Computer Aided Geometric Design ••• (••••) •••–•••

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Cluttered indoor scene modeling via functional part-guided graph matching

Jun Wang, Qian Xie, Yabin Xu, Laishui Zhou, Nan Ye *

College of Mechanical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Cluttered indoor scenes
Scene modeling
Graph matching
Raw point clouds

ABSTRACT

We propose an automatic method for fast reconstruction of indoor scenes from raw point scans, which is a fairly challenging problem due to the restricted accessibility and the cluttered space for indoor environment. We first detect and remove points representing the ground, walls and ceiling from the input data and cluster the remaining points into different groups, referred to as sub-scenes. Our approach abstracts the sub-scenes with geometric primitives, and accordingly constructs the topology graphs with structural attributes based on the functional parts of objects (namely, *anchors*). To decompose sub-scenes into individual indoor objects, we devise an *anchor-guided* subgraph matching algorithm which leverages template graphs to partition the graphs into subgraphs (i.e., individual objects), which is capable of handling arbitrarily oriented objects within scenes. Subsequently, we present a data-driven approach to model individual objects, which is particularly formulated as a model instance recognition problem. A Randomized Decision Forest (RDF) is introduced to achieve robust recognition on decomposed indoor objects with raw point data. We further exploit template fitting to generate the geometrically faithful model to the input indoor scene. We visually and quantitatively evaluate the performance of our framework on a variety of synthetic and raw scans, which comprehensively demonstrates the efficiency and robustness of our reconstruction method on raw scanned point clouds, even in the presence of noise and heavy occlusions.

1. Introduction

3D scene processing for both indoor and outdoor environments has been an important research problem in computer vision and graphics communities. Meanwhile, recent advances in scanning technology greatly improve the acquisition of point clouds both in speed and accuracy, which also renders point cloud processing receive increasing attention recently. Despite the advances in acquisition technology, the captured point cloud often suffers from severe noise and outliers, making the reconstruction of indoor models with faithful geometry and topology from such data rather arduous. In addition, the significant difficulty in indoor scans is the presence of heavy occlusions as the interior environments are usually relatively narrow and cluttered, even when multiple scan stations are set. Consequently, to automatically and efficiently reconstruct indoor scenes is particularly challenging, especially for the cluttered indoor environments with defect-laden, raw point clouds.

* Corresponding author.
E-mail address: isaac_yn@126.com (N. Ye).

http://dx.doi.org/10.1016/j.cagd.2016.02.012
0167-8396/

Recent works (Nan et al., 2012; Kim et al., 2012; Shao et al., 2012) take advantage of the learning-based technique to infer scene segmentation, and then detect and replace multiple instances of the object within the indoor scene using for instance partial matching so as to achieve scene reconstruction. These methods are able to obtain promising modeling results even in the presence of cluttered scenes thanks to the data-driven characteristics. However, they mostly assume the indoor objects are placed always with the upward direction in terms of the ground floor. Once the assumption is invalid, they may fail in segmentation and thus reconstruction. In addition, a certain level of interactions are required during reconstruction. Moreover, when the level of data imperfection becomes high, they would perform relatively poor on the indoor scenes, due to typical clutter, missing regions and noise. The goal of our work is to automatically reconstruct cluttered indoor scenes with arbitrarily oriented objects from raw point scans.

We propose an automatic method for fast modeling raw point data captured from cluttered indoor scenes. As observed, most man-made objects of indoor scenes are assembled by parts corresponding to primitive shapes. Accordingly, we fit the point data of sub-scenes with primitive shapes to obtain a concise representation. Moreover, we notice that a man-made object generally contains at least one functional structure (namely, *anchor*), which topologically relates to the other structures (i.e., primitives) of the object. Therefore, it is reasonable to construct a topology graph, formed by connecting the anchor to the other primitives, to represent the sub-scene. Thus, we abstract sub-scenes with the topology graphs with attributes, which adequately convey the geometrical and structural information of the sub-scenes. To analyze the graphs, rather than the original point data, we render an efficient and effective way to decompose individual objects from sub-scenes.

A sub-scene usually consists of several indoor objects. Analogously, the attributed graph comprises several subgraphs. Accordingly, we formulate sub-scene decomposition as a graph matching problem. Collecting a database of man-made indoor shapes, we construct the topology graphs for them, referred to as graph templates. We use the graph templates to partition the topology graphs into subgraphs, each of which corresponds to an individual indoor object. By constructing a matching similarity function, we find the correspondences between graphs by solving a maximization problem. By reducing the computation complexity during optimization, we minimize the number of similarity comparisons between graphs to sparsify the similarity matrix. Particularly, we only measure the similarity between graph nodes sharing the same primitive type. Moreover, we establish candidate graph matches only starting from anchor nodes, and then restrict the comparisons only between edges induced from the corresponding anchor nodes. In return, the number of similarity comparisons is significantly decreased, and our graph matching can be accomplished efficiently. As a result, the individual objects are decomposed from the sub-scene, while the category of each object is determined as well.

To reconstruct individual indoor objects, we present a data-driven modeling method based on the shape database. Specifically, we formulate object modeling as a model instance recognition problem. To this end, a Randomized Decision Forest (RDF) is introduced to solve this recognition problem. We define a set of shape features for learning of the RDF classifier. The features are discriminative and insensitive to noise, outliers and data sparsity. We then exploit template fitting to compute the transformations from database models to scanned objects, which are applied to achieve geometrically faithful reconstruction from the input indoor scene.

Overall, our contributions are as follows:

1. We propose a functional part-guided modeling method for cluttered indoor scenes with raw scans. It proceeds automatically and results in high fidelity to input scenes.
2. We design an anchor-guided graph matching algorithm for scene decomposition, which is capable of handling scenes with objects arbitrarily oriented.
3. We devise a data-driven approach for object modeling based on randomized decision forest, which is robust to data imperfections.

1.1. Related work

There is an extensive amount of literature on scene modeling, ranging from image-based (Saxena et al., 2009; Xiao et al., 2010; Quattoni and Torralba, 2009), RGBD-based (Izadi et al., 2011; Bo et al., 2013) to 3D point-based approaches (Frome et al., 2004; Rusu et al., 2008; Schnabel et al., 2008; Nan et al., 2010; Shen et al., 2011; Koppula et al., 2011; Kim et al., 2012). Here, we mainly focus on the most work to ours, particularly for those regarding scene modeling, scene reconstruction and object matching.

Scene modeling. The procedural modeling of large-scale scenes has gained much attention in recent years (Parish and Müller, 2001; Wonka et al., 2003; Müller et al., 2006; Musialski et al., 2013). With the significant advances in 3D scanning recently, increasing research work has been focusing on scene reconstruction directly from 3D scan data (Shao et al., 2012; Nan et al., 2012; Lin et al., 2013; Arikan et al., 2013; Mattausch et al., 2014). Nan et al. (2012) used the repetition characteristic to model urban facades. It requires a moderate amount of user interactions to reveal the architectural structures as repetitive patterns. Kim et al. (2012) utilized object repeatability to reconstruct indoor scenes with basic primitives. Outdoor scenes, e.g. building facades, usually exhibit symmetry and repetitions, while indoor scenes are generally cluttered and objects are arranged randomly. We concentrate on cluttered indoor scenes without any assumption of repeatability and regularity.

Indoor scene reconstruction. Indoor scene reconstruction has also attracted plenty of research interest recently (Du et al., 2011; Izadi et al., 2011; Ren et al., 2012; Henry et al., 2014). From a scene database, Chen et al. (2014) learned the contextual

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

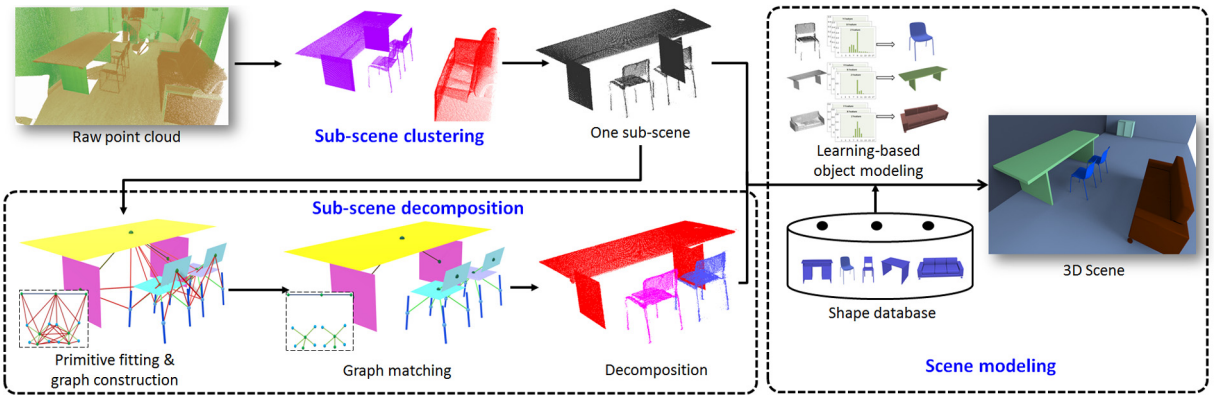

Fig. 1. Overview of our modeling method. It proceeds as sub-scene clustering, decomposition and learning-based object modeling.

information to perform RGB-D data reconstruction. Oesau et al. (2014) proposed an automatic reconstruction of permanent structures of indoor scenes. This reconstruction work mainly focuses on the primary structures, such as walls, floors and ceilings, while our method copes with the relatively complicated indoor objects. Shao et al. (2012) presented an interactive semantic modeling approach for indoor scenes. Nan et al. (2012) introduced a search-classify scheme for indoor scene modeling, which assumes that all objects are placed upward on the ground floor. In contrast, our method proceeds without any interactions. Moreover, ours is pose-invariant and independent on the orientations of indoor objects.

Object matching. As the topology of 3D model is an important feature of the shape, there exist some matching approaches using topology graphs (Hilaga et al., 2001; Tal and Zuckerberger, 2007; Schnabel et al., 2008; Spina et al., 2014). Tal and Zuckerberger (2007) decomposed each object into its "meaningful" components at the deep concavities of the object, and constructed the connectivity graph based on the components. These methods are working on individual models represented with triangular meshes, which are usually noise-free compared with raw point scans. Moreover, compared to their models, our inputs, i.e., cluttered scenes, are more complex. Schnabel et al. (2008) proposed a shape recognition algorithm based on graph matching over building facades with 3D laser scans. It could be fairly time-consuming when the number of the objects within the facade is big. Moreover, the matching would fail if the actual structure graph does not exactly match one of pre-defined configurations. In contrast, our topology graph is guided by the functional parts of objects, which significantly reduces the graph matching complexity. Moreover, we are able to handle the graphs with outliers as demonstrated in Fig. 10.

2. Overview

The input to our method is the raw scan of indoor scene, represented as unorganized point clouds. Generally, man-made indoor object can be decomposed into parts, which can be approximately represented by geometric primitives, such as plane, cylinder, etc. Accordingly, we propose a novel graph-based framework for indoor scene understanding and modeling. Given the raw point data of an indoor scene, our goal is to *automatically* pose 3D models accordingly to create a faithful scene reconstruction. Fig. 1 presents the overview of our proposed algorithm, which essentially consists of three main stages.

Sub-scene clustering. In Section 3, we extract a set of point groups as sub-scenes from the whole indoor scene, which is formulated as a cluster-based partition problem. We first detect and discard points representing the ground, walls and ceiling from the input data, followed by clustering the remaining point cloud into different groups based on region growing technique. Sub-scenes are then clustered from the input scene.

Sub-scene decomposition. For each sub-scene, we decompose it into a few semantic entities (i.e., individual objects) in Section 4. Given the point data of a sub-scene, we fit the points with geometric primitives to obtain a shape abstraction of the sub-scene. With these primitives, we treat the functional parts of objects as the anchors, and thereby construct the topology graph of the sub-scene with attributes.

Subsequently, we propose an effective algorithm to decompose sub-scenes into individual objects, which is formulated as a subgraph matching problem. Based on a database of models, we construct the corresponding topology graphs for them, namely, graph templates. On this basis, we design an anchor-based graph matching algorithm to partition the topology graph into subgraphs, each of which corresponds to an indoor object. As a result, all individual indoor objects are decomposed and classified from the sub-scene.

Scene modeling. We present a data-driven modeling algorithm for individual indoor objects in Section 5. Particularly, we formulate object modeling as a model instance recognition problem. It aims to retrieve the most similar object from the shape database to each individual object. We then exploit the template fitting technique to compute the transformations from database models to the scanned objects, and thus produce geometrically faithful reconstruction of the input indoor scene.

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

3. Sub-scene clustering

In this section, we present a clustering approach to partition the indoor scene into point groups, i.e., sub-scenes. The points within each group are spatially close to each other, while each group is relatively far from the others.

Outliers removal. In our context, outliers refer to as the points standing for the walls, ground and ceiling. The outlier points are first discarded from the input point cloud, since they would be influential on the following object detection. To this end, we take advantage of Hough Transform to detect the planar patches from the input data. As observed, the walls, ground or ceiling usually form relatively large planar regions. Accordingly, the detected patches with relatively large areas would be regarded as outliers and thus removed. As a result, only the points of indoor objects are retained, which are of our interest.

Sub-scene clustering. The goal of this stage is to partition the scene into sub-scenes. We define a sub-scene as a combination of objects which are spatially close to each other, and hence a sub-scene may consist of one or more than one object. For each sub-scene, the associated points are geometrically close, while it is spatially away from the other sub-scenes. Accordingly, we design a clustering method based on the spatial distance criterium. Given a set of points P drawn from a union of k sub-scenes $S_1, S_2, \ldots, S_k$, respectively, we are to partition all points into their corresponding sub-scenes, i.e.,

$$P = P_1 \cup P_2 \cup \ldots \cup P_k, \quad s.t.\, \forall i, j,\ \|P_i - P_j\| > \epsilon \tag{1}$$

where P_i is a set of points from S_i, ϵ is a given distance threshold, and $\|P_i - P_j\|$ stands for the minimal distance between the pairs of points from P_i and P_j, respectively.

Here, we exploit the region growing scheme to solve the clustering problem. In particular, randomly choosing an unclustered point as a seed, we add it into the current region and search the closest point within its nearest neighborhood, followed by calculating the geometric distance between them. If the distance is smaller than ϵ, we add the neighboring point into the current region and update it as the new seed. The above growing procedure is repeated iteratively until no more point can be added into the current region, which is then regarded as a cluster. Therefore, all points in this cluster form one sub-scene. In such a way, we are able to segment the whole point set P into $P_1, P_2, \ldots, P_k$. Therefore, the corresponding sub-scenes, $S_1, S_2, \ldots, S_k$ are obtained.

4. Subgraph matching based sub-scene decomposition

In this section, we present a decomposition algorithm to extract and classify individual objects from each sub-scene, which is converted as a subgraph matching problem. In particular, we fit the sub-scene point data with primitives. Note that our method can be easily extended to add more primitive shapes like sphere, cone and etc. On this basis, we construct the topology graph of the sub-scene guided by the functional parts of objects, referred to as *anchors*. Given a shape repository, we decompose the graph into subgraphs based on our proposed anchor-guided graph matching algorithm. As a consequence, all individual objects corresponding to subgraphs can be obtained within the sub-scene.

4.1. Primitive shape abstraction

It is observed that the objects of indoor scenes are generally man-made and composed of structural parts, which can be approximately represented by primitive shapes, such as planes and cylinders. Accordingly, to speed up processing, we take advantage of primitive shapes to abstract the geometry of indoor objects. For instance, a coffee table can be abstracted with a horizontal plane (corresponding to the top face) and four vertical cylinders (i.e., four legs). In particular, we only consider two types of primitives in our context, that is, plane and cylinder, which are sufficient to abstract the indoor objects according to our various experiments.

On the basis of RANSAC, we employ a "fitting-and-removing" strategy to generate primitive shapes from the point cloud of each sub-scene. All the associated points within a primitive shape are extracted as the *support* of the chosen primitive shape candidate. The *support* is removed from the sub-scene point data and the fitting operation restarts over the remaining points. The "fitting-and-removing" procedure is repeated iteratively until the number of points of the *support* is less than a specified threshold. As a result, all primary primitives can be generated.

4.2. Anchor-guided topology graph construction

Graph construction. After fitting primitives over the sub-scene, we are to construct the corresponding topology graph to the sub-scene. A straightforward strategy is to connect every couple of adjacent primitives to form a topology graph. Apparently, there are a large number of combinations accordingly and the size of the topology graph could be huge. As observed, there always exists a functional plane for man-made object, which is generally with a relatively big area. Moreover, the other parts usually have topological relations with the functional plane. Based on this observation, we propose a novel method to construct the topology graph of a sub-scene guided by the functional planes of indoor objects, referred to as *anchors*. In particular, all graph edges are induced from anchors to their adjacent primitives.

Given a sub-scene S abstracted by a set of primitives, we represent the topology of the sub-scene with a property graph $G = (V, E, A_V, A_E)$. V and E denote the graph node set and the edge set; A_V and A_E are the attributes of nodes and

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

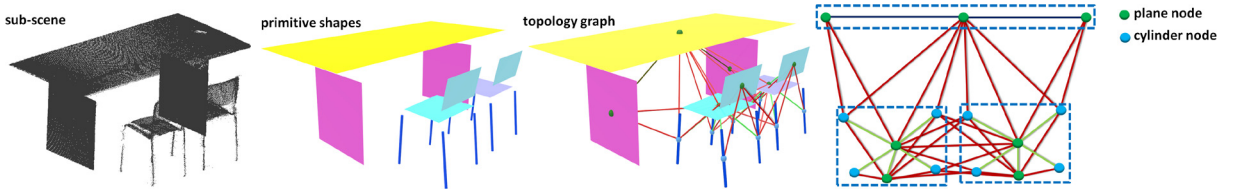

Fig. 2. Topology graph construction on a sub-scene. The point cloud is given (left), which is fitted with planes and cylinders (mid-left). We use these primitive shapes to construct a topology graph (right) to represent the scene.

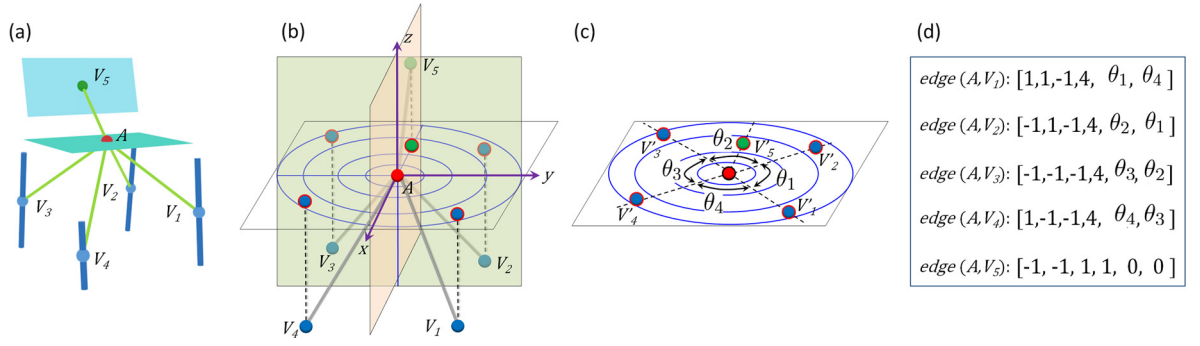

Fig. 3. Edge attributes of a chair. (a) Chair primitives and the topology graph; (b) Edge attribute: *edge orientation*. By constructing the LCS on the anchor plane, the corresponding quadrants of all nodes are determined in terms of the LCS and thus the *edge orientation* attribute for each edge is obtained. (c) Edge attribute: *primitive distribution*. By projecting the nodes onto the anchor plane, two adjacent angles for each projected edge are calculated, which, together with the number of nodes within the same concentric circle, compose the *primitive distribution* attribute; (d) Edge attributes for all edges within the graph.

edges, respectively. In particular, the graph nodes correspond to the fitted primitives. For each primitive plane, we project its associated point data onto the plane to obtain the convex hull over the projection. If the area of the convex hull is bigger than a specified value s, we would consider it as an anchor. Accordingly, a number of anchors are detected. For each anchor, we search its adjacent primitives by checking the distance from their geometric centers to the anchor center. If the distances are within the given range d, the corresponding nodes to those primitives are linked to the anchor so that graph edges are generated. By this means, the topology graph of the sub-scene is constructed, that is, V and E are obtained. Fig. 2 illustrates the topology graph construction of a sub-scene with one table and two chairs.

Attribute definition. Next, we define the attributes for graph nodes V and edges E, denoted by A_V, A_E, respectively. The node attributes in our context compose *anchor flag*, *primitive type*, *primitive size*, while the edge attributes consist of *edge orientation* and *primitive distribution*. For an arbitrary graph node, *Primitive type* gives its surface type, that is, 0 for plane and 1 for cylinder. *Primitive size* stands for the area if it is planar, or the height for the cylindrical node. *Anchor flag* indicates whether it is an anchor or not.

For each graph edge, it is induced from an anchor node. As the anchor is always planar, we construct a local coordinate system (LCS) on it, where the origin is the center of the bounding rectangle of the anchor; the Z-axis is the normal of the anchor plane, and the X-, Y-axis are arbitrarily chosen from two orthogonal axes on the anchor plane. With this LCS, all the connected nodes to the anchor node have their respective quadrants, which are used to represent the *orientation* attributes of edges. In particular, the *edge orientation* attribute is represented with a triple-tuple, the elements of which correspond to the X, Y and Z signs of the non-anchor node in the LCS. From the definition, the *edge orientation* attribute exhibits the topological relations of the connected nodes to the anchor node, as shown in Fig. 3.

The *primitive distribution* attribute indicates how the associated structures distribute in terms of the anchor. Specifically, we project all connected node centers onto the X–O–Y plane of the LCS. Meanwhile, we uniformly generate a series of concentric circles centered at the origin of the LCS. Given a concentric annulus, let $Prj = \{pri_i\}_{i=1}^{n}$ be the projection set of nodes connected with the anchor, in which all elements have been sorted clockwise on the anchor plane, as shown in Fig. 3. For each node projection pri_i, we calculate two associated angles formed by pri_{i-1}–O–pri_i and pri_i–O–pri_{i+1}. Accordingly, the cardinality of projection set, n, together with those two angles, compose the *primitive distribution* attribute, which is assigned to the corresponding graph edge formed by the anchor node and the original node of pri_i. We can see that the *primitive distribution* attribute conveys the structural constitution information as well as the topological relations among the structures of the object. Fig. 3 illustrates how to define the edge attributes on a chair.

4.3. Decomposition via subgraph matching

Once the topology graph of each sub-scene is constructed, we exploit a subgraph matching method to decompose the sub-scene into individual indoor objects. Particularly, we collect hundreds of 3D models of indoor objects within our model

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

repository, such as chairs, desks, sofas, coffee tables and so forth. The topology graphs are constructed for all repository models, which are referred to as graph templates. Given the topology graph of a sub-scene, we take advantage of the graph templates to match the topology graph so as to partition it into subgraphs, each of which corresponds to one of graph templates. As a consequence, individual indoor objects are decomposed from the sub-scene.

Graph matching. The goal of graph matching is to seek the optimal correspondences between two graphs. Specifically, let $G = (V, E, A_V, A_E)$ be the topology graph of a sub-scene, $G_t = (V_t, E_t, A_{V_t}, A_{E_t})$ a graph template from one of the repository models. V and E are the graph node set and the edge set; A_V and A_E denote the attributes of nodes and edges, respectively. There are $|V|$, $|V_t|$ nodes in G, G_t, i.e., $V = \{1, 2, \ldots, |V|\}$, $V_t = \{1, 2, \ldots, |V_t|\}$. For each node $i \in V$ of G, let $a_i^{(k)} \in A_V$ $(k = 1, 2, \ldots n^v)$ be its attributes, where n^v is the number of node attributes. For each edge $(i, j) \in E$, let $a_{ij}^{(k)} \in A_E$ $(k = 1, 2, \ldots n^e)$ be its attributes, where n^e is the number of edge attributes. The matching correspondences between G and G_t can be represented by a binary affinity matrix $\mathbf{X} \in \{0, 1\}_{|V|\times|V_t|}$. If the node $i \in V$ matches the node $i_t \in V_t$, then the corresponding entry of the matrix is 1 (i.e., $\mathbf{X}_{i,i_t} = 1$); 0 otherwise ($\mathbf{X}_{i,i_t} = 0$). By converting the matrix into a vectorized replica, i.e. $\mathbf{x} \in \{0, 1\}_{|V||V_t|\times 1}$, the graph matching between G and G_t can be formulated to find the optimal correspondences $\mathbf{x}^*$:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \mathcal{S}(\mathbf{x}|G, G_t), \;\; s.t. \;\; \forall i \in V, \sum_{i_t \in V_t} \mathbf{x}_{i,i_t} \leq 1, \forall i_t \in V_t, \sum_{i \in V} \mathbf{x}_{i,i_t} \leq 1 \tag{2}$$

where $\mathcal{S}(\mathbf{x}|G, G_t)$ is a function measuring the matching similarity between G and G_t under the correspondences $\mathbf{x}$, which is discussed in detail below.

We use a quadratic assignment formulation (Leordeanu and Hebert, 2009) to define the matching similarity function, which assumes the similarity function to measure the mutual similarity of graph attributes. According to the formulation, the first-order similarity function measures the node similarity from different graphs, while the second-order similarity function measures the edge similarity from different graphs. Therefore, we are able to encode these two types of functions into a symmetric similarity matrix $\mathbf{M}_{|V||V_t|\times|V||V_t|}$, and thus define the matching similarity function $\mathcal{S}(\mathbf{x}|G, G_t)$ as:

$$\mathcal{S}(\mathbf{x}|G, G_t) = \mathbf{x}^T\mathbf{M}\mathbf{x}, \quad \begin{cases} \mathbf{M}_{ii_t,ii_t} = \Omega_v(\mathbf{d}_{ii_t}, \mathbf{w}^v), \;\; i \in V, i_t \in V_t \\ \mathbf{M}_{ii_t,jj_t} = \begin{cases} \Omega_e(\mathbf{d}_{ii_t,jj_t}, \mathbf{w}^e), & (i, j) \in E, (i_t, j_t) \in E_t \\ 0, & \text{otherwise}, \end{cases} \end{cases} \tag{3}$$

where $\Omega_v(\mathbf{d}_{ii_t}, \mathbf{w}^v)$ is the first-order similarity function measuring the unary similarity for two nodes $i \in V$ and $i_t \in V_t$, which is set on the diagonal of $\mathbf{M}$; $\Omega_e(\mathbf{d}_{ii_t,jj_t}, \mathbf{w}^e)$ is the second-order similarity function measuring the pairwise similarity for two edges $(i, j) \in E$ and $(i_t, j_t) \in E_t$, which is on the non-diagonal of $\mathbf{M}$.

We define $\mathbf{d}_{ii_t} = \{d^{(k)}{}_{ii_t}\}_{k=1}^{n^v}$ as the Euclidean distance of node attributes, i.e., $d_{ii_t}^{(k)} = \|a_i^{(k)} - a_{i_t}^{(k)}\|$. Similarly, $\mathbf{d}_{ii_t,jj_t} = \{d_{ii_t,jj_t}^{(k)}\}_{k=1}^{n^e}$ is the Euclidean distance of edge attributes, that is, $d_{ii_t,jj_t}^{(k)} = \|a_{ij}^{(k)} - a_{i_tj_t}^{(k)}\|$. $\mathbf{w}^v$ and $\mathbf{w}^e$ are the weights for each node and edge attributes, respectively. Specifically, the node and the edge similarity functions are expressed as:

$$\begin{cases} \Omega_v(\mathbf{d}_{ii_t}, \mathbf{w}^v) = \max\left(0, \left(1 - d_{ii_t}^{(1)}\right)\left(w_0^v - \sum_{k=1}^{n_v} w_k^v d_{ii_t}^{(k)}\right)\right) \\ \Omega_e(\mathbf{d}_{ii_t,jj_t}, \mathbf{w}^e) = \max\left(0, w_0^e - \sum_{k=1}^{n_e} w_k^e d_{ii_t,jj_t}^{(k)}\right) \end{cases} \tag{4}$$

We strictly enforce the similarity functions to be positive so that all elements of $\mathbf{M}$ are positive, which facilitates solving the optimization problem. Apparently, we have $n_v = 3$, $n_e = 6$ in our context. In terms of the weights, we empirically set $w_0^v = 1$, $w_1^v = \frac{1}{3}$, $w_2^v = \frac{1}{3}$, $w_3^v = 0.26$, $w_0^e = 1$, $w_1^e = w_2^e = w_3^e = \frac{1}{6}$, $w_4^e = 0.15$ and $w_5^e = w_6^e = \frac{1}{3\pi}$ based on various experiments, which yield satisfactory results for all our examples.

The formulation of Equation (3) under the maximization Equation (2) is an NP complete problem. Therefore, we take some measures to try to reduce complexity, that is, sparsifying the similarity matrix (i.e., reducing the number of similarity values considered in the graph matching). Specifically, we only measure the similarity between nodes sharing the same primitive type. Moreover, we establish candidate matches only starting from anchor nodes and then restrict the comparisons only from the edges induced from the corresponding anchor nodes. Consequently, the number of similarity values is significantly decreased and our graph matching can be efficiently achieved. Various graph matching techniques can be used to solve the maximization problem in Equation (2), and the TRW-S algorithm (Kolmogorov, 2006) is exploited in our work.

Sub-scene decomposition. Based on the graph matching technique above, we are able to formulate our sub-scene decomposition problem. Given the graph template set $\mathcal{G} = \{G_t^i\}_{i=1}^{n^t}$ (n^t is the number of graph templates), the graph G of a sub-scene S can be partitioned by:

$$G = \bigcup G(G_t^*, \mathbf{x}^*), \quad (G_t^*, \mathbf{x}^*) = \arg\max_{G_t^i \in \mathcal{G}, \mathbf{x}} \mathcal{S}(\mathbf{x}|G, G_t^i) \tag{5}$$

where $G(G_t^*, \mathbf{x}^*)$ represents the subgraph extracted from G under the correspondences $\mathbf{x}^*$ to G_t^*. In particular, we exploit the "matching-and-removing" strategy to perform graph partition. Once we find the best template match G_t^* under the correspondences $\mathbf{x}^*$, the corresponding subgraph can be found and removed from G. For the remaining graph, we perform

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

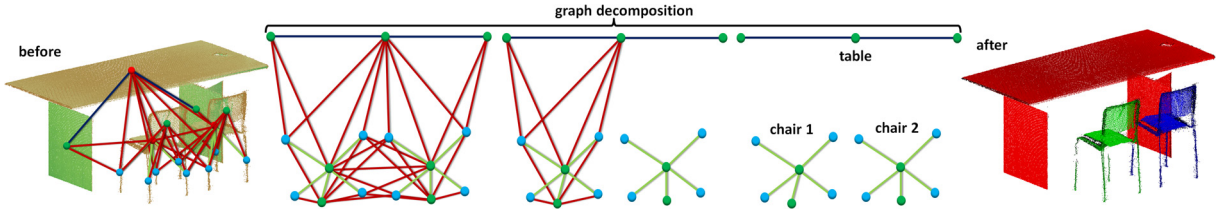

Fig. 4. Sub-scene decomposition. By constructing the topology graph (left), we seek the correspondences between the graph and template graphs. Once one subgraph is matched, it is removed from the topology graph (middle). As a result, the sub-scene is decomposed successfully and thus all individual objects are obtained, represented with different colors (right). (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

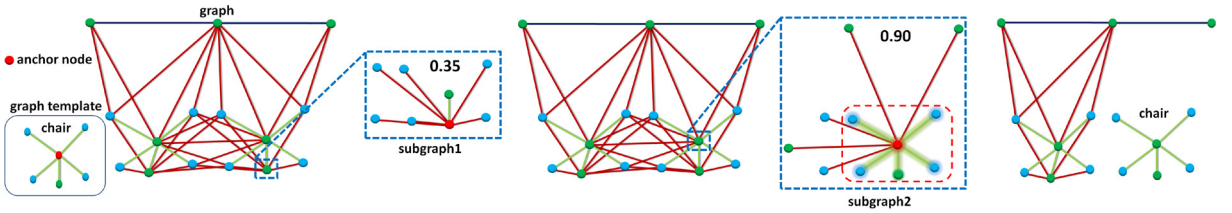

Fig. 5. Illustration of anchor-based graph matching. Taking a template graph, we match it with the topology graph obtained from Fig. 4. The topology graph includes several anchor nodes. "*subgraph1*" and "*subgraph2*" stand for the matching similarity values from two subgraphs with different anchor nodes to the same template graph. In "*subgraph1*", the matching similarity value is 0.35, while 0.90 in "*subgraph2*". Hence, the subgraph in "*subgraph2*" is considered matched and the corresponding chair is decomposed.

the matching procedure above iteratively until G is completely partitioned. Therefore, all subgraphs of G are obtained, each of which corresponds to an individual object. By this means, we extract all individual objects within the indoor scene. More importantly, the category of each object is determined simultaneously.

Fig. 4 gives an example of sub-scene decomposition using our method. The sub-scene contains one table with two chairs. By graph matching, the topology graph is partitioned iteratively into three subgraphs, each of which corresponds to one object. Fig. 5 illustrates the matching process on a subgraph from Fig. 4.

5. Learning-based scene modeling

As presented above, each individual object has been classified from the scene in Section 4. In this section, we present a data-driven modeling algorithm for individual indoor objects based on the shape database. We pose this model matching problem as a model instance recognition problem, which is solved by using a Randomized Decision Forest (RDF) classifier.

5.1. Learning based object recognition

Feature descriptors. To perform learning-based object recognition, we first need to define a set of features which can discriminatively describe objects with regard to scanned point data within the scene. There exist some classical local descriptors like spin images, curvature and SIFT, which are, however, not applicable for our context as the scanned point data are fairly noisy. We characterize our feature descriptors as: global, generic, discriminative for man-made indoor objects and efficient to compute. Specifically, we argue that the functional parts of indoor objects (i.e., the *anchors*) convey the most important topological and structural information of the objects. As observed, man-made objects consist of a natural segmentation along the normal direction of the anchor planes, and moreover the structures above the anchor planes are of significant difference from the counterparts under the anchors. In addition, the front view and left view of objects always exhibit distinct shapes. Based on these observations, we define our feature descriptors for indoor objects below, which are insensitive to noise, incompletion and sparsity.

Given an indoor object, we construct the Local Coordinate System (LCS) on the anchor plane of the object. As the topology graph of the object has the best match to one graph template, we thereby migrate the LCS of the graph template on its anchor plane to the object. With the LCS, the bounding box is computed for the point data of the object. We subsequently slice the bounding box into L slabs along its X-axis uniformly, as show in Fig. 6. By counting the number of points x_i in each slab, the first feature is generated as a vector of $X_f = (\frac{x_0}{N}, \frac{x_1}{N}, \ldots, \frac{x_{L-1}}{N})$ (N is the total number of the scanned points of the object). Similarly, we are able to obtain the feature vector along the Y-axis, i.e., $Y_f = (\frac{y_0}{N}, \frac{y_1}{N}, \ldots, \frac{y_{L-1}}{N})$.

The heights of the functional planes are usually distinct for different types of man-made objects, and even they are different for the same type of objects. Moreover, considering the practical scanning conditions, the scanned point data could be incomplete and some regions are missing. If the feature vector along Z-axis were calculated using the same means as the X-, Y-axis, the slab positions of anchor parts vary frequently. As known, the number of points on the anchor slab is always greater than other parts, as shown in Fig. 6, and hence the position variation of the anchor slab would lead to the

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

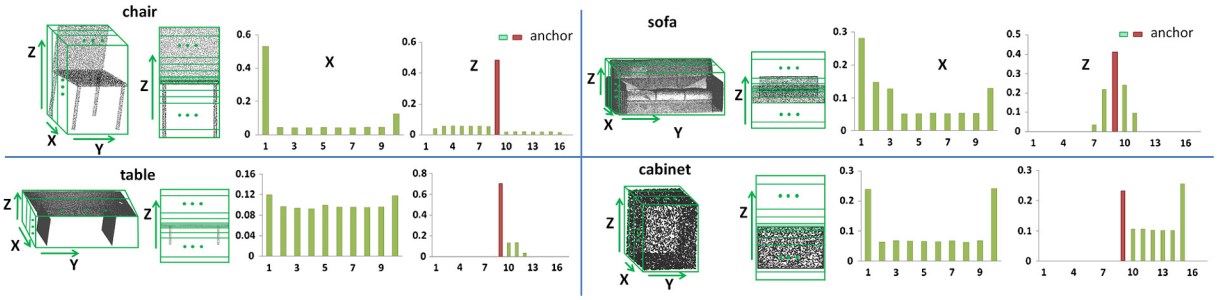

Fig. 6. Illustration of X, Z-axis feature descriptor definitions on chair, sofa, table and cabinet.

feature vector unreliable. To address this issue, we deliberately set the position of the anchor slab fixed in the feature vector, that is, in the middle of the vector. In particular, we slice the bounding box into $2M+1$ slabs by enforcing the anchor part to lay in the M-th slab (see Fig. 6). Accordingly, the third feature vector is formulated as $Z_f = (\frac{z_0}{N}, \frac{z_1}{N}, \ldots, \frac{z_{2M}}{N})$. Finally, the entire feature descriptors are obtained with the combination of three feature vectors.

Learning-based recognition. Due to occlusions, the scanned point data may contain considerable missing regions within the indoor scenes. Consequently, some parts could be missing. Therefore, we take advantage of Randomized Decision Forest (RDF) as our classifier to find the most similar models form the database to indoor objects. There are two advantages to use RDF (Nan et al., 2012): 1) it is an efficient classifier for multi-class classification and has proven to be available for a great deal tasks like classification in images; 2) it is also an effective method for handling missing data and maintains accuracy when a large proportion of the data are missing.

A RDF is an ensemble learning method for classification which works by constructing T decision trees at the training stage. A decision tree is a basic classifier in which each internal node composes a feature f and an attribute threshold ϵ. Given a set of points P, we can obtain the complete feature vector. By initiating from the root, the decision trees can classify P by comparing each feature f within our feature vector to the attribute ϵ from the root to the leaf node. Each leaf node of the tree t is labeled with a learned distribution $P_t(c|S, P)$ over classes c. Therefore, RDF achieves object recognition by averaging all trees in the forest, that is:

$$P(c|P) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|P) \tag{6}$$

A tree can be learned by choosing a series of features and the corresponding attributes that can split the given training data into subsets with same properties. In our implementation, the training set consists of nearly 500 different labeled objects, and we use both synthetic and scanned objects for our training. By testing, the RDF classifier can yield the best match model from the database to the indoor object.

5.2. Object modeling via template fitting

By performing object recognition, we have determined which model in the database is most similar to the scanned model. However, the transformation between two models is still unknown. Therefore, we need to optimize the transformation from the database model to the scanned one, including scale, translation and rotation, so that the largest overlap between two models can be achieved. Specifically, let P, $\mathcal{M}$ be the scanned points of the object and the matched model from the database, respectively, and then the optimized transformation $\{\mathbf{S}^\star, \mathbf{R}^\star, \mathbf{T}^\star\}$ can be obtained by maximizing the following objective function:

$$\{\mathbf{S}^\star, \mathbf{R}^\star, \mathbf{T}^\star\} = \arg\min_{\mathbf{S},\mathbf{R},\mathbf{T}} \sum_{p_i \in P} \|p_i - \mathbf{S}(\mathbf{R}\cdot\mathcal{M} + \mathbf{T})\|_2\,, \tag{7}$$

where $\mathbf{S}$, $\mathbf{R}$, $\mathbf{T}$ denote the scale, rotation and translation of the database model, respectively. $\|p_i - \mathbf{S}(\mathbf{R}\cdot\mathcal{M} + \mathbf{T})\|_2$ measures the Euclidean distance from point p_i to model $\mathcal{M}$ under the transformation $\{\mathbf{S}, \mathbf{R}, \mathbf{T}\}$. By aligning the anchor planes of the object and the database model, we can obtain a good initial registration, which facilitates the following optimization. We compute the distance by projecting p_i onto the transformed model along its normal, and minimize the distances between the scanned points to the database model in an ICP manner. Finally, we apply the respective transformations to all retrieved database models so that all corresponding 3D models are accurately placed into the indoor scene, resulting in a faithful reconstruction of the input scene.

6. Results and discussions

In this section, we evaluate our method on a large amount of scanned indoor scenes with various complexity and styles. Most of raw input point clouds used in our paper are scanned by our laser scanner, and the others are taken directly

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

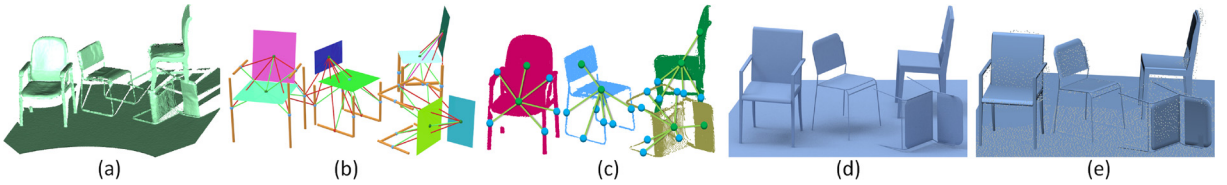

Fig. 7. Modeling of indoor objects with random poses. (a) The input point data of an indoor scene. (b) The topology graph of the entire scene; (c) Sub-scene decomposition and object classification; (d) The reconstructed models of the scenes and (e) The fusion of the reconstructed scene and the input point cloud. From (e), our reconstruction result has good geometric fidelity to the input scene.

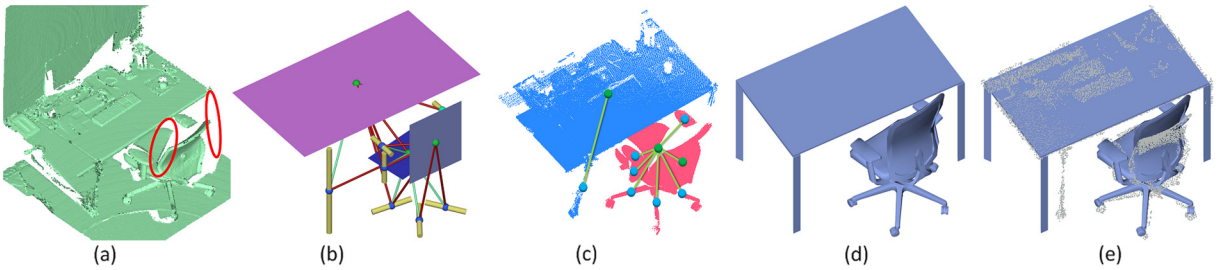

Fig. 8. Modeling of an indoor scene with missing data. (a) The input point cloud of an indoor scene, where partial data on desk legs are missing; (b) The topology graph; (c) The decomposed sub-scenes via graph matching; (d) The reconstructed models and (e) The fusion of the reconstructed scene and the input point cloud.

from Mattausch et al. (2014) and Nan et al. (2012). As seen, these raw scans are always noise and incomplete due to occlusions, which are directly processed and automatically modeled by our method.

6.1. Raw point scans of indoor scenes

Pose invariant. Fig. 7 shows the modeling result of an indoor scene, in which four chairs are randomly distributed and one of them even falls down. As discussed, our modeling method does not assume that all indoor objects are arranged along upward orientation as Nan et al. (2012). One of our key advantages is able to correctly identify objects with different poses. Our graph is to represent the topology information of objects, which is pose-invariant as shown in Fig. 7(b). The graph edge attributes are defined relative to the anchors, instead of the absolute coordinate system. Moreover, the graph matching algorithm is completely independent on the coordinate system. These make our object decomposition irrelevant to the definition of the coordinate system. Therefore, all chairs arbitrarily oriented are still decomposed and detected correctly in Fig. 7(c). On this basis, our classifier is able to find the most similar models for the detected objects from the database, and the reconstruction result is shown in Fig. 7(d). Fig. 7(e) shows the associated points together with the modeling result, which suggests our reconstructed models have good geometric fidelity to the original real scene.

Missing data. Fig. 8 presents the reconstruction result of the indoor scene with missing data. Due to occlusions, three legs of the table are missing and the seat of the chair is partially scanned (see the highlighted circles in Fig. 8(a)). For the completely missing parts, the corresponding primitives consequently cannot be fitted. For the partial anchor, the primitive can still be obtained. On this basis, the topology graph is constructed in Fig. 8(b). Thanks to the anchor-guided strategy, our graph matching algorithm tolerates the missing nodes in the object graph. In such a case, we match the template graph to *none*, which is added virtually in the object graph. Note that we also restrict that at least two nodes should be really existing in the object graph. The setting of *none* significantly reduces incorrect matching in practice. Therefore, the chair and the table are able to be segmented and detected correctly, as shown in Fig. 8(c). Furthermore, the reconstruction results are given in Fig. 8(d) and (e).

Complex cluttered scene. Fig. 9 shows the reconstruction result from a complex cluttered scene. The scene consists of more than 15 indoor objects, including tables, chairs, cabinets and so on. From Fig. 9(a), the objects are arranged disorderly and the types of objects are abundant. Moreover, the scanned point data contain a certain level of noise and severe incompletion. The whole scene is segmented into a few sub-scenes. Taking one sub-scene as an example in Fig. 9(b), the primitives are fitted accordingly. Note that every object in the sub-scene is incomplete and thus the constituent primitives of every object are fractional. Our matching algorithm is still able to decompose the topology graph into several subgraphs accurately, each of which corresponds to the individual object correctly, as shown in Fig. 9(c). The modeling result of the sub-scene is given in Fig. 9(d). Finally, the entire scene is shown in Fig. 9(e) together with the original point data. From the fusion result, our modeling method is capable of producing satisfactory results on the cluttered scenes from defect-laden, raw point data.

Scene with object outliers. Our database comprises desks, chairs, coffee tables, sofas and cabinets. We refer to objects with categories out of our database as outliers. Fig. 10 presents the modeling result of an indoor scene with outliers. There are

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

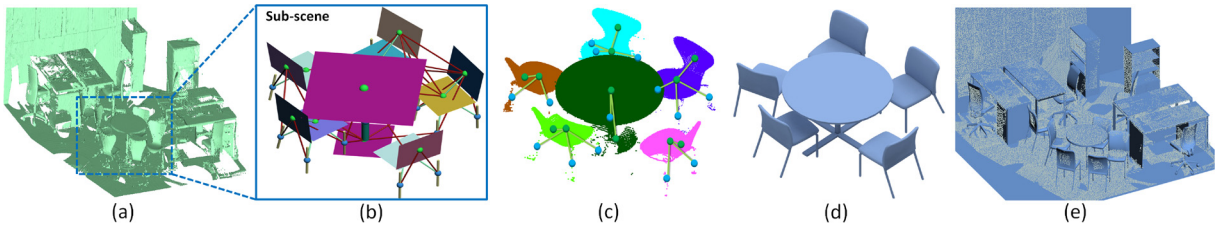

Fig. 9. Cluttered scene modeling. (a) The raw scan of an indoor scene with more than 15 objects. The scene is clustered into several sub-scenes and one of them is shown in (b). By graph matching, all individual objects are partitioned successfully in (c). (d) and (e) show the reconstructed models, and the fusion view of the modeled scene and the input raw scan, respectively.

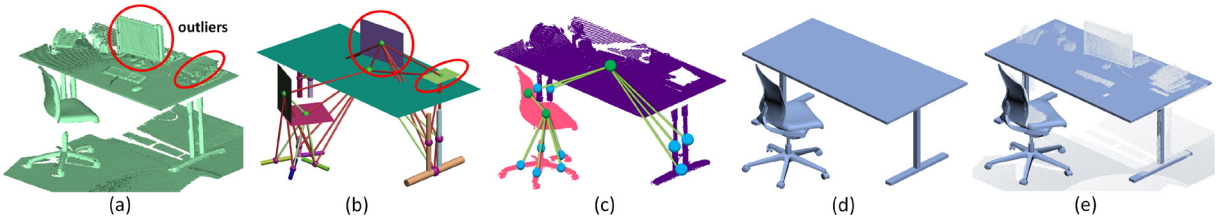

Fig. 10. Modeling of an indoor scene with object outliers. The topology graph still contains the nodes from those outliers in (b). Using our graph matching approach, only those subgraphs which have correspondences to the graph templates can be considered matched. Therefore, the nodes and edges from outliers would be isolated from the objects of interest, as shown in (c), (d) and (e).

Table 1
Timings of indoor scene modeling by our method (in s).

Figures	Fig. 7	Fig. 8	Fig. 9	Fig. 10	Fig. 11	Fig. 14(1)	Fig. 14(2)	Fig. 14(3)	Fig. 14(4)
Points	256,085	448,019	4,483,959	398,918	167,768	3,477,692	4,361,347	1,798,282	3,418,389
Time	35.26	67.83	356.58	59.94	27.85	314.52	395.85	124.25	336.24

a stack of books, a computer screen and a keyboard on the top of the desk (see Fig. 10(a)), which are regarded as outliers accordingly. Two primitives are fitted on those outliers, i.e., the book cover and the screen surface, and the corresponding topology graph is constructed in Fig. 10(b). As there are no graph templates on books and screens, our graph matching is able to detect those outliers correctly, and decomposes the corresponding subgraphs to the desk and the chair accurately, as shown in Fig. 10(c). The whole modeling results are given in Fig. 10(d) and (e). Certainly, to expand our shape database with more object categories would definitely enable us to reconstruct those object outliers successfully.

6.2. Comparison

We compare our modeling method with the *search-and-classify* approach from Nan et al. (2012) on an indoor scene. Two chairs stand on the ground with the upward orientation, while one falls down and two are placed in the slantwise direction. In Nan et al. (2012), the classification features as well as template fitting are performed on the assumption that objects are always with the upward direction with respect to the ground floor. Once the assumption is violated, both classification (e.g., the middle chair is misidentified as a table) and fitting (e.g., the right two chairs are aligned with ground) fail. Comparatively, our topology graph and recognition features are both defined on the basis of the functional parts (i.e., the anchors) of objects, which are independent on the placement orientation of objects and thus pose-invariant. Therefore, our modeling approach is able to handle indoor scenes with objects arbitrarily oriented.

6.3. Quantitative evaluation

The results shown in Figs. 11–13 have visually demonstrated the superiority of our algorithm in terms of pose invariance, data missing, scene complexity and outliers robustness. We provide some quantitative results in Table 1, and further explore the robustness of our feature definition in Fig. 13 and of our scene reconstruction in Fig. 12 on noise, outliers and sampling. Note that the added noise is provided by a zero-mean Gaussian function proportional to the diagonal length of the bounding box of the input data and the synthetic outliers are generated randomly in the bounding box.

Feature definition robustness. To evaluate the robustness of our feature definition to sampling, noise and incompletion, we generate some synthetic data by adding Gaussian noise and outliers onto the raw point scan of a chair. We plot the histograms of the feature descriptor along z-axis in terms of the sampling ratio, noise level and missing ratio in Fig. 12. By down-sampling the point data to 30%, the feature descriptor distributions are almost the same. After reducing the points to one tenth, the distribution difference is still inconsiderable. In terms of noise, even though 40% noise is added, the feature

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

Fig. 11. Comparison to Nan et al. (2012) on modeling the scene with objects randomly arranged. (a) The input raw scan of an indoor scene. The modeling results are from (b) Nan et al. (2012) and (c) Ours. Nan et al.'s (2012) method is built upon the assumption that objects are always with the upward direction with respect to the ground floor, while ours can handle arbitrarily oriented scenes based on our functional-part guided analysis techniques, which are pose-invariant, as demonstrated in (d).

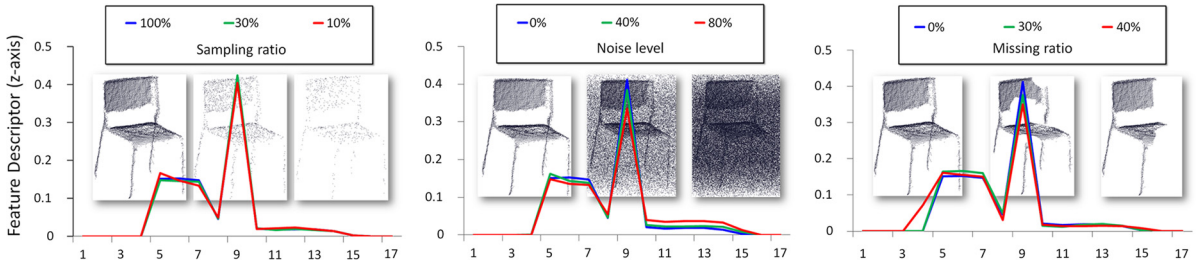

Fig. 12. Robustness of feature descriptors on sampling, noise and incompletion. The histograms of the feature descriptor along z-axis are plotted in terms of the sampling ratio, noise level and missing data ratio. The feature descriptor changes slightly even though the data have been down-sampled to 10%. We can see our feature descriptor definition is robust to data imperfection.

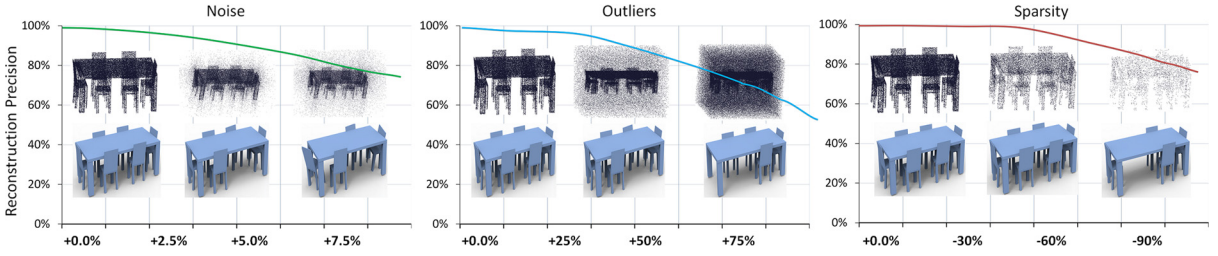

Fig. 13. Scene reconstruction robustness in terms of noise, outliers and sampling sparsity. From those tendencies, we note that our modeling method is insensitive to reasonably high levels of noise, outliers and sparsity. When the level of defection reaches extremely high, the modeling precisions could drop significantly.

descriptor stays stable. After increasing the level of noise up to 80%, there appears some deviation on the distribution. Regarding the missing data, when the missing ratio increases to 40%, the change on the feature descriptor distribution is not significant. Overall, our feature descriptor definition is quite robust to data sampling, noise and incompletion.

Scene reconstruction robustness. To further assess the performance of our modeling method, we investigate our modeling precisions with respect to data noise, outliers and sampling sparsity in Fig. 13. We create some synthetic data by manually designing a model of a table surrounded with several chairs in *Trimble Sketchup* and performing virtual scanning on it to generate point data. We can change the related setting of virtual scanning to acquire different density-level. Since we can easily distinguish between the original points from the scene and the added noise and outliers, such synthesized point data can be regarded as our ground truth. On this basis, we measure the corresponding modeling precisions to those three factors. From the figures, our modeling method is capable of handling high levels of noise, outliers and sparsity, and exhibits comparatively high robustness to defect-laden, raw point clouds as illustrated in Fig. 13. We experiment our method on a gallery of indoor scenes in Fig. 14.

Parameters. There are three main parameters in our method: 1) the distance threshold ϵ between sub-scenes; 2) the area threshold s to determine anchors in Section 4.2; 3) the range d within which nodes are connected in Section 4.2. The choice of ϵ depends on the crowding-distance in the indoor scenes. If the scene is severely cluttered, choosing a lower value of ϵ would be better; otherwise, two or more sub-scenes would be merged into one sub-scene, which has no influence on our results but could increase the computational time. We set empirically $\epsilon = 0.3$ for all experiments. The area threshold s is utilized to determine the anchors, and we conservatively set s to a relatively low value to avoid missing anchors. Specifically, we set $s = 0.05$ and $d = 1.0$.

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

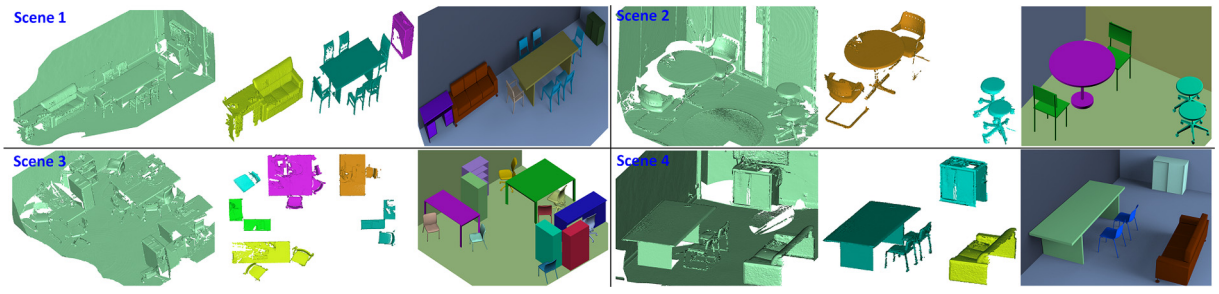

Fig. 14. Four reconstruction results from our modeling method. The input scans of point cloud on the left, sub-scenes show on the middle and reconstruction results are on the right.

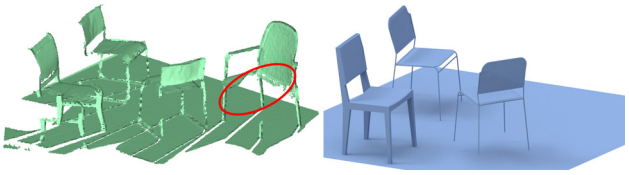

Fig. 15. Almost all of the points on the anchor plane are missing. Consequently, the anchor plane is not extracted and the corresponding chair fails to be reconstructed.

Performance. We have implemented our algorithm in *C++* and all experiments are performed on a PC with a 2.4 GHz CPU and 6.0 GB of RAM. Our implementation does not take advantage of the very parallelizable nature of some of the stages (e.g., sub-scene decomposition and object recognition), and apparently doing so would increase the efficiency significantly. We observe that the sub-scene clustering and object modeling stages are fast, while the sub-scene decomposition is relatively slow, especially for extremely cluttered scenes. However, the decomposition stage is not that time-consuming, as the number of graph templates is not huge even though the scale of database models could be large. Overall, the time performance is still comparatively efficient and suitable for real-time system. The experimental timings are given in Table 1.

Limitation. While our modeling algorithm performs quite well on a variety of real indoor scenes, it is not without limitations. As presented, we consider as the anchors the planar primitives with relative big areas. As shown in Fig. 15, considering a chair with a large portion of data missing on the seat, the anchor plane may not be able to be detected and consequently the chair could not be reconstructed. Therefore, more robust strategy on anchor detection is required in the future work.

7. Conclusions

We have presented a framework to efficiently decompose and reconstruct indoor scenes directly over raw scanned point clouds. The approach proceeds automatically without user interactions, and thus it is quite appropriate to real-time large-scale scanning, modeling and understanding applications. By introducing the anchor-guided strategy, our modeling method is capable of dealing with randomly arranged objects within complex, cluttered indoor scenes, instead of assuming all objects are always oriented with the upward direction. Based on the topology graphs of objects, our graph matching method is able to effectively decompose complex, cluttered scenes and detect individual indoor objects successfully. Furthermore, it is robust to noise and outliers by abstracting scenes with primitives. With discriminative feature descriptors defined, our recognition algorithm is able to tolerate a reasonably high level of data noise, outliers and sparsity. A variety of experiments on raw scans have demonstrated that our reconstruction method can generally produce geometrically faithful results from indoor scenes, even in the presence of severe data imperfection.

As discussed, we consider as the anchors the functional parts of indoor objects and our modeling method proceeds with the anchors guided. In case the anchors are missing from primitive fitting, we may not be able to detect the associated objects and consequently the reconstructed scenes are incomplete. Therefore, to seek more robust way to detect anchors needs to be studied in the future work.

Acknowledgements

We thank the anonymous reviewers for their valuable suggestions. The work was supported in part by National Natural Science Foundation of China (61402224), the Fundamental Research Funds for the Central Universities (NE2014402, NE2016004), Natural Science Foundation of Jiangsu Province (BK2014833), and Jiangsu Specially-Appointed Professorship.

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012

References

Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., Maierhofer, S., 2013. O-snap: optimization-based snapping for modeling architecture. ACM Trans. Graph. 32 (1), 6:1–6:15.

Bo, L., Ren, X., Fox, D., 2013. Unsupervised feature learning for rgb-d based object recognition. In: Experimental Robotics. Springer, pp. 387–402.

Chen, K., Lai, Y., Wu, Y.-X., Martin, R.R., Hu, S.-M., 2014. Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information. ACM Trans. Graph. 33 (6).

Du, H., Henry, P., Ren, X., Cheng, M., Goldman, D.B., Seitz, S.M., Fox, D., 2011. Interactive 3d modeling of indoor environments with a consumer depth camera. In: International Conference on Ubiquitous Computing (UbiComp).

Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J., 2004. Recognizing objects in range data using regional point descriptors. In: Computer Vision – ECCV 2004. Springer, pp. 224–237.

Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D., 2014. Rgb-d mapping: using depth cameras for dense 3d modeling of indoor environments. In: Khatib, O., Kumar, V., Sukhatme, G. (Eds.), Experimental Robotics. In: Springer Tracts in Advanced Robotics, vol. 79. Springer, Berlin, Heidelberg, pp. 477–491.

Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L., 2001. Topology matching for fully automatic similarity estimation of 3d shapes. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. ACM, pp. 203–212.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al., 2011. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. ACM, pp. 559–568.

Kim, Y.M., Mitra, N.J., Yan, D.-M., Guibas, L., 2012. Acquiring 3d indoor environments with variability and repetition. ACM Trans. Graph. 31 (6), 138.

Kolmogorov, V., 2006. Convergent tree-reweighted message passing for energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. 28 (10), 1568–1583.

Koppula, H.S., Anand, A., Joachims, T., Saxena, A., 2011. Semantic labeling of 3d point clouds for indoor scenes. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (Eds.), Advances in Neural Information Processing Systems 24. Curran Associates, Inc., pp. 244–252.

Leordeanu, M., Hebert, M., 2009. Unsupervised learning for graph matching. In: Computer Vision and Pattern Recognition. CVPR '09. June.

Lin, H., Gao, J., Zhou, Y., Lu, G., Ye, M., Zhang, C., Liu, L., Yang, R., 2013. Semantic decomposition and reconstruction of residential scenes from LiDAR data. In: Proc. of SIGGRAPH 2013. ACM Trans. Graph. 32 (4).

Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., Pajarola, R., 2014. Object detection and classification from large-scale cluttered indoor scans. Comput. Graph. Forum 33 (2), 11–21.

Müller, P., Wonka, P., Haegler, S., Ulmer, A., Van Gool, L., 2006. Procedural modeling of buildings. ACM Trans. Graph. 25 (3), 614–623.

Musialski, P., Wonka, P., Aliaga, D., Wimmer, M., Gool, L., Purgathofer, W., 2013. A survey of urban reconstruction. Comput. Graph. Forum 32, 146–177. The Eurographs Association & John Wiley & Sons, Ltd.

Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., Chen, B., 2010. Smartboxes for interactive urban reconstruction. In: Proceedings of SIGGRAPH 2010. ACM Trans. Graph. 29. Article 93.

Nan, L., Xie, K., Sharf, A., 2012. A search-classify approach for cluttered indoor scene understanding. ACM Trans. Graph. 31 (6), 137.

Oesau, S., Lafarge, F., Alliez, P., 2014. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. ISPRS J. Photogramm. Remote Sens. 90, 68–82.

Parish, Y.I.H., Müller, P., 2001. Procedural modeling of cities. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '01. ACM, New York, NY, USA, pp. 301–308.

Quattoni, A., Torralba, A., 2009. Recognizing indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE, pp. 413–420.

Ren, X., Bo, L., Fox, D., 2012. Rgb-(d) scene labeling: features and algorithms. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. CVPR. IEEE, pp. 2759–2766.

Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M., 2008. Towards 3d point cloud based object maps for household environments. Robot. Auton. Syst. 56 (11), 927–941.

Saxena, A., Sun, M., Ng, A.Y., 2009. Make3d: learning 3d scene structure from a single still image. IEEE Trans. Pattern Anal. Mach. Intell. 31 (5), 824–840.

Schnabel, R., Wessel, R., Wahl, R., Klein, R., 2008. Shape recognition in 3d point-clouds. In: The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, vol. 8. Citeseer.

Shao, T., Xu, W., Zhou, K., Wang, J., Li, D., Guo, B., 2012. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. ACM Trans. Graph. 31 (6), 136:1–136:11.

Shen, C.-H., Huang, S.-S., Fu, H., Hu, S.-M., 2011. Adaptive partitioning of urban facades. In: Proceedings of ACM SIGGRAPH ASIA 2011. ACM Trans. Graph 30 (6), 184:1–184:10.

Spina, S., Debattista, K., Bugeja, K., Chalmers, A., 2014. Scene segmentation and understanding for context-free point clouds. In: Keyser, J., Kim, Y.J., Wonka, P. (Eds.), Pacific Graphics Short Papers. The Eurographics Association.

Tal, A., Zuckerberger, E., 2007. Mesh retrieval by components. Adv. Comput. Graph. Comput. Vis., 44–57.

Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W., 2003. Instant architecture. ACM Trans. Graph. 22 (3), 669–677.

Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A., et al., 2010. Sun database: large-scale scene recognition from abbey to zoo. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition. CVPR. IEEE, pp. 3485–3492.

Please cite this article in press as: Wang, J., et al. Cluttered indoor scene modeling via functional part-guided graph matching. Comput. Aided Geom. Des. (2016), http://dx.doi.org/10.1016/j.cagd.2016.02.012