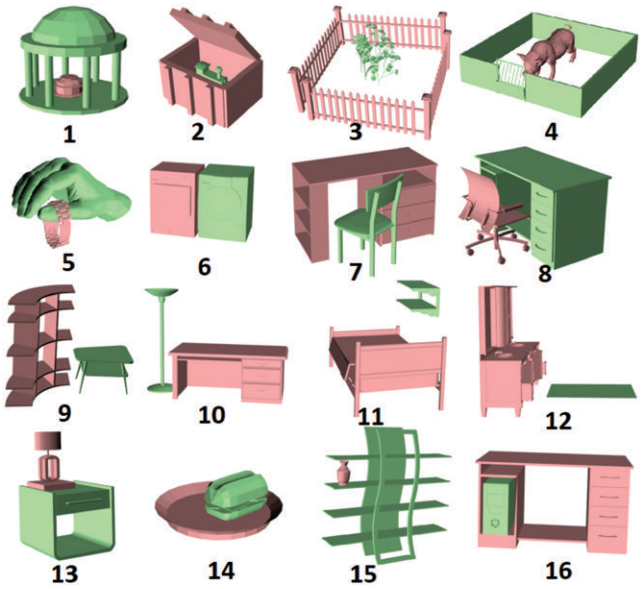

Fig. 8. Examples from 16 classes in our database. One example for each class. The 3D models shown above are from the Princeton Shape Benchmark [Shilane et al. 2004] (1–5) and the Stanford Scene Database [Fisher et al. 2012] (6–16).

in a bipartite graph. While their method focuses on the geometrical similarity of the parts in the hierarchy, our method computes similarities purely based on the relationship similarities. Also, we change the weights according to the distance such that the extended neighborhood is less influential to the results; this is due to the nature of the data we handle.

7. EXPERIMENTS AND EVALUATION

In this section, we present three experiments. The first is supervised classification of interactions between two objects, the second is building hierarchical structures for 3D scenes, and the third is relationship-based retrieval. For each experiment, we first explain the idea, then give experimental settings and results, and present the evaluation at the end.

7.1 Classification of Interactions

Here we show how geometrical and topological features in different combinations help in classifying two-object relationships.

Experiment. The dataset we use contains 1381 items, each an object pair. We ask the user to label them based on their spatial relations. The database consists of 16 classes. We show one example in each class in Figure 8. Descriptions for these classes are summarized in Table I. Note that there are some scenes from different classes with identical geometry but different object order, as the spatial relation between two objects is not symmetric. For example, there are two types of relation "enclose"; one object is enclosed by another and one object encloses the other. This is the same with other relation types except type 5 and type 6.

In order to facilitate the description, we refer to the interactions with Betti numbers $b_1 = 0, b_2 = 0$ as *simple relations*, and *complex relations* otherwise. The first part of our database contains 1289 items that are extracted from the Stanford Scene Database used in Fisher et al. [2012]. Since this mainly consists of simple relations,

Table I. Descriptions of Spatial Relationships of 16 Classes

Examples	Description	Examples	Description
1, 2	Enclose	3, 4	Encircle
5	Interlocked	6	Side by side, similar sizes
7, 8	Tucked in	9, 10	Side by side, one considerably higher
11, 12	Loosely above	13, 14	On top of
15, 16	Partially inside, with open areas		

we denote it as S. We manually label the data into meaningful classes, which turn out to be 11 classes (class 6 to class 16 in Figure 8). The second part of the database contains 92 examples of complex relations labelled into 5 classes (class 1 to 5 in Figure 8) by the user. As all data in this part represents complex relations, we refer to it as C. More examples from these 16 classes are shown in the supplementary material.

We do the experiments first on S and C individually and then on the whole database $S + C$. In each experiment, the data is split into a training set and a testing set in the ratio of 7:3. We performed classification on different combinations of features to investigate how their individual features and combinations influence the classification. Specifically, we test PFH (P), PFH+Direction (PDI), PFH+Direction+Distance (PDD), and PFH+Direction+Distance+Betti number (PDDB). The feature vector in each experiment is a concatenation of the involved individual features. Individual features are first normalized. In different experiments, we use different combinations of the normalized features. In other words, if we use a fixed-length feature vector to represent each feature with nonzero values on its corresponding dimensions and all other value zeroed, then the concatenation can also be seen as linearly summing up several features. We tried different weights for this linear combination to achieve good results. Empirically an equal weighting scheme is used for all the classification experiments. For comparison, we also tested two features: absolute height displacement and absolute radial separation used in Fisher and Hanrahan [2010] for the whole dataset, denoted by DIS.

We choose Support Vector Machines (SVMs) [Boser et al. 1992; Cortes and Vapnik 1995] for the classification task because of their simplicity. Specifically, we use the soft margin method [Cortes and Vapnik 1995]. For our multiclass problem, a *one-versus-one* scheme is used. For the kernel, we use a Radial Basis Function (RBF), $K(x, y) = e^{-\theta\|x-y\|^2}$, where x and y are the concatenated feature vectors. To find the best parameter values, we do fivefold cross-validation and hierarchical grid search. We start with coarser grids, then subdivide the best grid for another iteration of search until the improvement of the accuracy falls below 0.001 or it reaches the maximum iteration. Finally, we train the model with the whole training set again using the optimal values and then test it. For implementation, we use libSVM [Chang and Lin 2011].

Evaluation and Comparison. The prediction accuracy is shown in Table II. The first column consists of the dataset. The first row lists features. The cells are filled with prediction accuracies of each experiment. They are calculated by feeding the testing dataset into the trained SVM classifier and the accuracy is the percentage of correctly classified dataout of the whole testing dataset.

Overall, IBS features are more discriminative for one-to-one relationship classification than DIS used in Fisher and Hanrahan [2010]. Note that in the paper [Fisher and Hanrahan 2010], they achieve good retrieval results by using other information such as labelling, but we aim to avoid using it as such data is not always available. The

Table II. Prediction Accuracy

	P	PDI	PDD	PDDB	DIS
S	78.33%	84.07%	85.90%	82.77%	41.25%
C	80.00%	96.00%	92.00%	100.00%	44.00%
S+*C*	76.47%	81.37%	83.33%	84.31%	38.73%

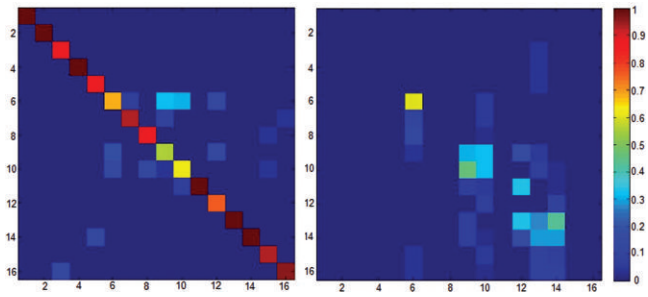

Fig. 9. Confusion matrices of PDDB (left) and DIS (right) on the whole dataset (16 classes). Results are normalized within each column.

results show that DIS does not perform as well as the IBS features under our setting.

For further comparison of features within PDDB, one can see that the PFH descriptor of the IBS already gives good results for this 16-class classification problem. On top of PFH, the direction improves the result. On the complex dataset, the distance is a bit detrimental to the result. This is because distance and direction can contradict one another in this dataset. However, we find that the prediction accuracy of the complex dataset is also 100% when just using PFH and Betti numbers. It reflects the fact that for complex relationships, direction and distances are not discriminative enough. It also shows that Betti numbers provide vital information, especially in classifying complex interactions.

One noteworthy point is that there is a slight decrement of accuracy from PDD and PDDB. One cause is the discrete nature of Betti numbers. Relations with similar geometric features may have different Betti numbers. Also, penetrations between objects can cause the Betti numbers to be calculated incorrectly. Although most scenes in the Stanford Database do not have penetrations, some still exist for geometrically adjacent objects. A preprocessing stage to exclude such penetrations can improve the results.

In Figure 9 we plot the confusion matrix. The values are normalized for each column. The classes that have the lowest prediction accuracies are class 9 and 10 (Figure 9, left). Most of their misclassified instances are in class 6. Shown in Figure 8 and Table I, classes 6, 9, and 10 all have a "side-by-side" relation. But classes 9 and 10 have one object higher than the other. The height difference in some scenes is not big enough to be classified correctly. This is the main source of the prediction error.

Performance. Table III shows the timing and accuracy information of cross-validation and training on the whole dataset ($S+C$). The first row contains best average accuracy of the fivefold cross-validation during the hierarchical grid search. The second row contains the time consumption for the grid search and the last row contains the training time after we find the optimal values of the parameters.

The configuration of the computer where these numbers are calculated is: Intel i7-2760QM CPU, 8GB memory, Windows 7 Professional 64 bit and Matlab R2012a (64 bit).

Table III. Cross-Validation Accuracy and Time Consumption

	PDDB	DIS
Cross-validation accuracy	83.56%	39.36%
Time for cross-validation (secs)	580.56	106.07
Time for training (secs)	0.22	0.06

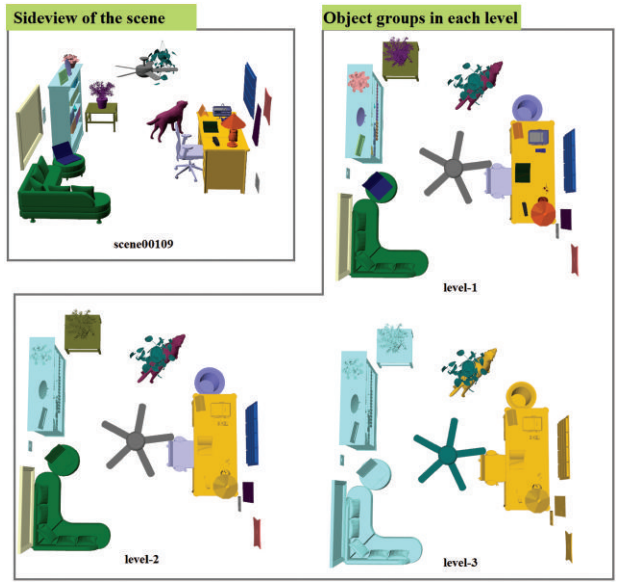

Fig. 10. An example of the hierarchical structure. The objects in the same group are shown in the same color. The 3D models shown in this figure are from the Stanford Scene Database [Fisher et al. 2012].

7.2 Building Hierarchical Structures for 3D Scenes

Experiment. We build hierarchical structures for 130 scenes in the Stanford Database by using Algorithm 1. One example of the results is shown in Figure 10. More results are shown in the supplementary document.

The parameter τ controls the speed of merging when building the hierarchy. A lower τ will merge more nodes together in each round, which means that the number of levels is smaller compared to the structure built with a higher τ. The choice of τ should depend on the nature of the data as well as the higher-level application. For the Stanford Scene Database, we found $\tau = 0.32$ gives visually reasonable structures for most of the scenes. The average number of levels of the hierarchical structure under this τ setting is 2.89.

Evaluation and Analysis. As the scene structure will be used as the input for content-based retrieval, the stability of the hierarchical structure with respect to the parameter setting is important. We evaluate the stability of our HAC algorithm in this experiment, and its benefits for retrieval will be evaluated together with the retrieval results in the next section.

Following the scheme by Goodman and Kruskal [1954], which has been employed to compute the stability of hierarchical algorithms for image segmentation [MacDonald et al. 2006] and speech classification [Smith and Dubes 1980], we assess the stability of generating the hierarchical structure using a consistency measure (denoted here by γ) of how the merges happen under different parameter settings. Briefly speaking, γ ($-1 \leq \gamma \leq 1$) is the difference

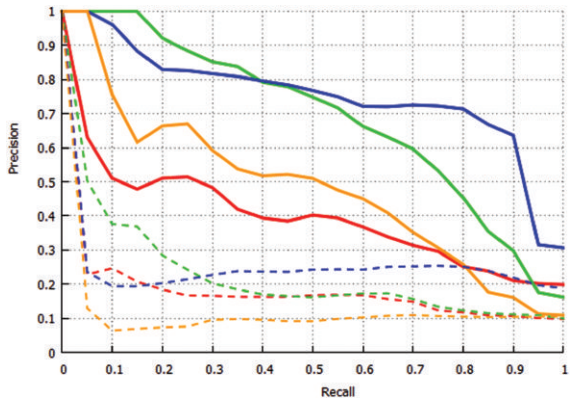

Fig. 11. The solid lines show the precision-recall curve for our algorithm on four test scenes. The dashed lines show the precision-recall curve when using the DIS feature. The IDS of the four queries in the Stanford Scene Database are: scene00050-object8(red), scene00118-object14(green), scene00109-object14(blue), scene00087-object37(yellow).

between the probability that the "correct" and the "wrong" order occurs. See Appendix C for the details of computing γ.

In our algorithm, the grouping threshold τ is the main parameter. We check the stability of the structure under different values of τ. For each scene, we compute five hierarchies by varying τ from 0.1 to 0.5 with an interval of 0.1. Then, we compute the γ for every pair within the five hierarchies. Finally, we use the mean of the γ, denoted here by $\hat{\gamma}$, as the indicator of the stability of our method.

We compute $\hat{\gamma}$ for 130 scenes, and the average $\hat{\gamma}$ is 0.989, with the lowest stability 0.779 (scene00042), which means the stability of our algorithm is very high.

7.3 Content-Based Retrieval

Experiment. We tested the capability of our algorithm for content-based scene retrieval using a dataset that consists of 130 scenes, which come from the Stanford Scene Database. We first calculate the hierarchical structures by the techniques explained in Section 5. The user then selects any single object from a scene as a query. Then the system returns objects in any scene which has similar spatial relations with their surrounding objects.

Evaluation and Comparison. The retrieval results are shown in Figure 14 and Figure 15. It can be observed that our system returns contextually similar results without using the geometry feature or label of each individual model. More retrieval results are presented in the supplementary document.

In order to evaluate our system, we prepared manually labelled data, which is produced as follows. Four query objects were selected from the Stanford Scene Database, and then for each query object an additional 500 objects are randomly selected from the database. The set of 500 objects were shown to the user with the scene in random order, and the user was asked if the object has similar spatial relations with the surrounding objects as compared to the corresponding query object. We label the spatial relations as similar if more then half of the users think they are similar. Ten users including students and staff from different schools in the university took part in the user study.

For quantitative evaluation of the results, the precision and recall curves (p-r curve) are drawn based on the search results and the ground-truth (manually labelled) data. In Figure 11, we present the curves based on our features (solid lines) and DIS (dashed lines).

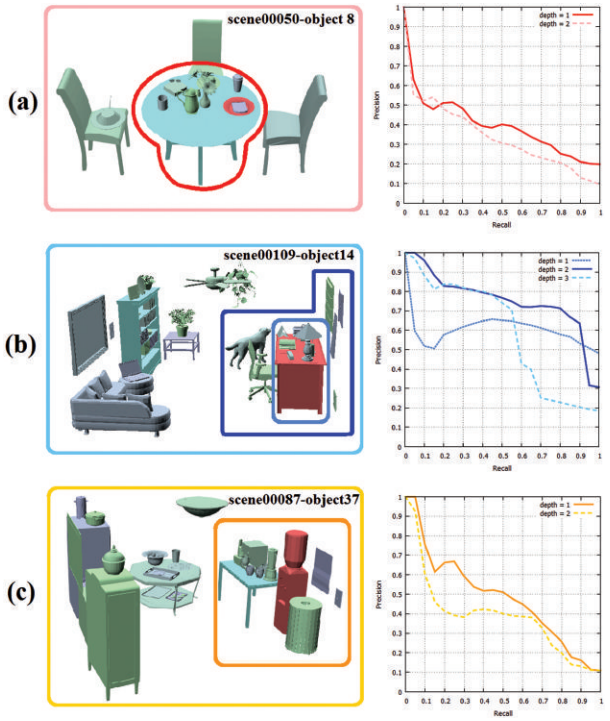

Fig. 12. Scene regions and precision-recall curves for different d_{depths}. Each row corresponds to one query we used for the user study. The red object in the scene is the query object. The boundary lines in the left column show the regions corresponding to the p-r curves of the same color. The 3D models shown in this figure are from the Stanford Scene Database [Fisher et al. 2012].

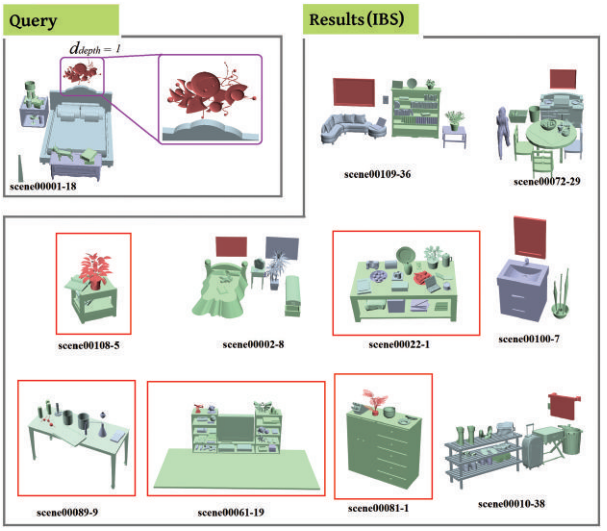

Fig. 13. Failure case. The results with red bounding box are not satisfactory, as they are "object on table" while the query is "object hanging above another object". The 3D models shown in this figure are from the Stanford Scene Database [Fisher et al. 2012].

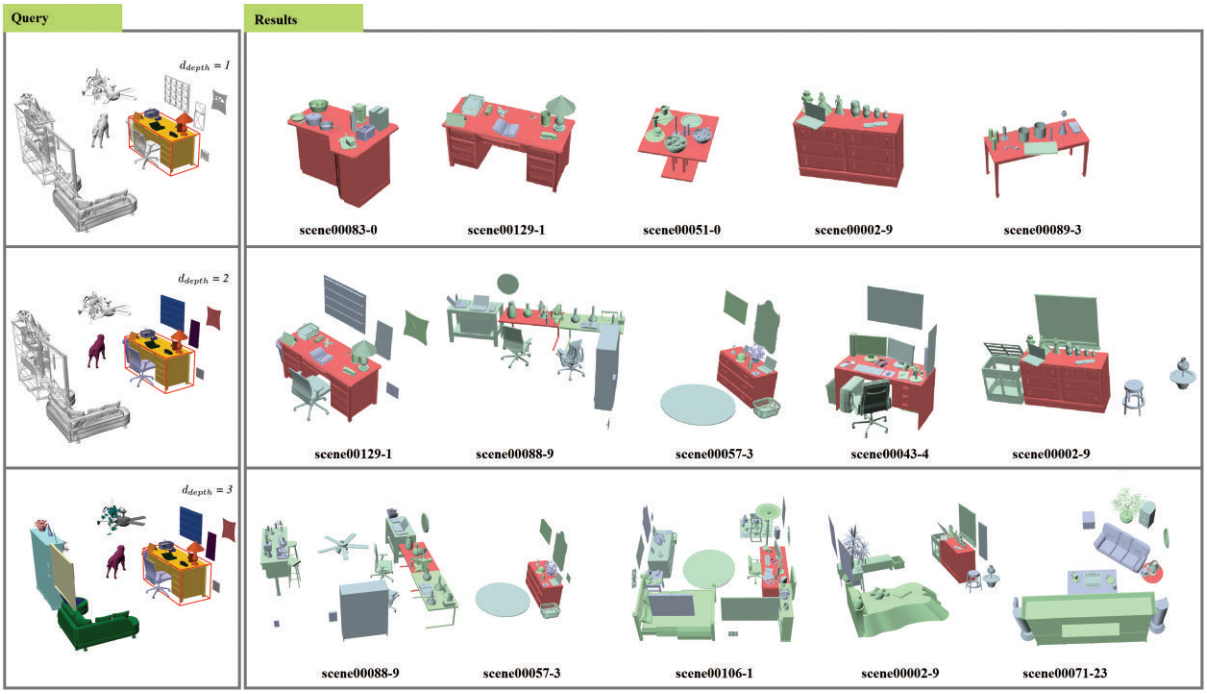

Fig. 14. Retrieval results by using IBS features. (left) In the query scene, the object with a bounding box (the desk) is the query object. We show the other objects within the search depth in color while leaving the rest of the scene in grey. (right) The resulting scenes from left to right are in order of similarity. The red object is the retrieved object with a similar context to the desk. To show the context clearly, we also render other objects within the search depth with slightly different pale green/blue color in the result scenes. The 3D scenes are from the Stanford Scene Database [Fisher et al. 2012].

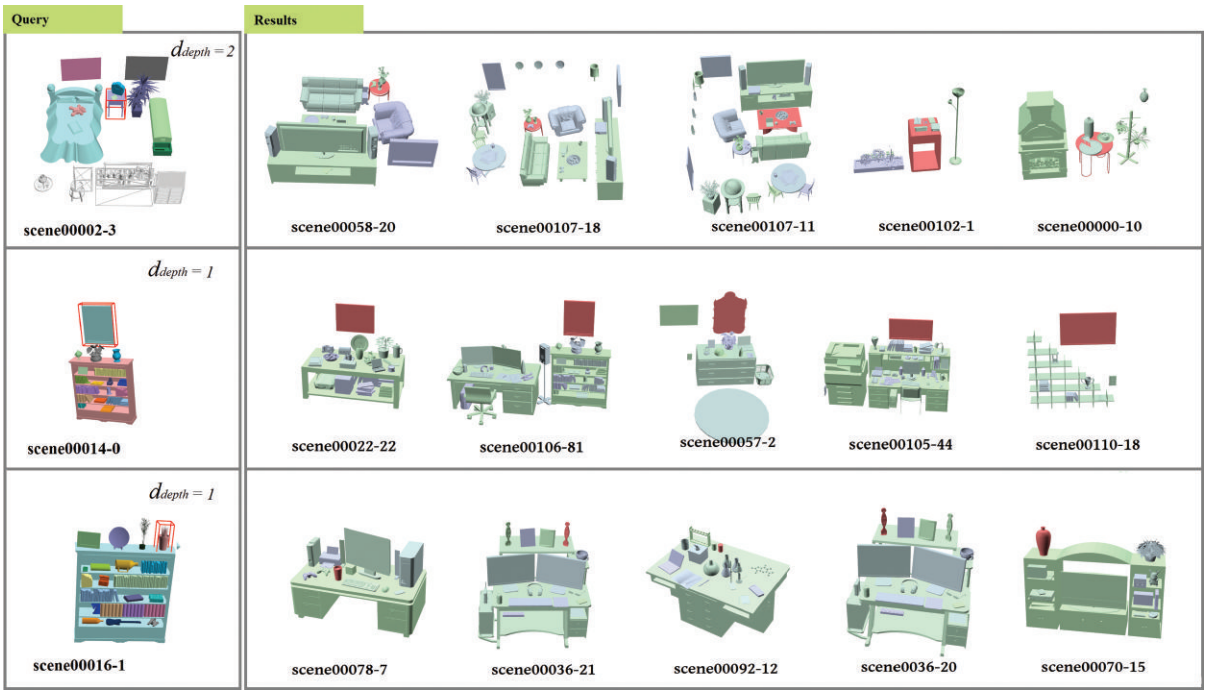

Fig. 15. More retrieval results by using IBS features. The 3D scenes are from the Stanford Scene Database [Fisher et al. 2012].

It can be observed that all the solid lines show significantly higher precision than the dashed lines of the same color, which indicates that our features outperform the DIS feature. Our algorithm returns 50% of similar results with a precision of at least 40%, meaning that at least 1 in every 2.5 resulting scenes is desirable.

We now show the results of analysing the extent of the community ($d_{\mathbf{depth}}$) the users take into account when comparing the similarity of relationships. Figure 12 shows the (p-r curves) for three query scenes with different $d_{\mathbf{depth}}$ values. In Figure 12(a), the p-r curve for $d_{\mathbf{depth}} = 1$ gives slightly higher precision than $d_{\mathbf{depth}} = 2$, which means that the users tend to pay more attention to the immediate neighbors of the "plate" (which are the toast, the table, and the other objects on the table) than those in the extended community. The results in Figure 12(c) are similar to (a), but there is a larger difference in precision between the two depths. In Figure 12(b), $d_{\mathbf{depth}} = 2$ gives the highest precision, which means the users also consider the objects in the extended communities when evaluating the similarity of the spatial relations. We can assume that factors such as the scale of the scene and the density of objects affect the perception of the neighborhood. The scene in Figure 12(b) has more objects densely located around the desk compared with the desk in Figure 12(c) and the scale is larger than the scene shown in Figure 12(a).

Figure 13 shows a failure case for our method on the Stanford Scene Database, with half of the top ten results (object on table) not matching the query object (decoration hanging above the bed). As the bottom of the decoration is almost in contact with the bedhead, it is similar to the "one object on another" examples in terms of spatial relationship. This can be a typical failure case of retrieval results not being consistent with the user's intuition due to the fact that we do not take into account any geometry information of the individual objects nor their semantic labels. The failure cases are mostly removed from the retrieval results when the decoration is lifted a little bit so that the relationships between the decoration and the bedhead are not misunderstood as contacts.

8. CONCLUSION AND DISCUSSIONS

In this article we have proposed a new descriptor called the Interaction Bisector Surface to capture spatial relationships between 3D objects. The rich information among numerous types of object relations is well contained in the IBS. Its capacity for describing these relationships lies in its topological and geometric features. Betti numbers enable us to recognize very sophisticated relations while the distance, the direction, and the shape of the IBS further indicate nuances at a finer level. The IBS is the cornerstone of the research and it provides a new perspective to model spatial relationships. In addition, we have proposed an automated mechanism to understand the structure of big scenes consisting of a large number of objects. Knowing the hierarchy of big scenes is crucial for applications such as content-based relationship retrieval. Because of the nature of the IBS, the calculation naturally rules out relationships between objects that are too far from each other or have too many objects in between. The structure of the IBS segments the scene into groups at every level. Therefore, we are able to automatically build up a hierarchical structure that has meaningful geographical groups. We also propose similarity metrics based on the IBS that effectively distinguish different types of spatial relationships between objects or object groups. These metrics are equipped with the scene hierarchy so that comparison can be made between objects based on their contexts. Finally, we also show how the features, metrics, and algorithms based on the IBS can be applied to solve practical problems.

Although our approach to computing the IBS is a heuristic, it is a good compromise in terms of computational cost and precision. For the computation of the IBS, we use a sampling-based approach in which points are sampled on the object surfaces and then the Quickhull algorithm [Barber et al. 1996] is used. This is a heuristic approach that does not guarantee the exact topology and geometry of the resulting medial axis. This could be an issue if we need to match the homotopy of the IBS. In order to avoid such confusion, we use abstract topology features (the first and second Betti numbers) whose values are less influenced by the accuracy of the IBS. Although the Betti numbers can be affected by topological noise such as holes, this is less likely to appear in the bisector surfaces as they are defined between distinct separate objects. Also, the geometric features of the IBS are statistical values that are less influenced by the accuracy of the IBS. In addition, exact methods to compute the medial axis [Culver et al. 1999] and bisector surface [Elber and Kim 1997] are not practical to be applied in a set of high-resolution meshes. In summary, our method makes use of features that are less computationally costly and less affected by parameter values.

Limitations. Although the IBS is good for identifying spatial relationships, the computational cost is higher compared with other simple features used in Fisher and Hanrahan [2010]. We believe that it is a fair trade-off between precision and performance. The method can be easily parallelised, and can greatly benefit from implementing on multicore systems. Second, the discriminative power of the IBS deteriorates when the distance between objects increases. When there are just two objects in the scene and they are far apart from each other, we suspect that IBS features can be replaced by simpler features used in Fisher and Hanrahan [2010] such as the height displacement and radial separation. Lastly, as we focus on relationship understanding in this research, individual geometry plays a less important role. This is different from previous works. Hence, for applications such as retrieval, it might cause confusion when the user tries to retrieve scenes not only with similar relationships but also with similar geometries.

Future Work. We believe that the potential of using the IBS for spatial relation representation has not been fully explored. In the future, one possible direction is to use it for comparing two scenes. This can be useful for whole scene retrieval. Another promising direction is to further exploit IBS features and explore along the time domain. By observing the feature variations on the time dimension, we might be able to understand, recognize, and classify animated scenes. At the same time, by adding human knowledge via learning algorithms, we can pursue a semantic understanding of the relationships between motions and environments.

APPENDIXES

A. SAMPLING

Here we describe how we sample points on the IBS where we compute the geometric features. This is done by calculating the weights of the triangles composing the IBS. First, we define a direction angle α for each triangle. A triangle T on the IBS is equidistant to sample points, s_b and s_c on o_1 and o_2 respectively. Let us define a vector, v, from the center of T (defined as s_a) to s_b and a normal n of T pointing towards the side of o_1. α (Figure 16(a)) is the angle between v and n. Note that this angle is the same if we compute it between T and o_2 because the normal is flipped in that case. The larger the angle, the higher the chance that the sample point is far away from the objects defining it and less informative

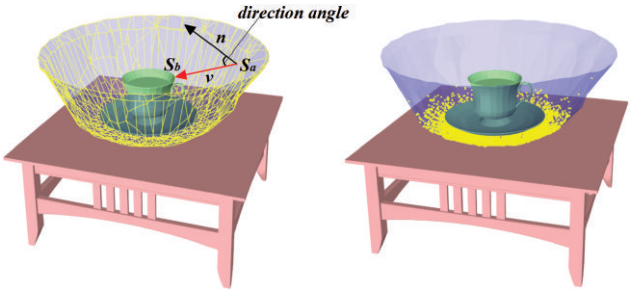

Fig. 16. (left) The direction angle of a polygon on the IBS. (right) The sampling result.

about the interaction. We compute a weight $W(T)$

$$W(T) = W_{\text{area}}(T) \times W_{\text{scene-distance}}(T) \times W_{\text{angle}}(T), \tag{13}$$

where $W_{\text{area}}(T)$ is the area of triangle T and W_{angle} is computed as

$$W_{\text{angle}} = \begin{cases} 1 - \frac{\alpha}{45^\circ} & \text{if } \alpha < 45^\circ \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

$W_{\text{scene-distance}}$ is computed by

$$W_{\text{scene-distance}} = \left(1 - \frac{d}{D}\right)^n, \tag{15}$$

where d is the distance between s_a and s_b (or s_c). $D = d_{\text{diag}}/2$ where d_{diag} is the length of the diagonal of the bounding box of the whole scene. We empirically set n equal to 20. We then normalized W(T) for all the triangles. Next, we set up a target number for all triangles and the final target number for each triangle is the target number times the triangle's weight. Finally, we use the final target numbers to do random sampling on every triangle. Figure 16(b) shows the result of the weighted sampling.

B. POINT FEATURE HISTOGRAM (PFH)

PFH is a feature for encoding the geometry of a point-cloud. Given two points (Figure 17), p_1 and p_2, with normals n_1 and n_2, three unit vectors (u, v and w) are built by the following procedure: (1) u is the normal vector of p_1, (2) $v = u \times \frac{p_2 - p_1}{d}$, (3) $w = u \times v$. $d = \|p_2 - p_1\|_2$. Then the difference between n_1 and n_2 is represented by three angles (α, θ, ϕ) which are computed as: $\alpha = v \cdot n_2$, $\phi = u \cdot \frac{p_2 - p_1}{d}$, $\theta = arctan(w \cdot n_2, u \cdot n_2)$. The triplet $< \alpha, \phi, \theta >$ is computed for each pair of points in the k-neighborhood, and are binned into a histogram. Usually each angle is divided into b equal parts, and the triplet can form a b^3-size histogram in which each bin represents a unit combination of the value ranges for each value. In our case, we compute the triplet for each pair of points in the point-cloud which is a set of samples computed by using the method described previously. We set $b = 5$. So the PFH feature we use is a 125-length vector.

C. STABILITY

Here we explain the definition of γ introduced by Goodman and Kruskal [1954]. Consider two hierarchical structures for the same data, h_1 and h_2, and two pairs of elements of this data $p_i = (x_{i1}, x_{i2})$ and $p_j = (x_{j1}, x_{j2})$. The rank $r(h, p)$ is defined as the level at which the two elements of pair p first appear in the same cluster in

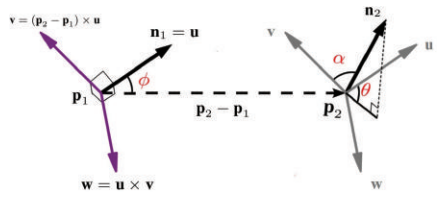

Fig. 17. PFH angles.

hierarchy h. Over all the pairs of elements in the data, set

$$\begin{aligned} \Pi_s = & \\ & P_r\{(r(h_1, p_i) < r(h_1, p_j) \wedge r(h_2, p_i) < r(h_2, p_j)) \\ & \vee (r(h_1, p_i) > r(h_1, p_j) \wedge r(h_2, p_i) > r(h_2, p_j))\} \\ \Pi_d = & \\ & P_r\{(r(h_1, p_i) < r(h_1, p_j) \wedge r(h_2, p_i) > r(h_2, p_j)) \\ & \vee (r(h_1, p_i) > r(h_1, p_j) \wedge r(h_2, p_i) < r(h_2, p_j))\} \\ \Pi_t = & \\ & P_r\{(r(h_1, p_i) = r(h_1, p_j)) \vee (r(h_1, p_i) = r(h_1, p_j))\}. \end{aligned} \tag{16}$$

Then

$$\gamma = \frac{\Pi_s - \Pi_d}{1 - \Pi_t}. \tag{17}$$

γ measures the difference between the probabilities of "right order" and "wrong order". In other words γ shows how much more probable it is to get the same rather than different orders in two hierarchies. It ranges from -1 for inconsistency to 1 for consistency.

ACKNOWLEDGMENTS

We thank the anonymous reviews for their constructive comments. We also thank Rami Ali Al-ashqar for his help in preparing 3D models used in Figure 1 and Figure 5, Shin Yoshizawa for the discussions and the test users for their help in evaluation of our system. The scene and object data are provided courtesy of the Stanford Scene Database [Fisher et al. 2012] and the Princeton Shape Benchmark [Shilane et al. 2004].

REFERENCES

L. A. Alexandre. 2012. 3D descriptors for object and category recognition: A comparative evaluation. In *Proceedings of the IEEE Workshop on Color-Depth Camera Fusion in Robotics at the RSJ International Conference on Intelligent Robots and Systems (IROS'12).*

N. Amenta, S. Choi, and R. K. Kolluri. 2001. The power crust. In *Proceedings of the 6th ACM Symposium on Solid Modeling and Applications.* 249–266.

D. Attali, J.-D. Boissonnat, and H. Edelsbrunner. 2009. Stability and computation of medial axes - A state-of-the-art report. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, T. Miller, B. Hamann, and R. D. Russell, Eds., Springer, 109–125.

C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* 22, 4, 469–483.

S. Belongie, J. Malik, and J. Puzicha. 2002. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 4, 509–522.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory.* ACM Press, New York, 144–152.

C.-C. Chang and C.-J. Lin. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2, 3, 27:1–27:27. http://www.csie.ntu.edu.tw/cjlin/libsvm.

M.-C. Chang and B. B. Kimia. 2011. Measuring 3d shape similarity by graph-based matching of the medial scaffolds. *Comput. Vis. Image Understand.* 115, 5, 707–720.

C. Cortes and V. Vapnik. 1995. Support-vector networks. *Mach. Learn.* 20, 3, 273–297.

T. Culver, J. Keyser, and D. Manocha. 1999. Accurate computation of the medial axis of a polyhedron. In *Proceedings of the 5th ACM Symposium on Solid Modeling and Applications*. 179–190.

C. J. A. Delfinado and H. Edelsbrunner. 1995. An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere. *Comput. Aided Geom. Des.* 12, 7, 771–784.

G. Elber and M.-S. Kim. 1997. The bisector surface of freeform rational space curves. In *Proceedings of the 13th Annual Symposium on Computational Geometry*. ACM Press, New York, 473–474.

M. Fisher and P. Hanrahan. 2010. Context-based search for 3d models. *ACM Trans. Graph.* 29, 6, 182.

M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan. 2012. Example-based synthesis of 3d object arrangements. *ACM Trans. Graph.* 31, 6, 135.

M. Fisher, M. Savva, and P. Hanrahan. 2011. Characterizing structural relationships in scenes using graph kernels. *ACM Trans. Graph.* 30, 4.

S. Giannarou and T. Stathaki. 2007. Object identification in complex scenes using shape context descriptors and multi-stage clustering. In *Proceedings of the 15th International Conference on Digital Signal Processing.* 244–247.

E. B. Goldstein. 2010. Sensation and perception. www.CengageBrain.com.

L. A. Goodman and W. H. Kruskal. 1954. Measures of association for cross classifications. *J. Amer. Statist. Assoc.* 67, 338, 732–764.

Z. Harchaoui and F. Bach. 2007. Image classification with segmentation graph kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07).* 1–8.

J. Hartung, G. Knapp, and B. K. Sinha. 2011. *Statistical Meta-Analysis with Applications*, Vol. 738. http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470290897.htm.

T. Hastie, R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning*. Springer, New York.

T. Imai. 1996. A topology oriented algorithm for the voronoi diagram of polygons. In *Proceedings of the 8th Canadian Conference on Computational Geometry*. Carleton University Press, 107–112.

E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun. 2012. A probabilistic model for component-based shape synthesis. *ACM Trans. Graph.* 31, 4, 55.

C.-M. Kim, C. I. Won, Y. Cho, D. Kim, S. Lee, J. Bhak, and D.-S. Kim. 2006. Interaction interfaces in proteins via the voronoi diagram of atoms. *Comput.-Aided Des.* 38, 11, 1192–1204.

D. MacDonald, J. Lang, and M. McAllister. 2006. Evaluation of colour image segmentation hierarchies. In *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision (CRV'06).* 27.

W. Massey. 1991. *A Basic Course in Algebraic Topology*. Vol. 127, Springer.

L. Paraboschi, S. Biasotti, and B. Falcidieno. 2007. Comparing sets of 3d digital shapes through topological structures. In *Graph-Based Representations in Pattern Recognition*, F. Escolano and M. Vento, Eds., Lecture Notes in Computer Science, vol. 4538, Springer, 114–125.

A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. 2007. Objects in context. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV'07).*

R. Rusu, Z. Marton, N. Blodow, and M. Beetz. 2008a. Learning informative point classes for the acquisition of object model maps. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV'08).* 643–650.

R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. 2008b. Persistent point feature histograms for 3d point clouds. *Intell. Auton. Syst.* 10, Ias-10, 119.

T. B. Sebastian, P. N. Klein, and B. B. Kimia. 2001. Recognition of shapes by editing shock graphs. In *Proceedings of the 5th IEEE International Conference on Computer Vision (ICCV'01).* 755–762.

L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, and H. Zhang. 2010. Contextual part analogies in 3d objects. *Int. J. Comput. Vis.* 89, 2–3, 309–326.

E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. 1995. Computation of the medial axis transform of 3-d polyhedra. In *Proceedings of the 3rd ACM Symposium on Solid Modeling and Applications.* 187–200.

P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. 2004. The princeton shape benchmark. In *Proceedings of the IEEE International Conference on Shape Modeling Applications*. 167–178.

S. P. Smith and R. Dubes. 1980. Stability of a hierarchical clustering. *Pattern Recogn.* 12, 3, 177–187.

A. Sud, M. Foskey, and D. Manocha. 2007. Homotopy-preserving medial axis simplification. *Int. J. Comput. Geom. Appl.* 17, 05, 423–451.

J. K. T. Tang, J. C. P. Chan, H. Leung, and T. Komura. 2012. Retrieval of interactions by abstraction of spacetime relationships. *Comput. Graph. Forum* 31, 2.

O. van Kaick, K. Xu, H. Zhang, Y. Wang, S. Sun, A. Shamir, and D. Cohen-Or. 2013a. Co-hierarchical analysis of shape structures. *ACM Trans. Graph.* 32, 4, 69.

O. van Kaick, H. Zhang, and G. Hamarneh. 2013b. Bilateral maps for partial matching. *Comput. Graph. Forum* 32, 6, 189–200.

Y. Wang, K. Xu, J. Li, H. Zhang, A. Shamir, L. Liu, Z.-Q. Cheng, and Y. Xiong. 2011. Symmetry hierarchy of man-made objects. *Comput. Graph. Forum* 30, 2, 287–296.

L.-F. Yu, S.-K. Yeung, C.-K. Tang, D. Terzopoulos, T. F. Chan, and S. J. Osher. 2011. Make it home: Automatic optimization of furniture arrangement. *ACM Trans. Graph.* 30, 4, 86:1–86:12.

Y. Zheng, D. Cohen-Or, and N. J. Mitra. 2013a. Smart variations: Functional substructures for part compatibility. *Comput. Graph. Forum* 32, 2, 195–204.

Y. Zheng, C.-L. Tai, E. Zhang, and P. Xu. 2013b. Pairwise harmonics for shape analysis. *IEEE Trans. Vis. Comput. Graph.* 19, 7, 1172–1184.

Received October 2013; accepted December 2013