Acquisition of 3D Indoor Environments with Variability and Repetition

Young Min Kim* Stanford University
Niloy Mitra† UCL
Dongming Yan‡ KAUST
Leonidas Guibas§ Stanford University

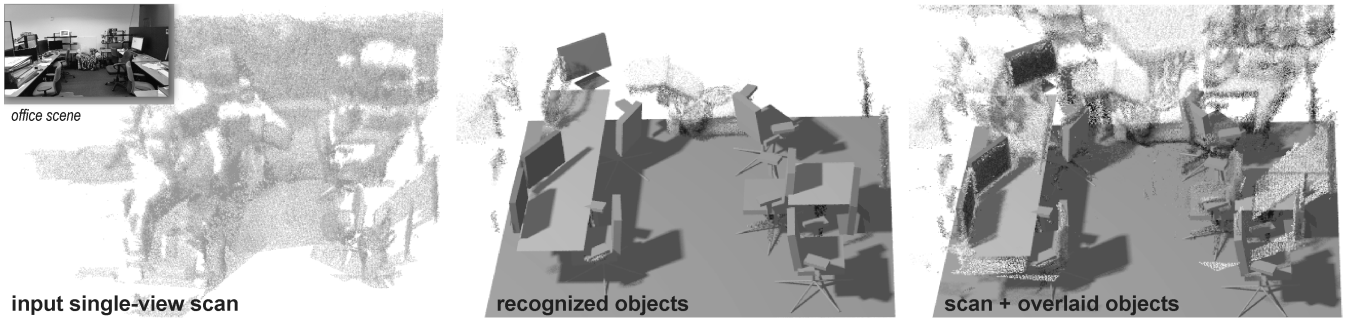

Figure 1: *(Left) Starting from a single view scan of a 3D environment obtained using a fast range scanner, we perform scene understanding by recognizing repeated objects, while factoring out their modes of variability (middle, right). The repeating objects have been learned beforehand as low complexity models, along with their variability modes. We extract these objects despite a poor quality input scan with large missing parts and many outliers. For reference, we also show a scene photograph, which is not available to the algorithm.*

Abstract

Large-scale acquisition of exterior urban environments is by now a well-established technology, supporting many applications in search, navigation, and commerce. The same is not true for indoor environments, however: access is often restricted and the spaces may be cluttered. In addition, such environments typically contain a high density of repeated objects (e.g., tables, chairs, monitors, etc.) in regular or non-regular arrangements with significant pose variations and articulations. In this paper, we exploit the special structure of indoor environments to accelerate their 3D acquisition and recognition with a low-end handheld scanner. Our approach runs in two phases: (i) a learning phase, where we acquire 3D models of frequently occurring objects and capture their variability modes from only a few scans, and (ii) a recognition phase, where from a single scan of new areas, we identify previously seen objects, but in varying poses and locations. This greatly accelerates the capture process (average recognition time of 200ms/model). We demonstrate our framework with the acquisition of typical areas of a university building including cubicle or desk areas, auditoriums, etc., using a Microsoft Kinect sensor.

1 Introduction

Significant advances have been made towards mapping the exteriors of urban environments through large-scale city capture efforts of Google, Nokia, Microsoft, etc. Acquiring 3D indoor environments in private and public office buildings, however, remains challenging. While sensor-instrumented vehicles can drive down streets to capture exterior spaces, mimicking similar setups for interior acquisition requires customization, manual intervention, and is cumbersome due to unreliable GPS signals, odometry errors, etc. Further challenges arise due to extensive variability commonly encountered in building interiors: doors and windows open and close, chairs get moved around, cubicles get re-arranged, etc.

*e-mail: ymkim@stanford.edu
†e-mail:n.mitra@cs.ucl.ac.uk
‡e-mail:yan@kaust.edu.sa
§e-mail:guibas@cs.stanford.edu

At the same time, inexpensive range cameras (e.g., the Microsoft Kinect) are easy-to-use, fast, and quickly becoming ubiquitous. This opens new possibilities for large-scale indoor acquisition. High frame-rate and increased portability, however, come at the cost of resolution and data quality: the scans are at best of modest resolution, often very noisy, invariably contain outliers, and suffer from missing parts due to occlusion. Thus, although we can very quickly acquire large volumes of data, there exists no general algorithm to extract high-level scene understanding. Further, unlike building exteriors whose facades are largely flat and have ample clearance for scanning, indoor objects are usually geometrically complex, contain significant articulations, and are found in cramped surroundings. Thus, a traditional acquisition pipeline is ill-suited: in a typical setting, one has to scan the scene multiple times from various viewpoints, semi-automatically align the scans, and finally construct a 3D model. The process is further complicated when the model deforms during acquisition.

We exploit three observations to make the difficult problem of indoor 3D acquisition much more tractable:

(i) Most office or public building indoor environments are comprised of basic elements, such as walls, doors, windows, furniture (chairs, tables, desks, lamps, computers, cabinets, etc.), which come from a small number of prototypes and repeat many times.

(ii) Such building components are generally formed of rigid parts whose geometry is often locally simple, i.e., they consist of surfaces that are well approximated by planar, cylindrical, conical, spherical proxies. Further, although variability and articulation are dominant (a chair moves or its base rotates, a door swings, a lamp is folded), such variability is limited and low-dimensional (e.g., translational motion, hinge joint, telescopic joint, etc.).

(iii) Mutual relationships among the basic objects satisfy strong priors (e.g., a chair stands on the floor, a monitor rests on the table).

In this paper, we present a simple yet effective pipeline to acquire models of indoor objects such as furniture, together with their variability modes, and discover object repetitions and exploit them to speed up large-scale indoor acquisition towards high-level scene understanding.

Our algorithm works in two stages. First a *learning* phase where, starting from a few scans of individual objects, we construct

primitive-based 3D models while explicitly recovering respective joint attributes and modes of variation. We formulate the problem of object acquisition as deciding on the types of primitives required for describing the object model and their mutual connections from a finite set of possible options, as well as recovering the small number of necessary variability parameters defining the object pose. This pre-processing phase of the approach is relatively expensive, but applies only to a small fraction of the geometry present. Second a fast *recognition* phase (about 200ms/model) where, starting from a *single*-view scan, we perform segmentation and classification into plausible objects and then recognize the objects involved and extract the pose parameters for the low complexity models generated in the learning phase. Intuitively, we use priors for primitive types and their connections, thus greatly reducing the number of unknowns to enable model fitting even from very sparse and low resolution datasets, while solving for part association via a novel Markov Random Field (MRF) formulation. Further, we demonstrate that simple inter-object relations and their relative placements greatly simplify segmentation and classification tasks necessary for high-level scene understanding. Effectively, we progressively model the surrounding environment and its modes of variability. As new data appears, we either explain it with our current scene model, or we incorporate the unexplained data by updating current models, or by creating new ones. Thus, we leverage the repeatability of common indoor environments and populate the models with instances of already acquired objects.

We demonstrate our method on a range of challenging real-world scenarios. We present, for the first time, basic scene reconstruction for massive indoor scenes (e.g., office desk spaces, building auditoriums) from unreliable sparse data by exploiting the low-complexity variability of common furniture objects and their repetitions. Interestingly we can now meaningfully detect changes in an environment. For example, we can hope to detect a new object places in a deskspace by rescanning despite articulations and motions of the previously extant objects (e.g., desk chairs, monitors, lamps). Thus, we can separate (and ignore) nuisance modes of variability (e.g., motions of the chairs, etc.) from variability modes that carry importance in an application (e.g., security, where the new object may be a threat).

Contributions. In summary, we (i) introduce a pipeline to acquire proxy models of common office furniture consisting of rigid parts, as well as of their low-dimensional variability modes, (ii) detect and recognize occurrences of such models from single low quality scans, and (iii) quickly populate large indoor environments with variability and repetition enabling novel recognition possibilities.

1.1 Related Works.

Surface reconstruction from unorganized point samples has been extensively studied in computer graphics, computational geometry and computer vision (see [Dey 2007] and references therein). Our main goal, however, is different. We focus on acquiring and understanding large 3D indoor environments.

Scanning technology. Powered by recent developments in real-time range scanning, everyday users can easily acquire 3D data at high frame-rates. The individual frames, however, are poor in quality. Hence, researchers have proposed algorithms to accumulate multiple scans for better quality acquisition [Henry et al. 2010; Izadi et al. 2011]. Unfortunately, such methods lead to ghosting artifacts if the camera or the scene moves abruptly in course of scanning. Furthermore, the raw scans do not provide any high-level understanding of the scene.

Scan processing. Rusinkiewicz et al. [2002] first demonstrated the possibility of real-time lightweight 3D scanning. In their framework, the user rotates a handheld object while the system continuously updates the model to provide real-time visual feedback

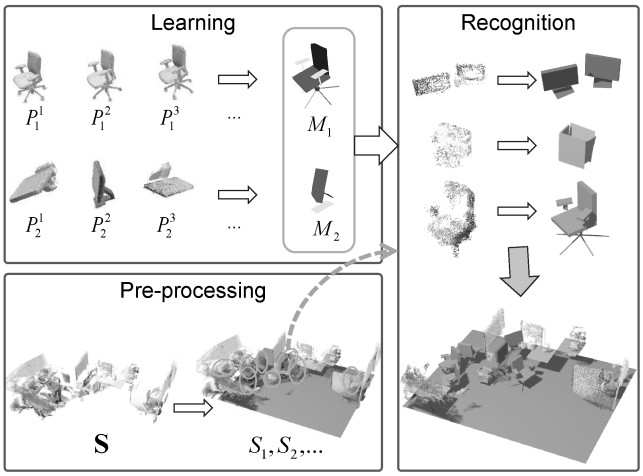

Figure 2: *Our algorithm consists of two main stages: (i) a learning phase, and (ii) a fast recognition phase, which takes average of 200 ms/model.*

guiding the user where to scan next. In other related efforts, researchers have used template models to learn the space of human bodies [Allen et al. 2003], morphed database models to fill in regions of missing parts [Pauly et al. 2005], used non-rigid alignment to better align (warped) multiple scans [Li et al. 2009], or exploited non-local repetitions to consolidate point clouds for urban facades [Zheng et al. 2010]. The methods, however, are not suitable for extracting a high-level scene understanding along with appropriate deformation models from sparse and unorganized inputs.

Shape analysis. Man-made objects populating indoor scenes typically have low-degree of freedom and are often arrangements of simple primitives. Schnabel et al. [2007] introduce an algorithm to automatically extract basic shapes (e.g., planes, cylinders, spheres, etc.) from unorganized point clouds. Subsequently, the GlobFit [Li et al. 2011] framework extracts a set of mutually consistent relations (e.g., coplanar, coaxial, equal length, etc.) and conform to the recovered relations for reverse engineering. Alternately, temporal information across multiple frames can be used to additionally track joint information in order to recover a deformation model ([Chang and Zwicker 2011] and references therein).

In the context of image understanding, Lee et al. [2010] construct a box-based reconstruction of indoor scenes using volumetric considerations, while Gupta et al. [2010] apply geometric constraints and mechanical considerations to obtain a block-based 3D scene model. In the context of 3D scans, there has been little efforts towards scene understanding of large datasets. Notable exceptions include: Triebel et al. [2010] present an unsupervised algorithm for segmentation and object detection in indoor scenes. They apply a graph-based clustering on pre-segmented input data and assign part labels using a Conditional Random Field (CRF). The method, however, does not consider object variability and cannot be applied to unorganized pointsets, as is our goal. Boyko et al. [2011] extracts high level information of road in noisy outdoor point sets. We also extract high level information in the context of indoor environments for quick and effective scene understanding.

2 Overview

Our framework works in two main stages: learning and recognition.

In the *learning stage*, we scan each object of interest a few times (typically 5-10 scans over different poses). Our goal is to con-

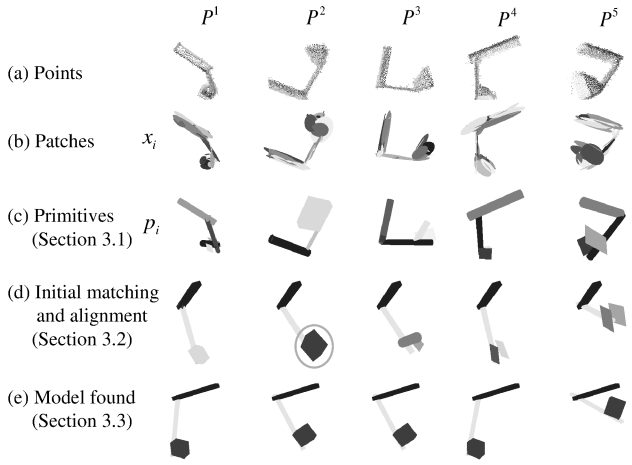

Figure 3: *Learning stage with lamp data set. Given the set of registered point clouds (a), in the learning stage we find a coherent model to explain the data jointly. Starting from individual primitives patches (b, c), the stable and consistent primitives are detected in the first iteration (d). Using such primitives to define a reference frame, we establish correspondence across the other primitives, and extract the (joint) deformation parameters. The remaining parts are then replaced by the best configuration among the sets (e). The process repeats until convergence.*

sistently segment the scans into parts as well as identify the junction between part-pairs to recover the respective junction attributes. This is an ambitious goal, given the quality of the inputs. We use two scene characteristics: (i) the objects are well approximated by a collection of simple primitives (e.g., planes, boxes, cylinders) and (ii) the types of junctions between such primitives are limited (e.g., hinge, translational) and of low-complexity. We first recover a set of primitives for each individual scan. For each object, we collectively process its scans to extract a primitive-based proxy representation along with the necessary inter-part junctions. Thus we obtain a collection of models $M_1, M_2, \ldots$.

In the *recognition stage*, we start from a single scan $\mathbf{S}$ of the scene. We first extract the dominant planes in the scene – typically they denote the ground, walls, desks, etc. We identify the ground plane by using the (approximate) up-vector from the acquisition device and noting that the points lie to one side of the ground. Planes parallel to the ground are tagged as tabletops if they are at heights as observed in the training phase (typically 1-3') — here we exploit that working surfaces have similar heights across rooms. We remove the points associated with the ground plane and the candidate tabletops, and perform connected component analysis on the remaining points (on a neighbor graph) to extract pointsets $S_1, S_2, \ldots$.

For each pointset S_i, we test if it can be satisfactorily explained by one of the object models M_j. This step, however, is challenging since the data is unreliable and can have large geometric variations due to pose changes. We handle such difficulties using a novel MRF formulation and incorporating simple scene priors: (i) we use placement relations (e.g., monitors are placed on desks, chairs rest on the ground, etc.), (ii) each model has allowable repetition modes (e.g., monitors are usually horizontally repeated, chairs are repeated on the ground). We assume such priors are available as domain knowledge (see also [Fisher et al. 2011]).

3 Learning Phase

The input to the learning phase is a set of registered point clouds $P^1, \ldots, P^n$ derived from the same object in different configurations. Our goal is to build a model M comprising of parts that are linked by joints. Note that although this stage can be unstable for very few scans, but once we have sufficient data to construct model M, the subsequent recognition stage becomes simple, robust, and interactive. During the learning phase, we segment each point set P^i into a collection of primitives p^i_j. We then establish correspondence among the parts across the scans, and from the matched parts build model M. Note that we also store transformations $\mathcal{T}^i_j$ between individual parts of the extracted model and the corresponding parts in the measurement, i.e., $\mathcal{T}^i_j(p^i_j) \approx m_j$. We refine the primitives by jointly fitting the matched parts across different measurements using the transformation $\mathcal{T}^i_j$ (see Figure 3).

In this work, we restrict the choice of primitives to cylinders and boxes, and joint types to rigid, telescopic, and rotational. The final model M contains information of the individual part primitives and possible deformation across them (cf., [Chang and Zwicker 2011]). In addition, we can also keep necessary features for robust matching. For example, the distribution of height from the ground plane can be a strong prior for tables, or objects can have preferred repetition direction, e.g., monitors or auditorium chairs are typically repeated sidewise. These learned attributes and relationships act as reliable regularizers in the recognition phase, when data can be very sparse and noisy.

3.1 Individual primitive fitting. Primitive fitting can alternately be thought of a segmentation problem: Given a set of measurements, we have to partition the data so that each partition is well explained by a primitive. Instead of fitting primitives directly to points, we partition the points into a set of surface patches $(x_1, x_2, ..., x_n)$, and fit primitives to such patches. We obtain initial surface patches by iteratively sampling seed points and growing the patches to local planar patches [Cohen-Steiner et al. 2004]. Such patches sufficiently approximate the surface, and also reduce the complexity of the problem. After convergence, we perform PCA for each patch x_i and keep the eigenvalues $\sigma_1(x_i) \geq \sigma_2(x_i) \geq \sigma_3(x_i)$ and corresponding eigenvectors $\mathbf{e}_1(x_i), \mathbf{e}_2(x_i), \mathbf{e}_3(x_i)$, respectively.

We then perform fitting using RANSAC directly on the patches. Starting with larger patches as seeds, we make a guess for the primitive parameters, and progressively add neighboring patches. We iterate between finding candidates and finding the parameters, and retain those with sufficient witness, i.e., inlier patches thus effectively grouping the initial patches to bigger ones. The initialization from patches differ for each primitive type as follows:

Algorithm 1 Incrementally complete a coherent model M

```
while |P̃^i_M| > 0, ∀i do
  for p^i_j ∈ P̃^i_M do
    find a connected part p^i_k ∈ P^i_M
    if such p^i_k exists then
      count number of points in all measurements explained by
      the part p^i_j using T^i_k
      keep the part that has the maximum count
    end if
  end for
  add the best candidate p^i_j to M and the corresponding transformation T^i_j to T^i.
  add rotation or scaling if any of the measurements cannot be
  explained by the part without deformation.
  remove parts in P̃^i_M that are covered by the new part.
end while
```

Box. We parameterize a box with three mutually orthogonal directions $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3 \in \mathbb{R}^3$, offset values $p_1, p_2, p_3 \in \mathbb{R}$, and lengths $l_1, l_2, l_3 \in \mathbb{R}$ along the respective directions. We initialize a box if (i) a patch that has small normal variation $\sigma_3(x_i) < \delta$ ($\delta = 0.01$), or (ii) a pair of patches are nearly orthogonal, i.e., $\mathbf{e}_1(x_i)^T \mathbf{e}_1(x_j) \approx 0$. In case (i), we initialize a direction $\mathbf{d}_1$ and an offset p_1. When a neighboring patch with a right angle is detected (also for the initialization case (ii)), then we replace the two directions $\mathbf{d}_1, \mathbf{d}_2$ and corresponding offsets p_1, p_2 by the actual values from the new patches.The remaining direction is given by $\mathbf{d}_3 = \mathbf{d}_1 \times \mathbf{d}_2$. Otherwise, the directions and lengths are approximated by principal components attributes, i.e., $\mathbf{e}_1(x_i)$ and $\mathbf{e}_2(x_i)$.

Cylinder. We fit a cylinder when a patch is not necessarily planar, i.e., $\sigma_3(x_i) > \delta$ and elongated, i.e., $\sigma_2(x_i)/\sigma_1(x_i) < \rho$ ($\rho = 0.5$). The cylinder axis is initialized using $\mathbf{e}_1(x_i)$ and the size of the patch, while the remaining directions provide a starting radius. Any neighboring patches that are close to the axis direction are also added.

3.2 Matching. After the individual measurements P^i are fitted with primitives $\{\mathbf{p}_j^i\}$, we find relatively large, consistent set of primitives among the measurements as a starting point for the model. We compare adjacent pairs of primitives and calculate how well they overlap, and pick the one with the most overlap. Thus, we give preference to larger parts, since smaller primitives arise from unstable fits and are unreliable at this stage.

We observe that a pair of primitives with their relative positions form a local reference frame. We add the pair to current model M and align the measurements by adjusting the parameters by jointly fitting the aligned primitives. We then replace the matched primitives by the model primitives with the transformation $\mathcal{T}_j^i \in \mathbf{T}^i$. Note that measurement P^i can have multiple transformations since each part can deform relative to each other (see Figure 3d).

3.3 Completing a coherent model. Primitives in measurement $p_j^i \in P^i$ are either mapped to the current model P_M^i or are tagged as unexplained $\tilde{P}_M^i$. After the previous matching step discovers a pair of parts, we start from $|P_M^i| = 2$. The matched parts serve as anchors for further matches among the remaining candidates, which are progressively attached to already matched parts (see Algorithm 1).

Man-made structures are often hierarchical. Hence, in the inner loop of testing each part p_j^i, we benefit from relative information between parent node and its connected children to find a regular structure. We consider mirror and rotational symmetries in our framework (see Figure 3e). We now describe how to consider such regularity in the testing stage.

Mirror symmetry. We test for mirror symmetry when the parent node is a box, which has 3 possible reflective symmetry planes. If the connected child parts can be separated by one of the symmetry planes for all n measurements, we flag a candidate mirror symmetry. Once the symmetry plane and the candidate parts are flagged, we similarly test each configuration with respect to measurement. If there were n measurements of the object, there are $2n$ sections that can be fitted by the same set of primitives under mirror symmetry. The symmetric parts can also undergo possible deformations. We use the best configuration to replace all the detected symmetric instances (see Figure 4c).

Rotational symmetry. In case of rotational symmetry, instead of looking for the plane of reflection, we search for the axis of rotation. For any pair of patches that are neighboring and not parallel to each other, we find the approximate point, or possible axis position. Then we run mean-shift clustering to detect candidate axis positions among all pairs of neighboring patches. If there is a coherent axis position for all the measurements, then there is possible rotational symmetry. Hence, after testing, we jointly fit a rotational structure.

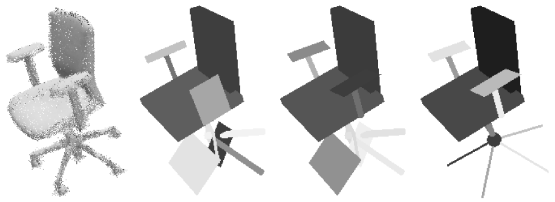

Figure 4: *The chair model starts from fitting primitives and detecting the dominant elements, i.e., the back and the seat (b), then evolves as the algorithm discovers and jointly optimizes the shape for mirror symmetry between the elements, i.e., arms (c), and rotational symmetry for the legs (d).*

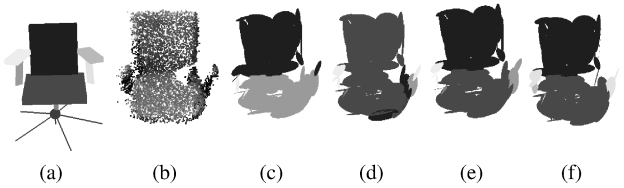

Figure 5: *In the recognition phase, we use a MRF formulation to match the learned chair model (a) to the input scan (b). While simply matching height (c) or relative size (d) results in wrong matches, we correctly identify the chair back and seat (e) using both the terms. The most likely assignment adding binary term is shown in (f). Note in areas of very sparse data, there can still be mismatches, requiring further refinements, especially for small parts, e.g., we fail to extract symmetry between the chair arms in this example.*

We fix the axis location but allow scale or rotational deformation along the axis direction (see Figure 4d).

4 Recognition Phase

Having learned a set of models (along with their deformation behavior) $\mathbf{M} := \{M_1, \ldots, M_k\}$ for a particular environment, in the recognition phase we can quickly collect and understand the environment. The scene $\mathbf{S}$ containing the learned models is collected using the code by [Engelhard et al. 2011] in a few seconds of time. As pre-processing, we extract large flat surfaces using RANSAC, which effectively are the ground plane, desk tops, and walls. We retain the plane equation of the most dominant plane (ground plane) and the (significant) ones parallel to the dominant plane (desks). The remaining points are partitioned into groups by connecting neighboring points (using a neighborhood graph with distance threshold of 1cm). The resulting clusters $\mathbf{S} = \{S_1, \ldots, S_n\}$ are the input for our recognition pipeline (see Figure 2).

For each clustered single view point cloud S_i, we try to match each of learned models M_j. The recognition pipeline returns how well S_i is matched to the given model M_j in addition to the relative transformation and deformation. If successful, we build a quick and lightweight 3D model of the environment by simply replacing the corresponding pointcloud by the matched model with correct transformation and deformation parameters (see Figure 5). For the rest of the section, we simply denote scan and model as S and M, respectively.

4.1 Initial match. Recall the model M is composed of a set of primitives and the connectivity between them. Similar to the learning step, we create patches for the input cluster S and generate initial

Figure 6: *Recognition results on synthetic scan of a virtual 3D scene.*

segments and candidate joints between neighboring segments. We jointly consider the deformation parameters and compare each pair of neighboring parts with joints in models via a Markov Random Field (MRF) formulation. Say there are segments $s_1, s_2 \in S$ and parts $m_1, m_2 \in M$. Then we find the most likely matches between them by minimizing the following energy function.

$$\mathcal{E} := \underbrace{\sum_i \mathcal{D}(s_i = m_i)}_{\text{unary term}} + \eta \underbrace{\sum_{i,j} \mathcal{V}(s_i = m_i, s_j = m_j)}_{\text{binary term}} \tag{1}$$

We can use the distance $d(f_s, f_m) \in [0,1]$ between features of $s \in S$ and $m \in M$ for the energy function. For example, the unary term can be written as sum of all possible feature distances:

$$\mathcal{D}(s = m) := \sum_i d^i / n, \text{ where} \tag{2}$$

$$\begin{aligned} d^{length}(f_s, f_m) &= \min\left(|f_s - f_m| / f_m, 1\right), \\ d^{angle}(f_s, f_m) &= |f_s - f_m| / \pi, \\ d^{height}(f_s, f_m) &= 1 - \sum_i \min\left(f_s(i), f_m(i)\right). \end{aligned} \tag{3}$$

Among many possible sets of features, we use relative length and height distribution. Height distribution is given as a histogram over the distance from the ground plane and the distance is obtained as a histogram intersection distance [Swain and Ballard 1991]. Note that it is easy to integrate alternate shape features, colors, or any other useful information to strengthen the recognition power. In all our tests, however, we only consider geometry and ignore texture information.

In our framework, we can robustly handle moderate and low quality scans reliably because of the binary term. We define this term by comparing features distances for neighboring patch pairs with joints in a model,

$$\mathcal{V}(s_1 = m_1, s_2 = m_2) := \begin{cases} \sum_i b^i / n & \text{if edge exists in model} \\ \psi & \text{if no edge exists.} \end{cases} \tag{4}$$

Due to the limitation of single-view data, not all of the joints are likely to be observed. The term $\psi > 1$ (we used 5) is a penalty for assigning non-existing joints, which captures the probability that a joint may not have been observed, or be wrongly assigned. The features b^i we consider for the binary term include relative location of contact, angle between the parts, length (the largest dimension from the contact point), width (the smallest dimension from the contact point). Since we know the possible deformation space of model joints from the model M, we adaptively compare only features that are invariant to the deformation. For example, we ignore angles for rotational joints, or ignore length of corresponding directions for telescopic joints.

Since the energy function is non-metric and non-convex, we used a message passing algorithm [Kolmogorov 2006] to solve for the minimum. The minimization is quick because the possible discrete assignment space is quite small. The result certainly depends on the individual terms used (see Figure 5). We find the initial transformation between S and M by using the matched joints with the smallest binary term $\mathcal{V}$. If there is no joint discovered, we use the part with smallest unary term $\mathcal{D}$ and relative ground direction.

4.2 Finding transformation and deformation parameters. After the initial match and transformation is extracted, we incrementally grow the match to find the local minimum for the remaining unknowns. Starting from the initial transformation, we find the best deformation parameter, and establish correspondence between S and M. We find the best transformation given the correspondence, and then iterate until convergence. We replace the points by the best matching model, if at least 80% of points are explained after convergence.

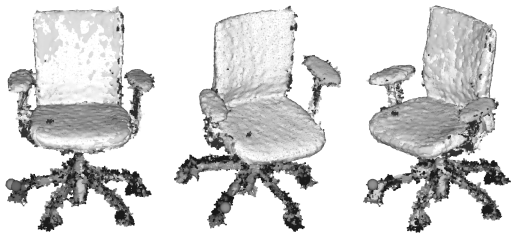

Figure 7: *Chang et al. [2011] can find meaningful rigid parts in the learning stage for good scans, and such a model can be used directly in the recognition stage. In our scenario, however, we found a proxy-based method to be more forgiving to poor data.*

5 Results

In this section, we present the results of our system on various synthetic and real-world scenes. For real-world datasets, we used Microsoft Kinect with an open source scanning library [Engelhard et al. 2011] for both the learning and the recognition phases. Note that a more accurate scanner can be used in the learning phase for better quality models. For example, in Figure 7, we show the results of Chang et al. [2011] on higher density scans of multiple chair poses. Although their current method is limited to hinge and ball joints, and hence miss the telescopic joints in the chair example, we believe their method can be generalized to be used in the learning stage. However, for sparse input scans, like in our setting, we found our primitive based learning method to be more robust. Intuitively,

scene	model	points per scan	no. of scans	no. of prim.	no. of joints
office	chair	41724	7	8	4
	monitor	20011	5	3	2
	trash bin	28348	2	4	0
auditorium	chair	31534	5	4	2
synthetic	chair	28445	7	10	4
	stool	19944	7	3	2
	monitor	60933	7	3	2

Table 1: *Models obtained from the learning stage (see Figure 9).*

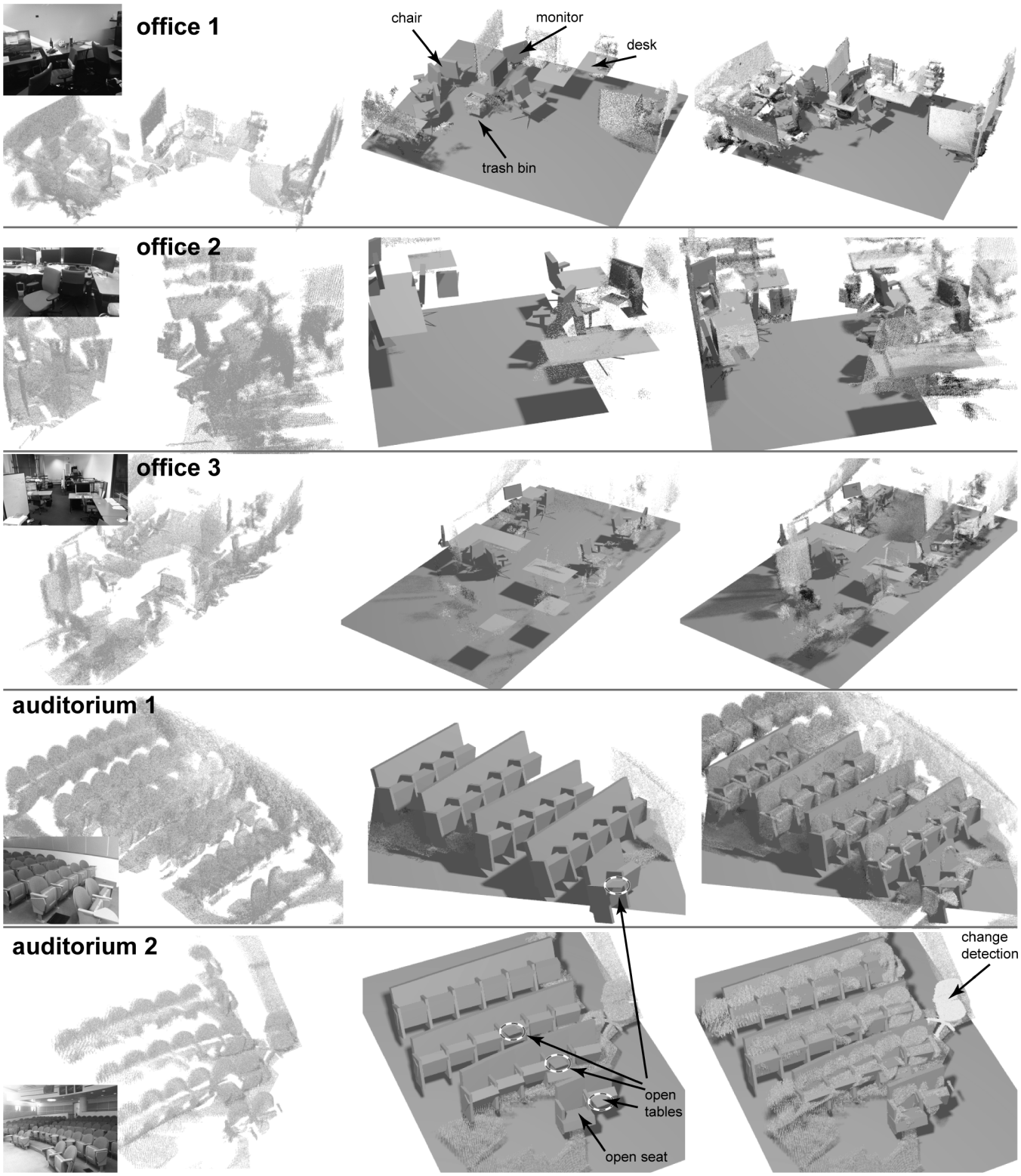

Figure 8: *Recognition results on various office and auditorium scenes. Since the input single view scans are too poor to understand the scene complexity, we include scene images just for visualization (they are not available to the algorithm). Note that for the auditorium examples, we even detect the tables on the chairs — this is possible since we have extract this mode of variation for chairs in the learning phase.*

we restrict the complexity of the model by our choice of primitives and joint types, and thus regularize the problem.

Comparison. We struggled to compare our method with existing techniques as we are not aware of any competing method that can handle the quality of our target data and at this scale. Even state-

scene	number of input points			objects	objects
	average	min	max	present	detected*
teaser	6187	2575	12083	5c 6m 0t	5c 4m 0t
office 1	2570	1129	3930	5c 2m 1t	5c 2m 1t
office 2	4952	1996	11090	4c 6m 2t	4c 5m 1t
office 3	2450	1355	5234	8c 5m 0t	6c 3m 0t
aud. 1	19033	11377	29260	26 chairs	26 chairs
aud. 2	9381	2832	13317	21 chairs	19 chairs
synthetic	3227	1168	9967	5c 3s 5m	5c 3s 5m

*c: chair, m: monitor, t: trash bin, s: stool

Table 2: *Statistics for the recognition stage. For each scene, we also indicate the corresponding scene in Figure 8 in parenthesis, when applicable.*

of-the-art methods like [Chang and Zwicker 2011] assume input quality well beyond the ones we process (see Table 2). A direct surface reconstruction on the input single view point sets is clearly unrealistic — even we as humans find it hard to visually understand the scene from input scans as shown in Figures 1 and 8. Further, scan consolidation [Zheng et al. 2010] did not help since we failed to detect the repetitions in the original scans due to the presence of model variabilities and non-regular arrangements. In fact, we failed to reliably consolidate the models even after the recognition stage because of ambiguity of point-part association near segment boundaries, especially in noisy and incomplete regions. In our experience, the low complexity deformation models extracted in the learning stage are critical: for example, using simple rigid models (by freezing the joints in the respective models) leads to significantly worse recognition results in the office scenes where the objects are all in different configurations (e.g., no two chairs have the same height).

Synthetic scenes. We evaluated our framework on a synthetic scene obtained from Google warehouse (see Figure 6). Note that we detected all the 5 monitors, the 5 chairs, and the 3 stools in the scene, along with their poses, the associated ground plane, and the desk area. The stools were particularly challenging given their thin stems, but we recovered their positions since (i) they stand on the ground, and (ii) their top and bottom parts provide enough support for the MRF to recover their poses. Our experience with other scans of the scenes was similar, and we only show one variation here. In scenes where the top part of the stools were not visible, we failed to detect them.

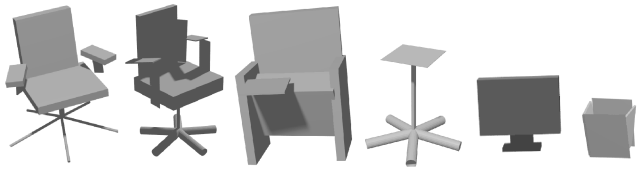

Figure 9: *Various models (chairs, stool, monitor, trash bin) learned and used in our setups (see Table 2).*

Real-world scenes. We tested our pipeline on a range of real-world examples each consisting of multiple objects arranged over large spaces. These are challenging especially due to the amount of variability in the individual model poses. Table 1 summarizes all the models learned for these scenes ranging from 3-10 primitives with 0-4 joints, learned from only a few scans (see Figure 9).

The recognition results are satisfactory, e.g., in Figure 1 we detect all the 5 chairs and the 4 monitors, although parts of the desktops go missing. Note that the sofa, which was not among the learned models, is not detected. The complexity of our problem setting can be appreciated by looking at the input scan, which is hard even for us to visually parse. We overlay the unresolved points on the recognized parts for comparison. Note that we show the colored points just for reference, while our algorithm only has access to the geometry but not any color or texture attributes.

After the preprocessing (learning) phase, our recognition stage is lightweight and fast taking on an average 205ms to compare a point cluster to a model on a 2.4Hz CPU with 6GB RAM. We summarize the results in Figure 8 for (cluttered) office setups and auditoriums. We detect the chairs, monitors, and trash bins across different rooms, and rows of auditorium chairs in different configurations. Surprisingly, we could also detect the small tables in the two auditorium scenes (1 in auditorium #1, and 3 in auditorium #2). Also notice the detected pose changes in the auditorium seats. Even under such demanding data quality, we can recognize the models and recover poses from data sets an order of magnitude sparser than those required in the learning phase. Specifically, we need around 1000-2000 points per model for recognition. This is possible since the learning stage extracts only the important degrees of variation, thus providing a very compact, yet powerful, model (and deformation) abstraction.

Overall the recognition results are satisfactory in most cases (see Table 2) except when an individual model gets split into multiple parts by occlusion, or objects are very distant (note that the data quality deteriorates nonlinearly with distance) and the resulting point clouds are severely distorted. In the office scenes, we fail to detect the white boards as they are confused with the walls. We also successfully detect the trash cans in office #1 and office #2 — note the size of the objects relative to the scan quality (and resolution). In office #3, we miss a couple of the chairs, which are mostly occluded and beyond what our framework can handle. In the auditorium scenes, although we detect all the chairs (along with the open tables), we fail to extract the slight arc of the chair arrangements. Such an error can only be corrected with more global reasoning. We decided against considering such specialized priors as they are very domain dependent. It is, however, possible to integrate a coupled global optimization to refine our recognition results in the auditorium case (see grid-based refinement in [Pauly et al. 2008]).

As an interesting possibility, we can also use our pipeline to efficiently detect *change* — by change, we mean introduction of a new object, previously not seen in the learning phase while factoring out variations due to different spatial arrangements or variation of individual model poses. For example, in the auditorium #2, a previously unobserved chair is successfully detected (flagged in yellow). This can be particularly useful for surveillance and automated investigation of indoor environments or for disaster planning (where it is unsafe for a human observer to go in).

Limitations. Clearly we cannot capture features that are below the sensor capabilities, especially small/thin protruding structures (e.g., wheels of chairs) or reflective objects. For example, we fail to correctly detect the pose for chairs with very thin legs, though we can estimate their heights based on their relation to the ground. Monitors in tilted angles also cannot be detected because the sensor could not provide reliable measurements due to the material properties. We are also limited in the learning phase. Given our part-based segmentation, we fail to correctly recognize parts of models when the segmentation of parts is ambiguous in certain configurations, e.g., a cabinet with closed doors. Finally, at present we cannot handle complex joint types (e.g., folding chairs) again primarily due to segmentation limits.

6 Conclusion

We presented a simple system for recognizing models in cluttered 3D indoor environments, while factoring out deformation and spatial placement, at a scale previously not demonstrated (to the best of our knowledge). Our pipeline is scalable and general with extension to more complex environments primarily requiring reliable acquisition of additional object models (with their variability modes) and good priors for the inter-object relationships.

Several future possibilities remain: (i) With increasing number of object prototypes, we will need more sophisticated search data structures. We hope to benefit from the existing literature in shape search and related areas. Scalability at the level of big building interiors is probably best tested in a company environment (with human resources), although we believe that given the simplicity of the setup a lot can be done at a smaller scale. (ii) In this work, we focused on a severely restricted form of sensor input, namely poor and sparse geometry. We intentionally left out color and texture, which can be very useful in future considerations, especially if appearance variations can be accounted for. (iii) Finally, we have to aware of the ultimate use of the acquired models — whether the goal is the production of interior CAD models for visualization, or more schematic representations that may be sufficient for navigation, or simply of scene understanding for threat detection, location of missing objects, etc. In all these settings the ultimate representation of the learned objects can vary, but the basic pipeline of coupled object learning, recognition, and environment modeling remains the same.

References

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: reconstruction and parameterization from range scans. In *ACM TOG (SIGGRAPH)*, 587–594.

BOYKO, A., AND FUNKHOUSER, T. 2011. Extracting roads from dense point clouds in large scale urban environment. *ISPRS Journal of Photogrammetry and Remote Sensing* (Oct.).

CHANG, W., AND ZWICKER, M. 2011. Global registration of dynamic range scans for articulated model reconstruction. *ACM TOG 30*, 26:1–26:15.

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approximation. *ACM TOG (SIGGRAPH) 23*, 905–914.

DEY, T. K. 2007. *Curve and Surface Reconstruction : Algorithms with Mathematical Analysis*. Cambridge University Press.

ENGELHARD, N., ENDRES, F., HESS, J., STURM, J., AND BURGARD, W. 2011. Real-time 3D visual slam with a hand-held rgb-d camera. In *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*.

FISHER, M., SAVVA, M., AND HANRAHAN, P. 2011. Characterizing structural relationships in scenes using graph kernels. *ACM TOG 30*.

GUPTA, A., EFROS, A. A., AND HEBERT, M. 2010. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*.

HENRY, P., KRAININ, M., HERBST, E., REN, X., AND FOX, D. 2010. Rgbd mapping: Using depth cameras for dense 3d modeling of indoor environments. In *In RGB-D: Advanced Reasoning with Depth Cameras Workshop in conjunction with RSS*.

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 *Proc. UIST*, 559–568.

KOLMOGOROV, V. 2006. Convergent tree-reweighted message passing for energy minimization. *IEEE PAMI 28*, 10, 1568–1583.

LEE, D. C., GUPTA, A., HEBERT, M., AND KANADE, T. 2010. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. *NIPS 24*.

LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM TOG (SIGGRAPH) 28*, 175:1–175:10.

LI, Y., WU, X., CHRYSATHOU, Y., SHARF, A., COHEN-OR, D., AND MITRA, N. J. 2011. GlobFit: consistently fitting primitives by discovering global relations. In *ACM TOG (SIGGRAPH)*, 52:1–52:12.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. J. 2005. Example-based 3D scan completion. In *Symp. on Geometry Proc.*

PAULY, M., MITRA, N. J., WALLNER, J., POTTMANN, H., AND GUIBAS, L. 2008. Discovering structural regularity in 3D geometry. *ACM TOG (SIGGRAPH) 27*, 3, #43, 1–11.

RUSINKIEWICZ, S., HALL-HOLT, O., AND LEVOY, M. 2002. Real-time 3D model acquisition. *ACM TOG (SIGGRAPH) 21* (July).

SCHNABEL, R., WAHL, R., AND KLEIN, R. 2007. Efficient RANSAC for point-cloud shape detection. *CGF (EUROGRAPHICS) 26*, 2, 214–226.

SWAIN, M. J., AND BALLARD, D. H. 1991. Color indexing. *International Journal of Computer Vision 7*, 1, 11–32.

TRIEBEL, R., SHIN, J., AND SIEGWART, R. 2010. Segmentation and unsupervised part-based discovery of repetitive objects. In *Proceedings of Robotics: Science and Systems*.

ZHENG, Q., SHARF, A., WAN, G., LI, Y., MITRA, N. J., COHEN-OR, D., AND CHEN, B. 2010. Non-local scan consolidation for 3D urban scenes. *ACM TOG (SIGGRAPH) 29*, 94:1–94:9.