# Behavior Recovery/Analysis in Cluttered and Dynamic Indoor Scene

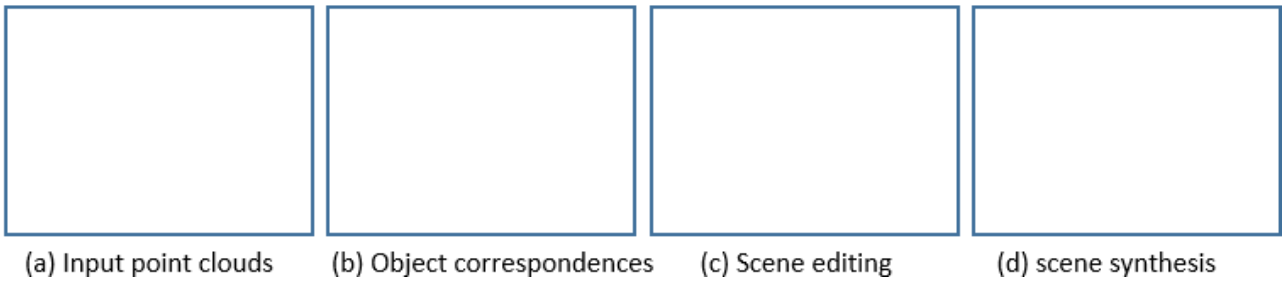


**Figure 1:** *Behavior recovery and behavior-based editing of dynamic cluttered indoor scenes. From a set of dense scans at different times (a), our system first extract the object correspondences (b) and behavior model ((xuejin: optionally show a graph) ). The recovered behavior model can be applied to many applications, such as scene editing (c) and indoor scene synthesis (d).*

## Abstract

While modeling static indoor scenes using RGBD cameras has been extensively studied in recent years, we introduce a *behavior recovery* system to investigate the behavior of objects in an cluttered indoor scene. In a daily indoor environment, because of object functions and human behaviors, the spatial placements of objects presents non-unique but statistically regular displacements. We take the *inverse problem* to recover object behaviors from a collection of point clouds captured at different times in a dynamic indoor scene. Our system consists of two key parts, *extraction of object correspondence* and *behavior model*. Given a collection of dense point clouds of an indoor scene in daily use at different times, the correspondence between objects are extracted using an iterative segmentation-and-registration process. Our algorithm is robust to noise and incomplete parts in imperfect scans. In the second part is to recover object behaviors, which represent the spatial arrangement of objects and interrelations between objects. Using our method, the correlation between concrete geometry and semantic behaviors of an indoor scene can be established. Therefore, the recovered behavior model can be adopted to many applications that requires labeled 3D database, such as scene synthesis, scene arrangement and so on. We evaluate our algorithm on a number of indoor scenes including office, bedroom and so on. The results demonstrate that our algorithm build accurate object correspondence from imperfect scans of cluttered indoor scenes, based on which, the recovered behavior provides natural principles for many other applications oriented to indoor scenes.

**Keywords:** Indoor scene, behavior analysis, dynamic, object correspondence

## 1 Introduction

Modeling indoor scenes has attracted a large amount of attentions for decades in computer graphics. Recently, many techniques have been presented to generate static 3D models for indoor scenes, including dense modeling from RGBD data [**?**; **?**; **?**; **?**], combing object classification and modeling [**?**; **?**; **?**], and synthesizing of 3D indoor scenes from large collection of examples [**?**; **?**]. While visually appealing models are obtained for rendering, it remains challenging to extract the semantics that the geometric representation essentially encodes. On the other hand, the semantics are required in many applications such as indoor scene understanding, scene editing, and etc.

Comparing with static scenes, dynamic scene analysis has significant value in interior design, animation making, etc. The manners of how furniture objects interact with each other and how furniture objects interact with users play a very important rule in interior design. Typically, the geometric representation including object model and spatial placements of objects at different times implicitly encode the object functions and human behavior in that environment. However, the dynamic indoor scene analysis has not been investigated much in computer graphics.

Though the geometry data collection carries the behavior information, it is non-trivial to extract behavior from imperfect scans by consumer-level RGBD cameras. The challenges are in two-fold because the scanned point clouds are noisy, incomplete and with errors. First, segmentation of objects with accurate boundary is tedious because the objects in a cluttered indoor environment are in a large variety of scales. Moreover, the massive occlusions and self-occlusions in a cluttered scene makes the segmentation more challenging. Second, it is an arduous task to figure out exact object correspondences given imperfect segmentations. Furthermore, there are a great of deal of similar structures in man-made objects in indoor scenes, which leads to large ambiguities of shape correspondences.

In this paper, we present a novel algorithm to explore object behaviors in cluttered indoor scenes from a set of point clouds scanned using consumer-level RGBD cameras without any training data. The consistency and difference between frames simultaneously provide valuable hints for recovering object correspondences. First, each frame is roughly segmented into patches, which are then clustered into objects as initial correspondences hypothesis. While we project the object models back to each frame with the corresponding transformation, the consistency between captured data and the project data strengthens the initial hypothesis while the difference indicates wrong correspondences. Based on this validation, each frame is re-segmented into patches. By iteratively perform the segmentation and registration steps, our work converges to coherent segmentations and correct correspondences between a bunch of objects at different scales in a large collection of points clouds.

In summary, the contributions of our system are three-fold:

1. To the best of our knowledge, our system, for the first time, performs behavior analysis in a dynamic indoor scene from point clouds scanned using consumer-level RGBD cameras without any database.

2. We present a global optimization framework to combine ob-

ject segmentation, correspondence extraction and behavior analysis in an iterative scheme.

3. We present a novel behavior model in dynamic indoor scenes, which can be applied directly to many appealing applications.

## 2 Related Work

Many techniques have been proposed to generate static 3D indoor scenes in computer graphics. Though none of them focus on dynamic scene analysis like our system, they provide valuable reference on the underlying techniques.

**Reconstruction from RGBD Images.** For static scenes, KinectFusion [**?**] enables the real-time reconstruction by holding and moving a depth camera. For large-scale indoor scenes with multiple rooms, reconstructing a dense 3D model from the noisy and incomplete scanned range data typically involves registration of point clouds in different views and a global optimization to reduce gaps in a large scene [**?**; **?**]. Their goal is mainly to generate high-quality point clouds but without semantic analysis of the objects appear in the scene. Recently, object classification is employed to assist modeling for massive indoor scenes that containing many instances of chairs, desks, etc. Koppula et al. [**?**] first introduce the learning algorithm to understand the RGBD data of an indoor scene. To further reconstruct the 3D model for a cluttered indoor scene, 3D model databases can be used as template by searching for similar 3D model and then fitting the template to the scanned data [**?**; **?**]. [**?**] do not manually collect 3D models to build the database. The template model is reconstructed by scanning the same object in different configuration. Each model has an additional presentation by geometric primitives. [**?**] trains the class model based on geometry and appearance features to segment and label the RGBD data captured under sparse views. By learned an initial model for each class of object in indoor environments from a pre-labelled database, the model are refined progressively with user-refined segmentation results. The 3D model can be generated by placing the most similar model in the database according to the RGBD data. If objects move in a scene, they can be detected and reposed by segmented and classified based on the learned model from previously reconstructed model [**?**]. Different with these techniques, we pay more attentions on analyzing the object behaviors from the dynamic range data.

**Reconstruction from Sequential Point clouds.** Many techniques have been proposed to reconstruct the object surfaces from the range data sequences. [**?**] uses a *statistical framework* to reconstruct the geometry from real-time range scanning. Each frame is divided into 3D pieces. A statistical model is used to iteratively merge adjacent frames by aligning pieces and optimizing their shapes. However, some geometric artifacts remain due to structured outliers and in some boundary regions. [**?**] presents a global registration algorithm to reconstruct *articulated 3D models* from dynamic range scan sequences. The surface motion is modeled by a reduced deformable model. Joints and skinning weights are solved in the system to register point clouds in different poses. (xuejin: We may also consider the furniture objects in indoor environments as articulated models, whose shapes under different poses can be deformed through connectors like hinge, slide, and so on. ) A new formulation of the ICP algorithm is proposed using sparse inducing norms [**?**]. While it achieves superior registration result on the data with outliers and missing region, only rigid alignment is handled. A proactive capturing is employed by asking the user to move the objects to capture both interior and exterior of a scene [**?**]. The correspondence between adjacent frames is built first then segmentation. Xu et al. [**?**] employ a robot to move objects during scene reconstruction so that the ambiguities in object structures can be solved from the dynamic data. All these techniques take the advantage of the differences caused by motion to extract valuable and semantic information of object structure. We do not only take the motion information in object modeling, but also take its advantage of implicitly encoding object behavior in a scene.

**Functionality Analysis using Context.** Many techniques of functional analysis of one category of objects using 3D model collections have been proposed [**?**; **?**]. Besides of functionality analysis of a single object, interaction between objects has drawn more and more attentions. The Icon descriptor is proposed to represent the functionality of 3D objects with its context [**?**]. Human action is involved to establish the correlations between the geometry and functionality of a region [**?**]. With the association of object arrangements and human activities, novel 3D scenes can be synthesized towards specific functions [**?**]. Inspired by these methods, we put our effort on behavior recovery, but from more challenging data. The point clouds can be easily to capture using RGBD cameras. However, the noisy and incompleteness brings tremendous challenges for extracting accurate object correspondences.

**Data-Driven Furniture Layout.** The general way producing the layout of furniture objects is to model a set of design rules and then to optimize an energy function given constraints by individuals. [**?**] formulates a group of layout guidelines in a density function according to professional manuals on furniture layout. When the user specifies the room shape and an initial arrangement of the set of furniture to be placed in the room, this system generates a number of layout suggestions by a hardware-accelerated Monte Carlo sampler. Instead of manually define the layout guidelines, the hierarchical and spatial relationships of the furniture objects can be learned from a set of examples [**?**]. Assembling these relationships and other ergonomic factors into a cost function, multiple arrangements can be yielded quickly by simulated annealing using a Metropolis-Hastings state search step. In these methods, manual labours are required in modeling the design rules and providing an initial layout. Fisher et al. [**?**] trains a probabilistic model for indoor scenes from a small number of examples. A variety of indoor scenes can be automatically synthesized from a few of user specified examples. Indoor scenes bring more difficulties for scene analysis because there are always many cluttered objects in different scales, shapes, and functions. A focal-driven analysis and organization framework is presented for heterogeneous collections of indoor scenes [**?**]. They develop a co-analysis algorithm which interleaves frequent patten mining and subspace clustering. The interrelations between objects play important role during furniture arrangement in these systems. However, the 3D scene models takes many efforts to collect for training. In comparison, our system provides an efficient framework to generate 3d model examples for many further applications.

## 3 Overview

The objective of our algorithm is to recover object behaviors from a set of point clouds scanned at different times for an indoor scene. To achieve this goal, objects and correspondences between objects and point clouds must be extracted. More specifically, our problem is to transfer the noisy and incomplete point-level data into object-level models and correspondences to understand the object behaviors in the scene.

Our system consists of two main steps, *point cloud segmentation* and *object registration*, as Figure **??** shows. The **input** is a set of point clouds scanned using a RGBD camera (a). We scan each scene at different times during a month using a Microsoft Kinect V1. A

**Figure 2:** *System overview.*

set of dense point clouds are generated using a real-time fusion system [**?**]. It is burdensome to scan every detailed structure in a cluttered indoor scene. As a result, each point cloud is incomplete and noisy due to object occlusions.

We first segment each frame simply using region growing (b). There are very likely many wrong boundaries in the generated patches. Then we cluster all the patches from all frames into $k$ clusters using $k$means. [**?**] have demonstrated the power of features based on bounding box in the segmentation of indoor objects. Therefore, we design the descriptor of each patch as the cascades of length, width and height of its bounding box, mean and standard deviation of the distance of each point to its bounding box, percentage of closest points to the faces of its bounding box. The feature dimension is then reduced using PCA. (xuejin: Currently, this step is manually done.)

In each cluster, the patches are registered using a joint registration method [**?**] to produce the object model of this cluster (c). There are inevitably wrong registration due to wrong clustering. Therefore, we project the generated object models back into each frame using the estimated transformation. Re-segmentation is performed using the model consistency and neighborhood information in each frame, as described in Sec. **??**. By iteratively register resegmented patches and re-segment frames, our system converges to a set of well-registered 3D object models and accurate correspondence between object model and all point clouds. Then we learn the behavior model (d) based the object correspondences from all input point clouds (Sec. **??**), then apply the behavior model into many applications (Sec. **??**).

# 4 Iterative Co-Segmentation and Joint Registration

#### 4.0.1 Logic Outline

如 Figure **??**所示展示了迭代分割与配准的系统流程。

## 4.1 Region Grow

区域生长的操作是对每一帧独立进行的。
输入是每一帧的点云以及每一帧所对应的 label。
输出是更新之后的 label。
所谓 label 就是一个整数的数组它赋予点云中每一个点一个 $id$，用来指示它所属的分割区域。我的实现中默认 $id = 0$ 的区域是未知区域（在图示中对应黑色区域）。在最开始时 label 的初始值是全为 0。
区域生长操作是针对每一帧中的未知区域进行的，所使用的准则是如果两个区域之间有点互相在以 R(默认 $R = 0.02$) 为半径的邻域中就将这两个区域合并为同一个区域。
选择这样的准则基于的假设是相距较远且不连续的两组点云不太可能属于同一物体。

我们认为这样的准则会自然得到欠分割（under-segment）的分割结果。以便于后续能够进行配准 (registration) 操作
Figure **??** 显示了第一次与第二次区域生长的过程。

## 4.2 Object Clustering ( Unify Label )

这一步操作是将每一帧的不同分割区域的 patch 无监督的聚成多个类以便于后续对每一类进行配准 (registration)。这一步输入是所有的点云以及它们对应的 label，输出仍然是更新之后的 label。先依照前一步的 label 将点云分成多个 patch，对所有的 patch 提取特征，然后依据投影矩阵将特征降维，然后再在低维空间中根据聚类中心为每一个 patch 重新分配 id 并更新每一帧的 label。
关于使用了哪些特征？
如何得到投影矩阵？
如何确定聚类中心？
以下分别进行说明：

#### 4.2.1 Features

Table **??**列举了所有使用的特征以及对应的维数。其中后三种 feature 实际上是以 bounding box 为参考提取的关于物体形状的特征。

| Feature | Dimension |
|---|---|
| RGB Color Histogram | 125 |
| Size of Bounding Box | 3 |
| Percentage of Points Closest to Front-Back Left-Right Up-Down of its Bounding Box | 3 |
| Mean Distances to its Bounding Box ( Seperately Accounted for Front-Back Left-Right Up-Down ) | 3 |
| Standard Deviation of Distances to its Bounding Box ( Seperately Accounted for Front-Back Left-Right Up-Down ) | 3 |

**Table 5:** *Patch Features*

#### 4.2.2 Projection Matrix

为什么降维，以及降维维数的选择:
1. 将特征向量进行线性投影相当于为不同的特征赋予不同的权重来衡量。
2. 之所以降维是为了后续能够进行 k-means 更新聚类中心。k-means 相当于一个简化的混合高斯的估计过程，而要有效估计

| 步骤 | 期望目标 | 目前算法实际达到的效果 | 主要差异 |
|---|---|---|---|
| Region Grow | 输入：一个由多个物体混在一起没有被 label 的区域（要求不能有一个物体部分被 label 而另一部分没有被 label）输出: 欠分割的结果————————将每一帧分为多个 patch 每一个 patch 由一个或多个物体组成 | 将每一帧中还没有被 label 的区域按照空间是否近邻分为多个 patch | 只要输入符合要求我认为在室内数据中是等价的 |
| Clustering | 输入所有的 patch，输出每个 patch 所属类别，要求每一类所有的 patch 中最大的（所占点最多）物体都相同 | 以 patch 最多的一帧的 patch 特征为初始用 k-means 来获取聚类中心，将每帧中特征空间中距离聚类中心最近的归为该类 | 完全不等价，也没有满足需求 |
| Joint Registration | 输入同一类的 patch，将它们按照类中 patch 共有的最大的物体配准到一起 | 输出一个模型与一组刚体变换，在存在 noise 和 outlier 的前提下，以高斯为先验最大化后验概率————————这些观察到的 patch 是由这个模型按照这组刚体变换生成的 | 并不等价，但是从结果来看我觉得是能够满足需求的 |
| Graph Cut | 将每一类的每个 patch 中除了主体物体之外的部分重新标记为没有被 label 的区域 | 一个 superpixel 如果按照某个类的变换能够在每一帧都匹配得（如能量项中的取 max 就是为了表达需要在每一帧中都匹配得好才叫匹配得好）很好则 label 为这个物体，果都匹配得不好则重新被标记为没有 label 的区域（匹配得好与不好的门限的设置依然是不清晰的） | 因为噪声和形变的存在所以并不能够等价 |

**Table 1:** *Logic Outline v0.0*

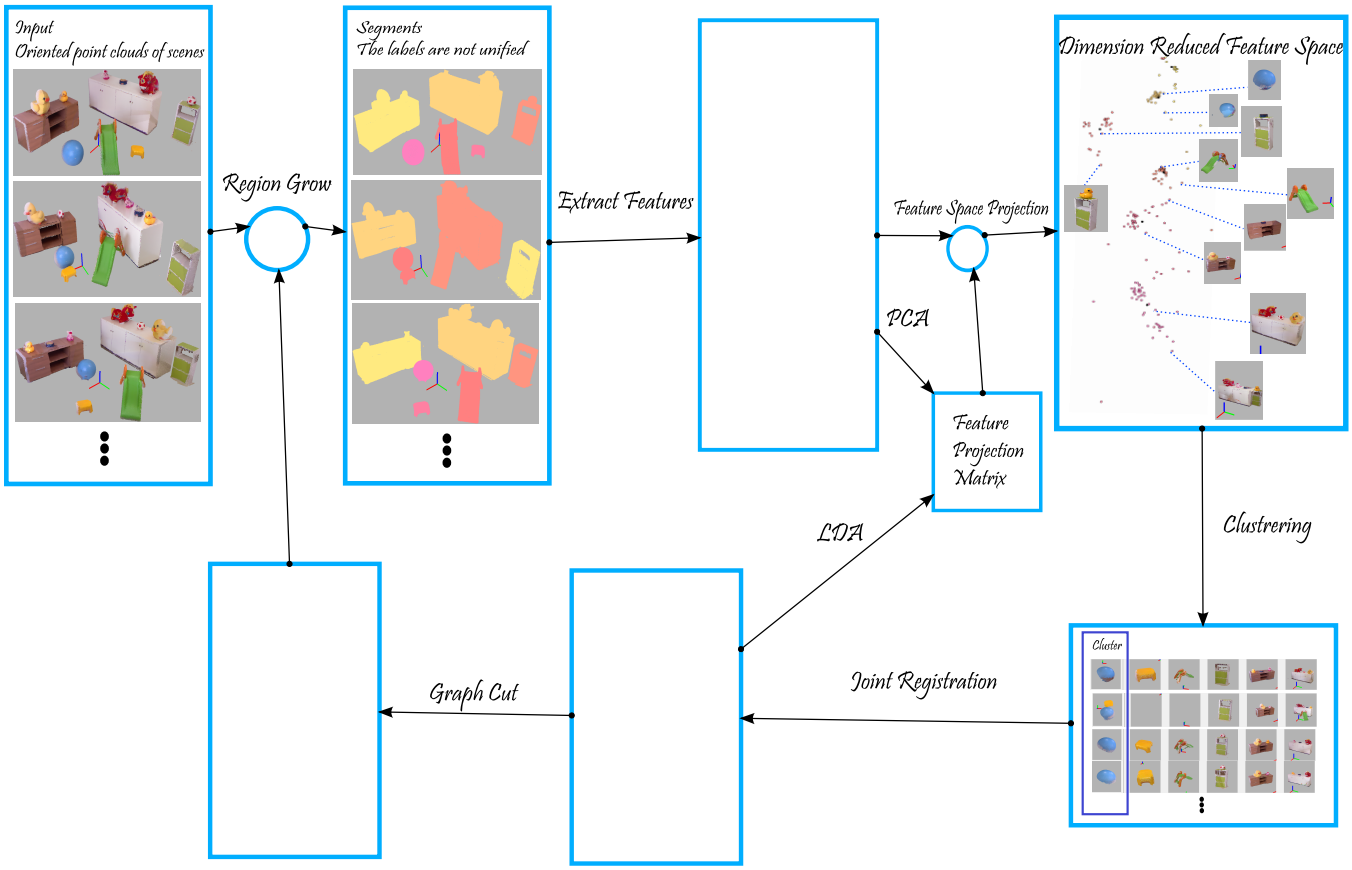


**Figure 3:** *Iterative co-segmentation and joint registration.*

| 步骤 | 期望目标 |
|---|---|
| Region Grow | 输出：<br>每一帧每个 pixel 有一个 label 值要求：<br>1. 在任意一帧中，属于同一个物体的 pixel 必须被赋予同一个 label 值。<br>2. 在任意一帧中，属于不同的物体的 pixel 也可以被赋予同一个 label 的值。<br>（被赋予同一个 label 值的 pixel 被认为组成一个 patch，<br>则这项要求等价于一个 patch 可以是一个或多个物体的集合）<br>3. 在不同的帧之间，属于同一个物体的 pixel 不需要被赋予同一个 label 值。 |
| Clustering | 输出：<br>每个 patch 赋予一个新的 label 的值，其中可以存在一个特殊的零 label<br>要求：<br>1. 在不同帧之间，同一个 label（零 label 除外）的 patch（称为同一个类的 patch）必须包含<br>同一个物体，且这个相同的物体在包含它的 patch 中占有的点的比例足够大（称为类内核心物体）<br>2. 允许原本多个 patch 被同时赋予零 label |
| Joint Registration | 输出：<br>除了零 label 的 patch 以外，对于每一个 patch 得到一个刚体变换，对于每一类得到一个物体模型。<br>要求：<br>1. 零 label 除外，每一类的 patch 按照所得到的刚体变换变换以后，<br>这一类的类内核心物体将会在三维空间中对齐。 |
| Graph Cut | 输出：<br>更新每个 pixel 的 label，使得原本属于类内核心物体的 pixel 的 label 与原本类的 label 相同，<br>而不属于类内核心物体的 pixel 的 label 被置为零 label<br>要求：<br>1. 允许部分属于类内核心物体的 pixel 被置为零 label。<br>2. 不允许不属于类内核心物体的 pixel 保留原来该类的 label。 |
| Region Grow | 输出：<br>给每一个零 label 的 pixel 赋予一个非零的 label 值。<br>要求：<br>1. 对于不属于上一步中类内核心物体的 pixel 要求符合第一次<br>region grow 的要求（属于新分出来的物体的 pixel）。 |
| Clustering | 输出：<br>每个 patch 赋予一个新的 label 的值，不存在零 label 要求：<br>1. 在不同帧之间，同一个 label（零 label 除外）的 patch（称为同一个类的 patch）<br>必须包含同一个物体，且这个相同的物体在包含它的<br>patch 中占有的点的比例足够大（称为类内核心物体）<br>2. 允许 patch 恢复为上一步聚类时的 label 值 |

**Table 2:** *Logic Outline v0.1*

| 步骤 | 期望目标 |
|---|---|
| Region Grow | 输入：<br>同一个场景不同时刻扫描的 T 个 Point Cloud:$\{F_t\}$<br>场景中总共包含没有被区分开的 N 个物体：$\{O_n\}$<br>第 n 帧点云有 $I(t)$ 个点，$p_{ti}$ 表示第 N 帧的第 i 个点<br>输出：<br>对于每一帧输出 $J(t)$ 个 Patch$\{P_{tj}\}$<br>要求：<br>1.Patch 之间没有重叠。<br>2. 同一个物体不能分到两个不同的 Patch 中。<br>3. 不同的物体可以被分到同一个 Patch 中。 |
| Clustering | 输入：<br>所有的 Patch$\{P_{tj}\}$<br>输出：<br>将所有的 Patch 分配到 M+1 个类当中 $\{C_m\}$<br>其中 $C_0$ 为无语义类，其余类为有语义类。要求：<br>1. 同一帧的两个 Patch 不能属于同一个有语义类。<br>2. 属于同一个有语义类的 Patch 必须包含同一个物体，并且这个物体在包含它的 Patch 中必须占有足够大的比例。（这个 |
| Joint Registration | 输入：<br>已经被分配到个 M+1 类中的 Patch<br>设第 m 类有 K(m) 个 Patch，则 $P_{mk}$ 表示第 M 类的第 k 个 Patch<br>输出：<br>除了 $C_0$ 以及属于 $C_0$ 的 Patch 之外，对于每一个 $P_{mk}$ 得到一个刚体变换，<br>对于每一类得到一个物体模型 $M_m$<br>要求：<br>1. 除了属于 $C_0$ 的 Patch 之外，每一类的 Patch 按照所得到的刚体变换变换以后，<br>这一类的类内核心物体将会在三维空间中对齐。 |
| Graph Cut | 输入：（前一步的输出）<br>输出：<br>将不属于类内核心物体的点从该类的 Patch 中移除<br>要求：<br>1. 允许部分属于类内核心物体的点被错误的移除。<br>2. 不允许没有被移除干净的点。 |
| Region Grow | 输入：<br>前一步中被从 Patch 中移除的点。<br>输出：<br>从没有 Patch 归属的点中重新生成 Patch<br>要求：<br>1. 对于没有被错误移除的点要求在生成 Patch 时满足第一次 region grow 的要求 |
| Clustering | 输入：<br>上一步中新生成的 Patch<br>输出：<br>将所有的新 Patch 分配到 M 个新的类当中 $\{C_m\}$<br>要求：<br>1. 同一帧的两个 Patch 不能属于同一个有语义类。<br>2. 属于同一个有语义类的 Patch 必须包含同一个物体，<br>并且这个物体在包含它的 Patch 中必须占有足够大的比例。（这个物体称为类内核心物体）<br>3. 新生成的 Patch 如果不能被分配到一个有语义的类当中则<br>将他合并回到之前的 Patch 中去。 |

**Table 3:** *Logic Outline v0.2*

| 步骤 | 期望目标 |
|---|---|
| Region Grow | 输入：<br>同一个场景不同时刻扫描的 T 个 Point Cloud:$\{F_t\}$<br>场景中总共包含没有被区分开的 N 个物体：$\{O_n\}$<br>第 n 帧点云有 $I(t)$ 个点，$p_{ti}$ 表示第 N 帧的第 i 个点<br>输出：<br>对于每一帧输出 $J(t)$ 个 Patch$\{P_{tj}\}$<br>要求：<br>1.Patch 之间没有重叠。<br>2. 同一个物体不能分到两个不同的 Patch 中。<br>3. 不同的物体可以被分到同一个 Patch 中。 |
| Clustering | 输入：<br>所有的 Patch$\{P_{tj}\}$<br>输出：<br>将所有的 Patch 分配到 M+1 个类当中 $\{C_m\}$<br>其中 $C_0$ 为无语义类，其余类为有语义类。要求：<br>1. 同一帧的两个 Patch 不能属于同一个有语义类。<br>2. 属于同一个有语义类的 Patch 必须包含同一个物体，并且这个物体在包含它的 Patch 中<br>必须占有足够大的比例。（这个物体称为类内核心物体）<br>3. 应该考虑利用 Joint Registration 的反馈信息在不出错的前提下尽量多的将 Patch 分配到有语义的类当中。 |
| Joint Registration | 输入：<br>已经被分配到个 M+1 类中的 Patch<br>设第 m 类有 K(m) 个 Patch，则 $P_{mk}$ 表示第 M 类的第 k 个 Patch<br>输出：<br>除了 $C_0$ 以及属于 $C_0$ 的 Patch 之外，对于每一个 $P_{mk}$ 得到一个刚体变换，<br>对于每一类得到一个物体模型 $M_m$<br>要求：<br>1. 除了属于 $C_0$ 的 Patch 之外，每一类的 Patch 按照所得到的刚体变换变换以后，<br>这一类的类内核心物体将会在三维空间中对齐。 |
| Graph Cut | 输入：（前一步的输出）<br>输出：<br>将不属于类内核心物体的点从该类的 Patch 中移除<br>要求：<br>1. 允许部分属于类内核心物体的点被错误的移除。<br>2. 不允许没有被移除干净的点。 |
| Region Grow | 输入：<br>前一步中被从 Patch 中移除的点。<br>输出：<br>从没有 Patch 归属的点中重新生成 Patch<br>要求：<br>1. 对于没有被错误移除的点要求在生成 Patch 时满足第一次 region grow 的要求 |
| Clustering | 输入：<br>上一步 Region Grow 中新生成的 Patch<br>输出：<br>将所有的新 Patch 分配到 M+N 个类当中 $\{C_m\}$<br>要求：<br>1. 首先尝试将新 Patch 分配到原有的类当中，再利用其中剩下聚类产生新的类。<br>2. 同一帧的两个 Patch 不能属于同一个有语义类。<br>3. 属于同一个有语义类的 Patch 必须包含同一个物体，<br>并且这个物体在包含它的 Patch 中必须占有足够大的比例。（这个物体称为类内核心物体）<br>4. 新生成的 Patch 如果不能被分配到一个有语义的类当中则<br>将他合并回到之前的 Patch 中去。 |

**Table 4:** *Logic Outline v0.3*

一个一维的高斯模型就按只需要五个样本（一般估计一个线性模型都至少要五个样本）来算，二维的要达到同样的采样密度需要 25 个样本，三维则需要至少 125 个样本。而我们的每组数据实际只有不到一百帧每个物体都没有 125 个这么多的样本。因此我认为降维到两维是合理的选择，同时这也方便对特征空间进行 visualize。
降维的方法：
初始时投影矩阵是对所有 patch 的特征做 PCA 来生成的. 后续则根据配准的结果进行 LDA 来获得, 使用 LDA 相当于学习一组新的特征权重来重新衡量。LDA 获得的新的投影矩阵应会使得特征在新的子空间内的类内方差变小类间方差变大。类与类之间的可区分度更好。具体的做法在章节 **??**中的对应小节中还会仔细说明。

#### 4.2.3 Cluster Centers

聚类是在投影后的特征子空间内进行的，初始时候聚类中心是将 patch 最多的一帧的 patch 所对应的特征点作为聚类中心, 然后进行 k-means 更新聚类中心。再进行 registration 之后还会根据 registration 的结果再更新聚类中心。具体细节在具体的做法在章节 **??**中的对应小节中还会仔细说明。

#### 4.2.4 Cluster Assignment

算法 **??**说明了根据聚类中心在特征子空间中为一帧中的每个 patch 指定类别的算法过程，目的在于要保证每一帧中最多有一个 patch 被指定到某个类别。

**Algorithm 1** Assign Patch to Cluster

**Input:**
$\{P_i\}$:Patch Features of One Frame
$\{C_j\}$:Cluster Centers
**Output:**
$\{Id_i\}$:Identity for Each Patch in this Frame

1. Calculate Distance Matrix $D_{ij} := eucl\_dist(P_i, C_j)$
2. Set Column Index $CIndex := 1 : J$
3. Set Row Index $RIndex := 1 : I$
   **while** $D$ is not empty **do**
4. Find $(i, j) = min(D)$
5. Set $Id_{RIndex(i)} := CIndex(j)$
6. Remove Row $i$ and Column $j$ from $D$
7. Remove Element $i$ from $RIndex$
8. Remove Element $j$ from $CIndex$
   **end while**

### 4.3 Joint Registration

联合配准的步骤参考 [**?**] 的算法把之前步骤中被分为同一类的 patch 进行配准，对于整个系统而言这一步主要起到以下几方面的作用：
1. 为之前的聚类结果提供反馈。(更新特征投影矩阵和聚类中心)
2. 建立起帧与帧之间的对应关系。为后续的联合分割能量项的生成提供桥梁。
3. 在最后的迭代中配准生成的物体模型将作为结果输出。

#### 4.3.1 Remove Unreliable Cluster Center By Registration Result

Figure **??**中展示了若干个联合配准 (joint registration) 之后将各个 patch 对齐摆放后的图片。我们以配准的结果模型为参考计算各个 patch 与 object 模型 match 的程度，Figure **??**中的分数显示的是 match 的得分的分布情况包括最小值 (min), 最大值 (max), 均值 (mean), 中值 (med), 方差 (var), 标准差 (stddev). 我现在使用均值 (>0) 与标准差 (<0.1) 上设置阈值来判断是否为有效地聚类中心。(Figure **??**中**??**所示的床头柜实际混入了不少其它 patch) 其中匹配程度的得分计算如下：

$$Score = \frac{1}{N} \sum_{(i,j)\in M} \frac{\vec{n_i}\vec{n_j}}{1+\alpha||p_i - p_j||}$$

其中 $N$ 表示 object model 点的数量 M 是以 object model 搜索 patch 中最近点所生成的点对的集合，$\vec{n}$ 表示点法向量，$p$ 表示点的空间位置。$\alpha$ 是一个参数。对于不满足阈值的类就去除掉，这是为了避免错误的配准对后续步骤造成破坏，对于不确定的聚类留到下一次迭代再考虑。为什么不是越匹配越好？而还要对匹配的得分的标准差做约束：我们实际上是将以某个物体为核心的一大块区域放在一起做配准，匹配程度低很可能只是说明除了核心物体外混入的物体其它物体较多，而并不能说明这个聚类不是我们想要的。例如 Figure **??**比 Figure **??**匹配得分的均值要小就是因为以桌子为主体的 patch 中混入的其它物体较多。

#### 4.3.2 Update Project Matrix By Registration Result

我选择使用 LDA 的方法来更新特征投影的矩阵主要出于以下几点考虑：1. 与初始的聚类方法能够较好的衔接 (实现上只需要更新一个矩阵就好了)。
2.LDA 相当于对不同维的特征赋予不同的权重使得类内方差尽量小而类间方差尽量大，从而在特征子空间中增加不同类别之间的可区分度。Figure **??** 显示了 PCA 投影产生的特征子空间与 LDA 投影产生的特征子空间的对比。

### 4.4 Global Consistent Graph Cut

这一步主要利用之前配准所获得 object model 以及对应的刚体变换为中介来进行重新分割具体的能量项的设计如下：
重新分割操作是在每一帧上单独进行，但是其中 data 项的计算是通过与其它帧的比较来计算获得：
smooth term：

$$E(p_i, p_j, L_i, L_j) = \begin{cases} \sum_{(i,j)\in M} \frac{|\vec{n_i}\vec{n_j}|}{(1+\alpha||p_i - p_j||)(1+\beta||c_i - c_j||)} & L_i \neq L_j \\ 0 & otherwise \end{cases}$$

其中：$\vec{n}$ 表示 superpixel 的法向量 $p$ 表示空间位置，$c$ 表示 RGB 色彩。$\alpha$ 和 $\beta$ 是常数 (目前实际取值就是 1)
data term:

$$E(p_i, L_i) = \begin{cases} C & L_i = 0 \\ ||p_i - p_k|| + \max_N(\frac{||T_{nk}(p_i) - p_{nk}||}{T_{nk}(\vec{n_i})\vec{n_{nk}}}) & L_i = 1...K \end{cases}$$

其中 C 是一个很大的常数，相当于一个门限，倘若 label 成任何一个物体都误差都超过 C 的时候希望被 label 为 0。其中 $p_k$ 是通过 $p_i$ 在当前帧中对应于第 k 个 object 的 patch 中找到的最近点。$T_{nk}(p_i)$ 是根据第 k 个 object 将点的空间位置 $p_i$ 变换到第 n 帧相应的 $T_{nk}(\vec{n_i})$ 是将法向量进行变换。$p_{nk}$ 是与 $T_{nk}(p_i)$ 最近的点。$||p_i - p_k||$ 这一项的加入是为了应对存在多个不运动，或运动始终相同的物体时鼓励将当前点 label 为最近的点所属的物体。

## 5 Behavior Model

Our behavior model in a dynamic scene consists of three parts. One is the behavior of single object, which is represented by the statistics of the object motion (rotations/translations) interpreted from its orientations and positions in frames. The second is the behavior of

(a) Invalid Object
min:0.0494,max:0.6272
mean:0.4763,med:0.5639
var:0.0306,stddev:0.1749

(b) Monitor
min:0.7006,max:0.8442
mean:0.7937,med:0.8033
var:0.0012,stddev:0.0360

(c) Vacuum Sweeper
min:0.5714,max:0.7993
mean:0.7278,med:0.7503
var:0.0031,stddev:0.0555

(d) Invalid Object
min:-0.1001,max:0.5751
mean:0.1733,med:0.1071
var:0.0384,stddev:0.1961

(e) Bed Stand(Mainly)
min:0.2278,max:0.6439
mean:0.5249,med:0.5707
var:0.0134,stddev:0.1159

(f) Chair
min:0.5774,max:0.7348
mean:0.6731,med:0.6736
var:0.0014,stddev:0.0380

(g) Desk
min:0.2860,max:0.4823
mean:0.4130,med:0.4253
var:0.0020,stddev:0.0449

(h) Bed
min:0.5031,max:0.6298
mean:0.5787,med:0.5699
var:0.0012,stddev:0.0350

**Figure 4:** *Registered Object Models and Scores*

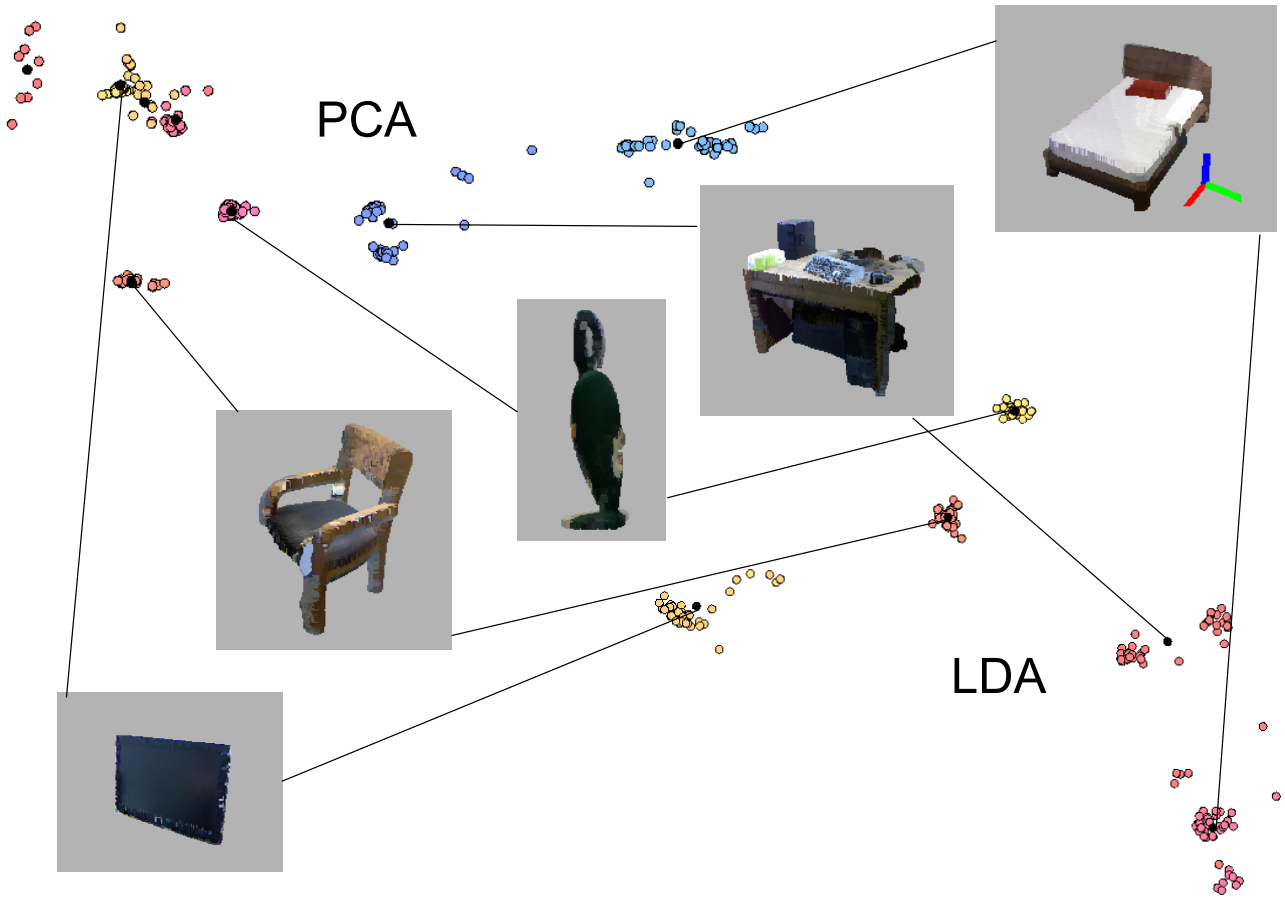


**Figure 5:** *Feature Space Different Projection Matrix*

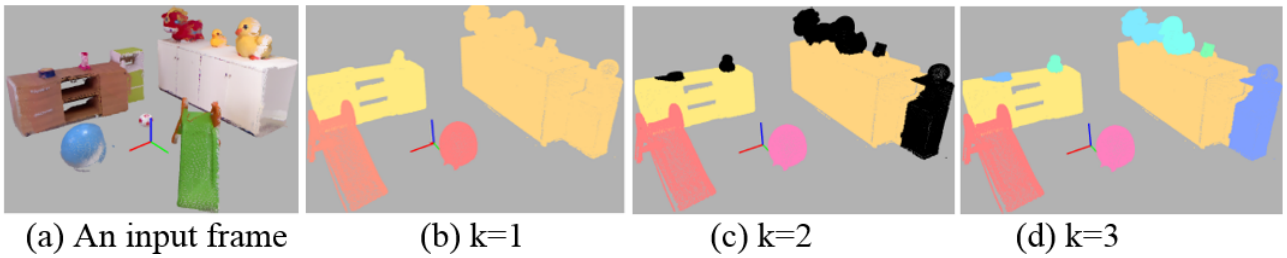


**Figure 6:** *The segmentation of each frame is progressively refined based on registered object models. From left to right: an input point cloud (a), segmentation updates at three iterations. (xuejin: show corresponding object model at each iteration.)*

a group of object, which means the correlations of the statistics of the motions of objects. Their supporting/proximity relationship or arrangements of a group of objects keep consistent. c. The relationship variations caused by motion of the objects in the set of point clouds. In other words, the behavior of the object relationship. For example, the support relationship of vase can change from one desktop to another desktop or so.

### 5.1 Pairwise Object Relations

In a cluttered indoor scene, there are many pairs of objects that usually co-occur in the scene with some typical geometric relations in a long time period. In our system, we consider two types of pairwise relations, which are commonly used to describe object relations in many previous methods [**?**; **?**; **?**; **?**].

**Support** $SR(\mathbf{O}_a, \mathbf{O}_b)$ indicates that $\mathbf{O}_a$ is supported by $\mathbf{O}_b$. We use a Gaussian function to indicate the distribution of $\mathbf{O}_a$ on the supporting surface of $\mathbf{O}_b$; In each frame, we check any pair of objects connected by a surface. Typically, smaller object is supported by the larger one, the upper object is supported by the bottom object. We transform the object been supported $\mathbf{O}_a$ into the local coordinates of its support object $\mathbf{O}_b$ to get its location $\mathbf{x}_a$ and orientation $\theta_a$. According to all the examples of this pair in the input frames, we compute a Gaussian function

$$SR(\mathbf{O}_a, \mathbf{O}_b) = \mathcal{N}_{position}(\mu_x, \Sigma_x)\mathcal{N}_{orientation}(\mu_o, \sigma_o) \quad (1)$$

Each relation has a confidence or frequency $w$ in all the frames.

**Proximity** $PR(\mathbf{O}_a, \mathbf{O}_b)$ indicates $\mathbf{O}_a$ is close to $\mathbf{O}_b$ in with a Gaussian function $\mathcal{N}(\mu, \sigma)$ including position $x, y$ in the local coordinate system of $\mathbf{O}_a$ and the orientation $\theta_b$ of $\mathbf{O}$ in the local coordinate system of $\mathbf{O}_b$, defined as

$$PR(\mathbf{O}_a, \mathbf{O}_b) = \mathcal{N}_{position}(\mu_x, \Sigma_x)\mathcal{N}_{orientation}(\mu_o, \sigma_o). \quad (2)$$

If the background floor and wall are taken into account for pairwise relations, the relation between a single object and the background is actually the behavior of this object itself.

### 5.2 Group Behavior

Based on all the extracted pairwise relation, a directed graph $G(V, E)$ is constructed to encode the mutual relations of all the objects in the scene, as shown in Figure **??**. Each node is an object model in the scene. Each directed edge represents a pairwise geometric relation between two objects. The information carried in each edge includes:

1. **Relation type**: supporting or proximity. Each edge is directed from a small object to an larger object indicating that the small object is usually associated with a larger object.
2. **Edge weight**: is the frequency of that two objects co-occur in the scene. We define it

$$w(i \to j) = \frac{c(i \to j)}{K} \quad (3)$$

where $c(i \to j)$ is the total counts of that two objects co-occur and follow the relation defined by the edge in all the frames. $K$ is the number of all frames.

3. **Relation description**: Associated with each edge, the statistics of the spatial relations defined in Sec. **??** is also stored.

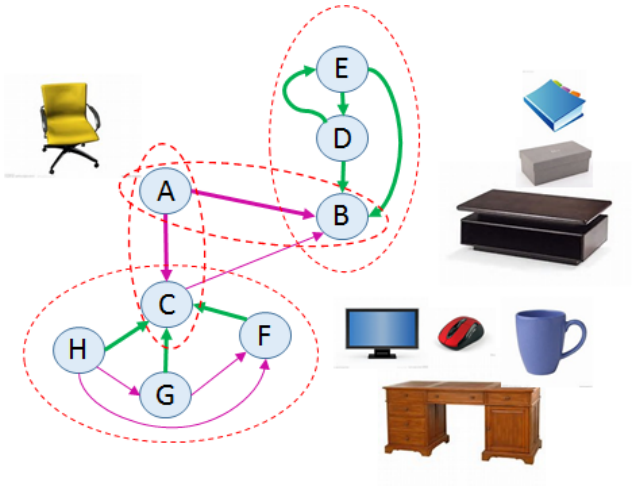


**Figure 7:** *Construction a directed geometric graph from pairwise relations. Green edges indicate supporting relations. Purple edges represent proximity relations. The edge width indicates the frequency of each relation. Each dashed ellipse represents a structural group. The pair (B,C) should be very weak because they are far away from other other, and can be removed. Otherwise, (A,B,C) is another structural group.*

A **structural group** $G_s$ is defined as a *complete graph* in which every pair of nodes in connected by a unique edge, to describe the mutual relations between nodes in a subset. From the directed graph $G(V, E)$, a set of structural groups can be extracted. There are many potential structural groups in $G$. The smallest structural group is an edge connecting two nodes. However, we only focus on reliable structural groups for further application, such as layout re-arrangement, unusual event detection, etc. The reliability of a structural group $G_s(V_s, E_s)$ with $k$ objects is defined as defined as

$$\rho_{sg} = \Big(\prod_{e_i \in E_s} w(e_i)\Big)^{\frac{1}{k}} \quad (4)$$

## 6 Applications

## 7 Results and Discussions

## 8 Conclusions

In this paper, we present

## References

Bouaziz, S., Tagliasacchi, A., and Pauly, M. 2013. Sparse iterative closest point. *Computer Graphics Forum (Symposium on Geometry Processing) 32*, 5, 1–11.

Chang, W., and Zwicker, M. 2011. Global registration of dynamic range scans for articulated model reconstruction. *ACM Transactions on Graphics 30*, 3.

Evangelidis, G., Kounades-Bastian, D., Horaud, R., and E.Z., P. 2014. A generative model for the joint registration of multiple point sets. In *European Conference on Computer Vision (ECCV)*.

Fisher, M., Ritchie, D., Savva, M., Funkhouser, T., and Hanrahan, P. 2012. Example-based synthesis of 3D object arrangements. *ACM Trans. Graph. 31*, 6 (Nov.), 135:1–135:11.

Fisher, M., Savva, M., Li, Y., Hanrahan, P., and Nießner, M. 2015. Activity-centric scene synthesis for functional 3d scene modeling. *ACM Transactions on Graphics (TOG) 34*, 6.

Henry, P., Krainin, M., Herbst, E., Ren, X., and Fox, D. 2012. RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research (IJRR) 31*, 5 (April), 647–663.

Hu, R., Zhu, C., van Kaick, O., Liu, L., Shamir, A., and Zhang, H. 2015. Interaction context (icon): Towards a geometric functionality descriptor. *ACM Trans. Graph. 34*, 4, 83:1–83:12.

Huang, Q., Wang, F., and Guibas, L. 2014. Functional map networks for analyzing and exploring large shape collections. *ACM Trans. Graph. 33*, 4 (July), 36:1–36:11.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. 2011. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *ACM Symposium on User Interface Software and Technology*.

Jia, Z., Gallagher, A. C., Saxena, A., and Chen, T. 2015. 3D reasoning from blocks to stability. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 5, 905–918.

Kim, Y. M., Mitra, N. J., Yan, D.-M., and Guibas, L. 2012. Acquiring 3d indoor environments with variability and repetition. *ACM Transactions on Graphics 31*, 6, 138:1–138:11.

Koppula, H., Anand, A., Joachims, T., and Saxena, A. 2011. Semantic labeling of 3D point clouds for indoor scenes. In *Conference on Neural Information Processing Systems (NIPS)*.

Liu, Z., Tang, S., Xu, W., Bu, S., Han, J., and Zhou, K. 2014. Automatic 3D indoor scene updating with rgbd cameras. *Computer Graphics Forum (Pacific Graphics) 33*, 7.

Merrell, P., Schkufza, E., Li, Z., Agrawala, M., and Koltun, V. 2011. Interactive furniture layout using interior design guidelines. *ACM Trans. Graph. (Siggraph'11)*.

Nan, L., Xie, K., and Sharf, A. 2012. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2012) 31*, 6.

Nathan Silberman, Derek Hoiem, P. K., and Fergus, R. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*.

Nießner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. 2013. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph. 32*, 6, 169:1–169:11.

Savva, M., Chang, A. X., Hanrahan, P., Fisher, M., and Nießner, M. 2014. Scenegrok: Inferring action maps in 3d environments. *ACM Transactions on Graphics (TOG) 33*, 6.

Shao, T., Xu, W., Zhou, K., Wang, J., Li, D., and Guo, B. 2012. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Trans. Graph.*, 136–136.

Su, H., Huang, Q., Mitra, N. J., Li, Y., and Guibas, L. 2014. Estimating image depth using shape collections. *ACM Trans. Graph. 33*, 4 (July), 37:1–37:11.

Wand, M., Jenke, P., Huang, Q., Bokeloh, M., Guibas, L., and Schilling, A. 2007. Reconstruction of deforming geometry from time-varying point clouds. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SGP '07, 49–58.

Xiao, J., and Furukawa, Y. 2012. Reconstructing the world's museums. In *Proceedings of the 12th European Conference on Computer Vision*, ECCV '12.

Xu, K., Chen, K., Fu, H., Sun, W.-L., and Hu, S.-M. 2013. Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics 32*, 4, 123:1–123:12.

Xu, K., Ma, R., Zhang, H., Zhu, C., Shamir, A., Cohen-Or, D., and Huang, H. 2014. Organizing heterogeneous scene collection through contextual focal points. *ACM Transactions on Graphics, (Proc. of SIGGRAPH 2014) 33*, 4, to appear.

Xu, K., Huang, H., Shi, Y., Li, H., Long, P., Caichen, J., Sun, W., and Chen, B. 2015. Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia 2015) 34*, 6, to appear.

Yan, F., Sharf, A., Lin, W., Huang, H., and Chen, B. 2014. Proactive 3d scanning of inaccessible parts. *ACM Transactions on Graphics(Proc. of SIGGRAPH 2014) 33*, 4.

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