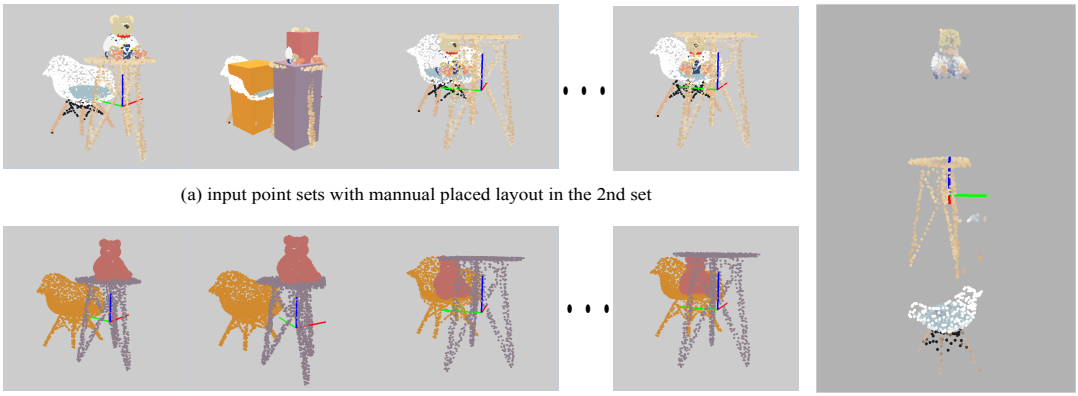
EUROGRAPHICS 2018 / D. Gutierrez and A. Sheffer
(Guest Editors)

Volume 37 (2018), Number 2

Point Set Joint Registration and Co-segmentation

Paper 1115



(a) input point sets with mannual placed layout in the 2nd set

(b) co-segmentation result

(c) joint registration result
(final Gaussin centroids)

Figure 1: *Given a series of point sets and user-placed object layout boxes in one point set (a), our method jointly segment and register objects in the point sets to generate object-level co-segmentation (b) and registration to object models represented as GMM (c).*

Abstract

We present a novel approach of joint registration and co-segmentation for point sets where objects move in different ways. Considering the joint registration and co-segmentation as two problems heavily entangled with each other, we represent the input point sets as samples from a generative model and bring up with a novel formulation based on Gaussian mixture model. By maximizing the posterior probability of the samples, we gradually recover the latent object models as well as an object-level segmentation, and simultaneously align the segmented points to the latent object models. Along with the formulation, we design an interactive tool that helps users intuitively intervene the process to optimize the registration and segmentation result. The experiment results on a group of synthetic and scanned point clouds demonstrate that our method is powerful and effective for joint registration and co-segmentation on point sets of multiple objects.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [COMPUTER GRAPHICS]: Applications—I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene Analysis—Range data

1. Introduction

Many research projects and applications of indoor scenes require segmented, and even annotated 3D databases [NXS12, DSS12, FRS*12, CLW*14, FSL*15]. One way to build such a database is to interactively compose scenes using 3D meshes for objects, resulting in scenes with object segmentation and annotation naturally available, or to manually segment and annotate existing 3D scenes. This procedure is tedious and time-consuming, despite many efforts of improving the interaction experience [MSL*11, XCF*13]. Another way is to automatically compose a scene model from an image based on existing 3D shape models [LZW*15, CLW*14]. In these methods, a retrieval procedure is usually needed and inevitably limits the result to a certain set of 3D models, without producing the actual 3D shapes that appear in the input image.

Generating scene models directly from captured point clouds will significantly facilitate dataset construction and increase the variety of the dataset. However, there is a large gap between the desired 3D model dataset and current available scene capturing tools. Typically, clean, complete and separated models for objects are desired to construct a scene database. By contrast, a noisy

and incomplete point set of different objects all in one is usually obtained with current available consumer-level scene capturing frameworks [IKH*11, NZIS13, DNZ*17]. Thus, a generic object-level segmentation and modeling method from scanned point sets is a strong demand to fill the gap.

A generic object-level segmentation is not an equivalence of the multi-label classification problem since segmentation is not limited to a fixed number of object categories predefined in the training data. Existing approaches for segmenting scanned 3D data require additional knowledges, such as a block-based stability [JGSC15], or motion consistency of rigid objects [XHS*15]. While a robot is employed to do proactive pushes, movement tracking is used to verify and iteratively improves the object-level segmentation result [XHS*15]. However, it remains significantly challenging to recover the motion consistency in a non-invasive way.

In this paper, we explore the motion consistency of rigid objects in a new aspect. While the motion consistency of objects in indoor scenes is naturally revealed by human activities over time, we hope to segment the objects in a scene from scanned point sets at different times. With respect to this idea, we are facing the choice of scanning schemes. One way is to record the change of a scene along with human activities. Another option is to schedule a periodic sweep that only records the result of human activities but avoids capturing human motion. In both schemes, it is non-trivial to recover object correspondences in different point sets due to occlusions. The occlusions are probably caused by human bodies in the first scheme or sparse sampling on times in the second scheme. In the first scheme, extra challenging processing may be required such as tracking objects with severe occlusions by human bodies. Therefore, we choose the second scanning scheme.

Thus, our original intention of building 3D scene datasets from scanned point sets leads us to the problem of coupled joint registration and co-segmentation. As shown in Figure 1, by solving the problem of coupled joint registration and co-segmentation we not only partition point sets into objects but also solve rigid motion of the objects among different point sets. In this problem, registration and segmentation are entangled in each other. On the one hand, the segmentation problem depends on the registration to connect the point clouds into series of rigid movement so that the object-level segmentation can be done based on the motion consistency. On the other hand, the registration problem relies on the segmentation to break the problem into a series of rigid joint registration of objects, otherwise the registration of multiple scenes is a non-rigid joint registration with *non-coherent point drift*. Non-coherent point drift means that a pair of points are close to each other in one point set, but their corresponding pair of points in another point set is far from each other. This happens when two points actually belong to different objects. This makes a big difference from non-rigid registration problems where point motions are smooth everywhere (such as the problem studied in [MS10]). Solving such a non-coherent non-rigid joint registration is non-trivial. Instead, breaking it up into a series rigid joint registration with object-level segmentation makes it possible to tackle the problem. In our method, we employ a group of Gaussian mixture models (GMM) and each of these Gaussian mixture models represents a potential object. This representation unentangles the registration and segmentation in the way that the segmentation can be done by evaluating the probability of belonging to the Gaussian mixture models for each point, while the registration can be done by evaluating a rigid registration in different point sets against each Gaussain mixture model. In summary, our work makes the following contributions:

1. To the best of our knowledge, we first put forward the problem of joint registration and co-segmentation of multiple point sets.
2. We propose a generative model to simultaneously solve the joint registration and co-segmentation of point sets.
3. We design an interactive tool for joint registration and co-segmentation based on the generative model.

2. Related Work

In this section, we review a series techniques on point set registration and segmentation that are related to our method.

Point set registration with GMM representation. Gaussian mixture models are widely used for point set registration problems due to its general ability of representing point sets for both rigid and non-rigid registrations and its robustness against noise. A comprehensive survey about point set registration approaches using Gaussian mixture models can be found in [JV11]. They also present a unified framework for rigid and nonrigid registration problems. These methods select one of the point sets as the "template model" and fit other point sets to this "template model". Myronenko and Song consider the registration of two point sets as a probability density estimation problem [MS10]. They use GMM to represent the geometry and force the GMM centroids to move coherently as a group to preserve the topological structure of the point sets. This method is applicable to both rigid registration and non-rigid registration. Unlike above approaches, [EKHP14] treats all point sets equally as the realizations of a GMM and the registration is cast into a clustering problem. A more recent method pushes this idea to the application on a large-scale dataset [CP16]. Comparing to these methods, our method can be seen as an extension of the formulation of [EKHP14] to simultaneously handle joint registration and co-segmentation. The difference between our method and non-rigid registration techniques is that we handle the non-coherent point drift by estimating independent transformation for each object.

Interactive image segmentation and image co-segmentation. Many interactive methods have been proposed to leverage human interaction on high-level hints and the powerful computational ability of computers. An influential technique for interactive image segmentation is GrabCut [RKB04]. It uses two GMMs for foreground and background respectively. To initialize these two Gaussian mixture models, a rectangle is manually placed to contain the foreground. Our design of user interaction draws on the experience from [RKB04]. The difference is that our tool is designed for 3D space and handles multi-object segmentation rather than a binary segmentation. Dating back to 2006, a serious of works have been done on image co-segmentation [RMBK06]. These works are based on the basic idea of exploring inter-image consistent information to extract common objects from multiple images. A more recent work of [TSS16] jointly recovers co-segmentation and dense per-pixel correspondences in two images. Though our input and

output are totally different from [TSS16], we share with [TSS16] the idea of jointly recovering co-segmentation and point-to-point correspondences (by registration).

Segmentation from motion. Object motion, as a strong hint for object segmentation, is widely used in many approaches. [XHS*15] employs a robot to do proactive pushes and tracks the motion to learn object segmentation. [LPRD16] exploits motions in a video and uses the motion edges as training data to learn an edge detector for images. These methods lean on the motion that is continuous over time and can be tracked. In comparison, our method handles motion that is non-continuous over time.

3D object recognition based on correspondence grouping. By allowing interactively input the scene layout, the joint registration and co-segmentation problem can be treated as a series of 3D object recognition problems in point sets. Our method should be classified as one of the correspondence grouping method. Comparing to previous methods [TS10, CB07], our method simultaneously solves the problem for multiple target models in multiple scenes.

3. Problem Definition

In this section, we introduce our formulation of the joint registration and co-segmentation problem for point sets. The input of our problem is a group of 3D point sets $\mathcal{V} = \{\mathbf{V}_m\}_{m=1}^{M}$ that are captured at M different times in a scene, where objects move in different ways. Each point set $\mathbf{V}_m = \{\mathbf{v}_{mi}\}_{i=1}^{L_m}$ contains L_m 3D points. Our problem is to simultaneously partition the point sets into N objects and figure out the transformations from objects to each point set. For partitioning, we output point-wise label vectors $\{\mathbf{y}_m\}$ for each input point set to indicate its object partition. For registration, we output $\{\mathbf{R}_{mn}, \mathbf{t}_{mn}\}$ to indicate the transformations from N objects to M point sets, respectively.

3.1. Basic Formulation

For robustness, we do not model a point set as a simple composition of transformed 3D points in each object model. Instead, we model each point set as a realization of an unknown central Gaussian mixture model (GMM) from of the transformed object models. In other words, we explicitly separate total K_{all} Gaussian models to N groups to represent N objects $\{O_n\}_{n=1}^{N}$ as

$$\{\underbrace{\{\mathbf{x}_1,\Sigma_1\},\cdots,\{\mathbf{x}_{K_1},\Sigma_{K_1}\}}_{O_1},\underbrace{\{\mathbf{x}_{K_1+1},\Sigma_{K_1+1}\},\cdots,\{\mathbf{x}_{K_1+K_2},\Sigma_{K_1+K_2}\}}_{O_2},\cdots\cdots,\underbrace{\{\mathbf{x}_{K_S+1},\Sigma_{K_S+1}\},\cdots,\{\mathbf{x}_{K_S+K_n},\Sigma_{K_S+K_n}\}}_{O_n},\cdots\} \tag{1}$$

where $K_S = \sum_{i=1}^{n-1} K_i$.

The Gaussian centroids $\{\mathbf{x}_k\}$ represent the point positions in objects. $\{\Sigma_k\}$ quantify the variance of point positions in objects. O_n has K_n Gaussian models and $\{K_n\}_{n=1}^{N}$ are predefined, as described in Sec. 4. The total number of Gaussian centroids is denoted as $K_{all} = \sum_{n=1}^{N} K_n$. Each object O_n is rigidly transformed to each point set $\mathbf{V}_m$ with a transformation $\phi_{mn}(\mathbf{x}_k) = \mathbf{R}_{mn}\mathbf{x}_k + \mathbf{t}_{mn}$ for $\mathbf{x}_k \in O_n$. Figure 2 shows a simple illustration for this formulation. Hence, for each point $\mathbf{v}_{mi}$ in a point set $\mathbf{V}_m$, given object models $\{O_n\}$ and their rigid transformations $\{\phi_{mn}\}$ to the point sets, we can write

$$P(\mathbf{v}_{mi}) = \sum_{k=1}^{K_{all}} p_k \mathcal{N}(\mathbf{v}_{mi}|\phi_{mn}(\mathbf{x}_k),\Sigma_k), \tag{2}$$

where the observed point $\mathbf{v}_{mi}$ is a sampling point from a large Gaussian mixture model that represents N objects together. $\{p_k\}_{k=1}^{K_{all}}$ are weights for K_{all} Gaussian models.

Given the generative representation, the unknown parameters of our joint registration and segmentation problem are

$$\Theta = \big\{\{p_k,\mathbf{x}_k,\Sigma_k\}_{k=1}^{K_{all}},\{\phi_{mn}\}_{m=1,n=1}^{M,N}\big\}. \tag{3}$$

Once we estimate these parameters, each point in all input point sets can be assigned to one of the Gaussian models according to the largest sampling probability. Since the Gaussian models are predefined to be one of the N objects, we can further deduce the $\{\mathbf{y}_m\}_{m=1}^{M}$ indicating vectors of object-level co-segmentation for each input point set based on such assignment. To estimate the parameters Θ to fit all the input point sets without knowing object labels for all 3D points, the problem can be solved in an Expectation-Maximization (EM) framework. In particular, we bring in hidden variables as:

$$\mathcal{Z} = \{z_{mi}|m = 1...M, i = 1...L_m\}, \tag{4}$$

such that $z_{mi} = k$, $k \in \{1,2...,K_{all}\}$ assigns the observed point $\mathbf{v}_{mi}$ to the k^{th} Gaussian model $\mathbf{x}_k,\Sigma_k$. We aim to maximize the expected complete-data log-likelihood:

$$\mathcal{E}(\Theta|\mathcal{V},\mathcal{Z}) = E_{\mathcal{Z}}[\ln P(\mathcal{V},\mathcal{Z};\Theta)|\mathcal{V}] = \sum_{\mathcal{Z}} P(\mathcal{Z}|\mathcal{V},\Theta)\ln P(\mathcal{V},\mathcal{Z};\Theta). \tag{5}$$

This formulation can be seen as an adaption of the joint registration formulation in [EKHP14], upon which we separate Gaussian models into groups to express multiple objects. Under the assumption that the input points are independent and identically distributed, we can rewrite the objective defined in Eq. (5) into:

$$\Theta = \arg\max \sum_{m,i,k} \alpha_{mik}\big(\ln p_k + \ln P(\mathbf{v}_{mi}|z_{mi} = k;\Theta)\big), \tag{6}$$

where $\alpha_{mik} = P(z_{mi} = k|\mathbf{v}_{mi};\Theta)$. By bringing in Eq. (2) and ignoring constant terms, we can rewrite the objective as:

$$\Theta = \arg\max\Big(-\frac{1}{2}\sum_{m,i,k}\alpha_{mik}\big(||\mathbf{v}_{mi}-\phi_{mn}(\mathbf{x}_k)||^2_{\Sigma_k} + \ln|\Sigma_k| - 2\ln p_k\big)\Big), \tag{7}$$

where the $|\cdot|$ denotes the determinant and $||\mathbf{x}||^2_{\mathbf{A}} = \mathbf{x}^T\mathbf{A}^{-1}\mathbf{x}$. It is predefined that $\mathbf{x}_k$ is one of the Gaussian centroids used to represent the n^{th} object, which is why we apply the transformation ϕ_{mn} on $\mathbf{x}_k$. For the convenience of computation, we restrict the model to isotropic covariances, i.e., $\Sigma_k = \sigma^2\mathbf{I}$ and $\mathbf{I}$ is an identity matrix. Now, we can optimize the objective through iterating between estimating α_{mik} (Expectation-step) and maximizing $\mathcal{E}(\Theta|\mathcal{V},\mathcal{Z})$ with respect to each parameter in Θ (Maximization-step).

E-step: this step estimates the posterior probability α_{mik} of $\mathbf{v}_{mi}$ to be a point generated by the k^{th} Gaussian model.

$$\alpha_{mik} = \frac{p_k\sigma_k^{-3}exp(-\frac{1}{2\sigma_k^2}||\mathbf{v}_{mi}-\phi_{mn}(\mathbf{x}_k)||^2)}{\sum_s^{K_{all}} p_s\sigma_s^{-3}exp(-\frac{1}{2\sigma_s^2}||\mathbf{v}_{mi}-\phi_{mn}(\mathbf{x}_s)||^2)} \tag{8}$$

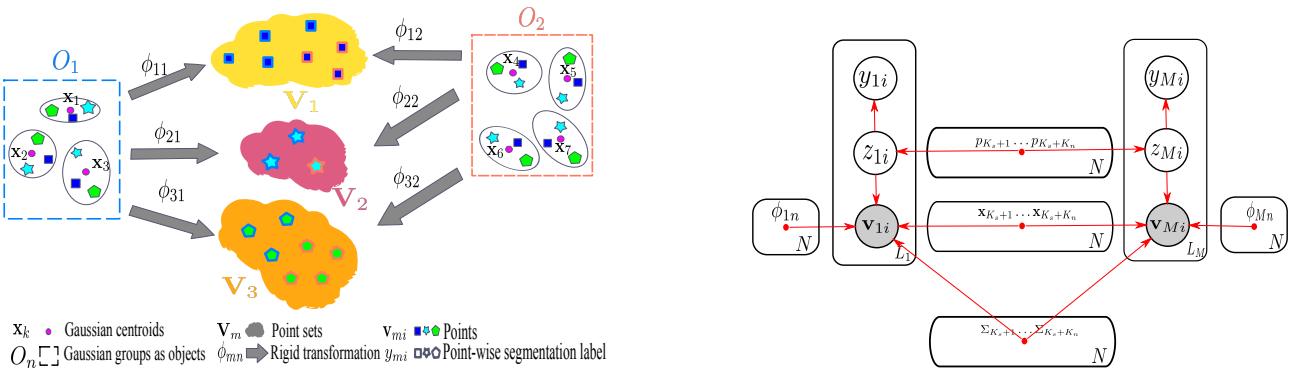


Figure 2: *Our generative model for joint registration and co-segmentation (a) and its associated graphical model (b). (a) illustrates 7 Gaussian models $\{\mathbf{x}_i, \Sigma_i\}_{i=1}^7$ are grouped into two object models O_1 and O_2. Each object is transformed to a point set $\mathbf{V}_i$ by ϕ_{mi}. A 3D point in a point set $\mathbf{V}_m$ is a sampling point from a Gaussian mixture model composed of the 7 transformed Gaussian models.*

M-step-a: this step updates the transformations ϕ_{mn} that maximize $\mathcal{E}(\Theta)$, given instant values for α_{mik}, $\mathbf{x}_k$, σ_k. We only consider rigid transformations, making $\phi_{mn}(\mathbf{x}) = \mathbf{R}_{mn}\mathbf{x} + \mathbf{t}_{mn}$. The maximizer $\mathbf{R}^*_{mn}, \mathbf{t}^*_{mn}$ of $\mathcal{E}(\Theta)$ is the same with the minimizers of the following constrained optimization problems

$$\begin{cases} \min_{\mathbf{R}_{mn},\mathbf{t}_{mn}} & ||(\mathbf{W}_{mn} - \mathbf{R}_{mn}\mathbf{X}_n - \mathbf{t}_{mn}\mathbf{e}^T)\Lambda_{mn}||_F^2 \\ s.t. & \mathbf{R}_{mn}^T\mathbf{R}_{mn} = I, |\mathbf{R}_{mn}| = 1 \end{cases} \tag{9}$$

where Λ_{mn} is a $K_n \times K_n$ diagonal matrix with elements $\lambda_{mnk} = \frac{1}{\sigma_k}\sqrt{\sum_i^{L_m} \alpha_{mik}}$, L_m is the number of point for the m^{th} input point set, $\mathbf{X}_n = [\mathbf{x}_{K_S+1}, \mathbf{x}_{K_S+2}, ..., \mathbf{x}_{K_S+K_n}]$ is the matrix stacked by the centroids of Gaussian models that are predefined to represent the n^{th} object. $\mathbf{e}^T$ is a vector of ones, $||\cdot||_F$ denotes the Frobenius norm, and $\mathbf{W}_{mn} = [\mathbf{w}_{m(K_S+1)}, \mathbf{w}_{m(K_S+2)}, ..., \mathbf{w}_{mk}, ..., \mathbf{w}_{m(K_S+K_n)}]$ where $\mathbf{w}_{mk}$ is a weighted average point as

$$\mathbf{w}_{mk} = \frac{\sum_{i=1}^{L_m} \alpha_{mik}\mathbf{v}_{mi}}{\sum_{i=1}^{L_m} \alpha_{mik}} \tag{10}$$

This problem has a similar solution with [EKHP14]. The only difference is that we are estimating the transformation from Gaussian models to the input point sets instead of the transformation from input point sets to Gaussian models, since there are multiple groups of $\mathbf{x}_k$ corresponding to multiple objects in our Gaussian models. The optimal can be given by:

$$\mathbf{R}^*_{mn} = \mathbf{U}_{mn}\mathbf{C}_{mn}\mathbf{V}_{mn}^T \tag{11}$$

$$\mathbf{t}^*_{mn} = \frac{1}{tr(\Lambda^2_{mn})}(\mathbf{W}_{mn} - \mathbf{R}_{mn}X_n)\Lambda^2_{mn}\mathbf{e} \tag{12}$$

where $[\mathbf{U}_{mn}, \mathbf{S}, \mathbf{V}_{mn}] = svd(\mathbf{W}_{mn}\Lambda_{mn}\mathbf{P}_{mn}\Lambda_{mn}\mathbf{X}_n^T)$ and $\mathbf{P}_{mn} = \mathbf{I} - \frac{\Lambda_{mn}\mathbf{e}(\Lambda_{mn}\mathbf{e})^T}{(\Lambda_{mn}\mathbf{e})^T\Lambda_{mn}\mathbf{e}}$, $\mathbf{I}$ is identity matrix. $C_{mn} = diag(1, 1, |\mathbf{U}_{mn}||\mathbf{V}_{mn}|)$.

M-step-b: in this step we update the parameters related to the Gaussian mixture model and the indicating vector for object segmentation

$$\mathbf{x}^*_k = \frac{\sum_{m=1}^M \sum_{i=1}^{L_m} \alpha_{mik}(\mathbf{R}_{mn}^{-1}\mathbf{v}_{mi} - \mathbf{t}_{mn})}{\sum_{m=1}^M \sum_{i=1}^{L_m} \alpha_{mik}} \tag{13}$$

where $\mathbf{x}_k$ is one of the Gaussian centroids that is predefined to represent the n^{th} object.

$$\sigma^{*2}_k = \frac{\sum_{m=1}^M \sum_{i=1}^{L_m} \alpha_{mik}||(\mathbf{v}_{mi} - \mathbf{t}_{mn} - \mathbf{R}^*_{mn}\mathbf{x}^*_k)||_2^2}{3\sum_{m=1}^M \sum_{i=1}^{L_m} \alpha_{mik}} \tag{14}$$

$$p^*_k = \frac{\sum_{m,i} \alpha_{mik}}{M} \tag{15}$$

$$y^*_{mi} = \arg\max_n \sum_{k=\sum_{s=1}^{n-1} K_S+1}^{\sum_{s=1}^{n} K_S} \alpha_{mik} \tag{16}$$

where y_{mi} is the i^{th} entry of the indicate vector $\mathbf{y}_m$ and it assigns the i^{th} point of the m^{th} point set to one of N objects.

3.2. Bilateral Formulation

In the basic formulation, only position information is used in Gaussian models. When considering point-wise features, such as color, texture, we can add bilateral terms into the generative model.

$$P(\mathbf{v}_{mi}, \mathbf{f}_{mi}) = \sum_{k=1}^{K_{all}} p_k \mathcal{N}(\mathbf{v}_{mi}|\phi_{mn}(\mathbf{x}^v_k), \sigma v_k)\mathcal{N}(\mathbf{f}_{mi}|\mathbf{x}^f_k, \sigma^f_k), \tag{17}$$

where $\mathbf{f}_{mi}$ is the feature vector for point $\mathbf{v}_{mi}$ and $\mathbf{x}^f_k$ is the feature vector for k^{th} point in object model. As shown in the formulation, there is no transformation applied onto $\mathbf{x}^f_k$, which means that this formulation is only suitable to the feature that is rotation and translation invariant. For example, we use the point color as a 3D feature vector in this paper. In this formulation $\mathcal{N}(v_{mi}|\phi_{mn}(xv_k), \sigma v_k)$ is the spatial term and $\mathcal{N}(\mathbf{f}_{mi}|\mathbf{x}^f_k, \sigma^f_k)$ is the feature term. For the bilateral formulation, iteration steps will be as follows:

E-step:in this step the calculation of posterior probability need to consider both the spatial term and the feature term.

$$\alpha_{mik} = \frac{p_k P_v(\mathbf{v}_{mi}, \phi_{mn}(\mathbf{x}^v_k), \sigma v_k)P_f(\mathbf{f}_{mi}, \mathbf{x}^f_k, \sigma^f_k)}{\sum_s^{K_{all}} p_s P_v(\mathbf{v}_{mi}, \phi_{mn}(\mathbf{x}^v_s), \sigma^v_s)P_f(\mathbf{f}_{mi}, \mathbf{x}^f_k, \sigma^f_s)} \tag{18}$$

where $P_v(\mathbf{x},\mathbf{y},\sigma) = \sigma^{-3} exp(-\frac{1}{2\sigma^2}||\mathbf{x}-\mathbf{y}||^2)$ and $P_f(\mathbf{x},\mathbf{y},\sigma) = \sigma^{-D(\mathbf{x})} exp(-\frac{1}{2\sigma^2}||\mathbf{x}-\mathbf{y}||^2)$ and $D(\mathbf{x})$ means the dimension of the vector $\mathbf{x}$.

M-step-a: for bilateral formulation, this step is the same with the basic formulation and the update can be done as Eq. (11) and Eq. (12).

M-step-b: for bilateral formulation, this step needs not only update model centroids and variance for the spatial term as Eq. (13) and Eq. (14). but also update the centroids and variance for the feature term as in Eq. (19) and Eq. (20).

$$\mathbf{x}_k^{f*} = \frac{\sum_{m=1}^{M}\sum_{i=1}^{L_m}\alpha_{mik}\mathbf{f}_{mi}}{\sum_{m=1}^{M}\sum_{i=1}^{L_m}\alpha_{mik}} \tag{19}$$

$$\sigma_k^{f*2} = \frac{\sum_{m=1}^{M}\sum_{i=1}^{L_m}\alpha_{mik}||\mathbf{f}_{mi}-\mathbf{x}_k^{f*}||_2^2}{D(\mathbf{f})\sum_{m=1}^{M}\sum_{i=1}^{L_m}\alpha_{mik}}, \tag{20}$$

where $D(\mathbf{f})$ is the feature dimension. The update of p_k for bilateral formulation is the same as the basic formulation in Eq. (15).

4. Initialization and Optimization

Based on our formulation described in Sec. 3, our method for joint registration and co-segmentation can be summarized in Algorithm 1. In this section, we will explain in detail the parameter initialization and the user-guided optimization of our algorithm.

Algorithm 1 Joint Registration and Co-segmentation (JRCS)

Input:
$\{\mathbf{V}_m\}$: M 3D point sets
Θ^0: Initial parameters
$\{\beta_{ik}\}_m$: Layout prior
Output:
Θ^q: Final parameters
1. $q \leftarrow 1$
2. **repeat**
3. E-step: Use Θ^{q-1} to estimate α_{mik}^q according to Eq. (8) (use Eq. (18) for the bilateral formulation);
4. **if** $q < q_{alt}$ **then** Alter α_{mik}^q with $\{\beta_{ik}\}_m$ according to Eq. (23);
5. M-step-a: use α_{mik}^q, $\mathbf{x}_k^{q-1}$ to estimate $\{\mathbf{R}_{mn}^q\}$ and $\{\mathbf{t}_{mn}^q\}$ according to Eqs. (11)(12);
6. M-step-b: use α_{mik}^q, $\{\mathbf{R}_{mn}^q\}$ and $\{\mathbf{t}_{mn}^q\}$ to update other parameters for Gaussian models according to Eqs. (13)(14)(15)(16) (or Eqs. (19)(20) for the bilateral formulation);
7. $q \leftarrow q+1$
8. **until** $q > q_{max}$
9. **return** Θ^q

4.1. Initialization

In our formulation, there are a large number of parameters that can not be easily initialized. We provide an interactive tool to help with the initialization, as shown in Figure 3. A set of boxes can be manually placed to indicate a rough segmentation of different objects

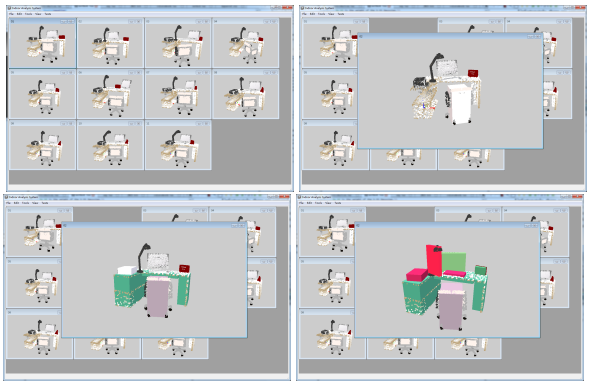


Figure 3: *From the 1^{st} to the 4^{th}, the four images show the procedure of interaction: Given all the point sets as input, the user picks one point set and place boxes to indicate the layout for this point set. The box in white is the box currently under editing. The boxes in other colors are boxes placed to represent object layouts. One color represents one object. The interaction allows multiple boxes to represent same object. (e.g. the desk is represented by three boxes in same color)*

in one point set. Each object can be roughly indicated by multiple boxes. Based on the roughly placed boxes, we can initialize the parameters in our formulation.

Number of objects N: N is naturally determined as the number of placed box groups in the point set.

Number of Gaussian models in each object $\{K_n\}_{n=1}^N$: While objects in an indoor scene have varying volumes, we use different number of Gaussian models for objects according to their volumes. We set K_n as

$$K_n = \frac{V_n}{\sum V_n} K_{all}, \tag{21}$$

where V_n represents the total volume of the boxes for O_n. The total number of Gaussian models K_{all} in the scene is initialized as $\frac{1}{2}\left(median(\{L_m\}_{m=1}^M\right)$, where L_m is the number of points in $\mathbf{V}_m$. This is an empirical choice borrowed from [EKHP14].

Gaussian parameters $\{p_k,\mathbf{x}_k,\Sigma_k\}_{k=1}^{K_{all}}$: We initially set $p_k = \frac{1}{K_{all}}$, which means each Gaussian model is equally weighted at the beginning. For object O_n, we initialize its K_n Gaussian centroids $\{\mathbf{x}_k\}_{K_S+1}^{K_S+K_n}$ as random positions uniformly distributed on the surface of a sphere, whose radius r is chosen as the median of the radius of the input point sets. The radius of a point set is defined as half of the length of diagonal line of its axis-aligned bounding box.

The center of the n^{th} sphere is $\mathbf{c}_n = (0,0,z_n)$, where $z_n \in \{-(N-1)r,-(N-3)r,...,(N-1)r\}$. This means that the object models are vertically arranged in latent space as shown in Figure 1(c). We choose vertical arrangement for groups of object merely for the convenience of visualization. Figure 5(b)E00 shows an example of the initial Gaussian centroids of a scene with three objects. The variance $\{\Sigma_k\}$ are all initialized as $\Sigma_k = \sigma^2\mathbf{I}$ in which $\sigma = r$. Without any prior knowledge, such initialization for Gaussian parameters put all the objects at similar starting points and they can

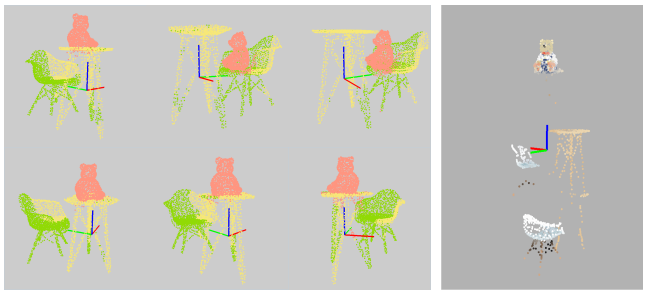


Figure 4: *An example result when our algorithm converges to a local optimal. (a) The segmentation result in six point sets of this local optimal. (b) The Gaussian centroids of the latent object models. It shows that the chair and the table are not perfectly segmented.*

compete fairly to group points in the input point sets. If we set r differently for each object based on the size of input boxes, it could be easily stuck to a local minimum that all the points are assigned to the largest object.

Transformations $\{\phi_{mn}\}_{m=1,n=1}^{M,N} = \{\mathbf{R}_{mn}, \mathbf{t}_{mn}\}_{m=1,n=1}^{M,N}$: Since we have chosen spheres as the initial shapes, we can initialize all the $\mathbf{R}_{mn}$ to an identity matrix. For translations, we initialize them as $\mathbf{t}_{mn} = -\mathbf{c}_n$ so that all the object models start with position at the origin point when they are transformed to the space of each input set. However, if boxes are manually placed in the point set $\mathbf{V}_m$, we treat the associated $\mathbf{t}_{mn}$ differently:

$$\mathbf{t}_{mn} = \frac{\sum_{\mathbf{v}_{mi} \in B_n} \mathbf{v}_{mi}}{N(B_n)} - \mathbf{c}_n, \tag{22}$$

where $N(B_n)$ here is the number of points enclosed by the manually placed boxes indicating object O_n.

4.2. Layout Constrained Optimization

Our formulation inherits the disadvantage of easily getting stuck to a local optimal from the EM framework. Without further constraint, the EM framework usually fails to get a globally optimal solution, as Figure 4 shows. The chair and the table are not perfectly segmented, To cope with this problem, we adopt the user placed boxes as soft constraints to guide the optimization and confine the shape of generated object models. Such constraints are enforced by altering the posterior probability as

$$\alpha^*_{mik} = \frac{\alpha_{mik}\beta_{mik}}{\sum_{i,k} \alpha_{mik}\beta_{mik}} \tag{23}$$

where β_{mik} is the prior probability according to the boxes, defined as:

$$\beta_{mik} = \begin{cases} 1, & \mathbf{v}_{mi} \in B_n \\ \exp(-\frac{\min_{\mathbf{v}_{mj}} ||\mathbf{v}_{mi} - \mathbf{v}_{mj}||_2^2)}{2r^2}), & \mathbf{v}_{mi} \notin B_n \text{ and } \mathbf{v}_{mj} \in B_n \end{cases} \tag{24}$$

where B_n is a set of points that are enclosed by the boxes used to represent object O_n. $\min_{\mathbf{v}_{mj}} ||\mathbf{v}_{mi} - \mathbf{v}_{mj}||_2^2$ is the minimum distance from a point $\mathbf{v}_{mi}$ to the points $\{\mathbf{v}_{mj}\}$ in object O_n. r is the median of the radius of input point sets. This alteration on posterior probability is only done for the points in the point set where boxes are manually placed.

This alteration can prevent the object models from deforming into an arbitrary shape. Figure 5 demonstrates the converging procedure with box constraints. We can see that with the boxes placed in one point set as constraints, our framework converges to a good segmenation result. The point sets are finally segmented to three objects, and the object models develop from a initial sphere shape at $q = 1$ to a dense point cloud which fits the input point sets well. However, in Figure 5(a), the objective function is not monotonically increasing. This is due to our alteration on the posterior probability in Eq. (23). This alteration is a quite brutal solution to enforce the shape constraint and it will interfere with the convergence of EM algorithms. This makes it difficult to set a stop criteria based on the objective value. We now stop the iteration when the maximum iteration number q_{max} is reached.

As highlighted in Figure 5(b)"A01"-"A08", the segmentation in the first point set seldom changes until the last few iterations. This is due to the alteration in Eq. (23) as well. In order to constrain the object shape, we do alteration on the posterior probability of the point set where boxes are placed. This alteration is only done in q_{alt} iterations as shown in Algorithm 1 step 4. However, the initial segmentation based on the boxes is not accurate. Therefore, we no longer do such alteration in the last few iterations and let the algorithm to refine the segmentation based on the result of registration. We set $q_{alt} = q_{max} - 10$ for all experiments in this paper.

For initialization and object shape constraint, the boxes are first roughly placed in one point set only. In more challenging cases, if the user is not satisfied with the segmentation and registration results, we also allows the user to add more box-shape constraints in different point sets to refine the results. The same alteration as Eq. (23) is performed in optimization. We will discuss an example of such case later in Sec 5.3.

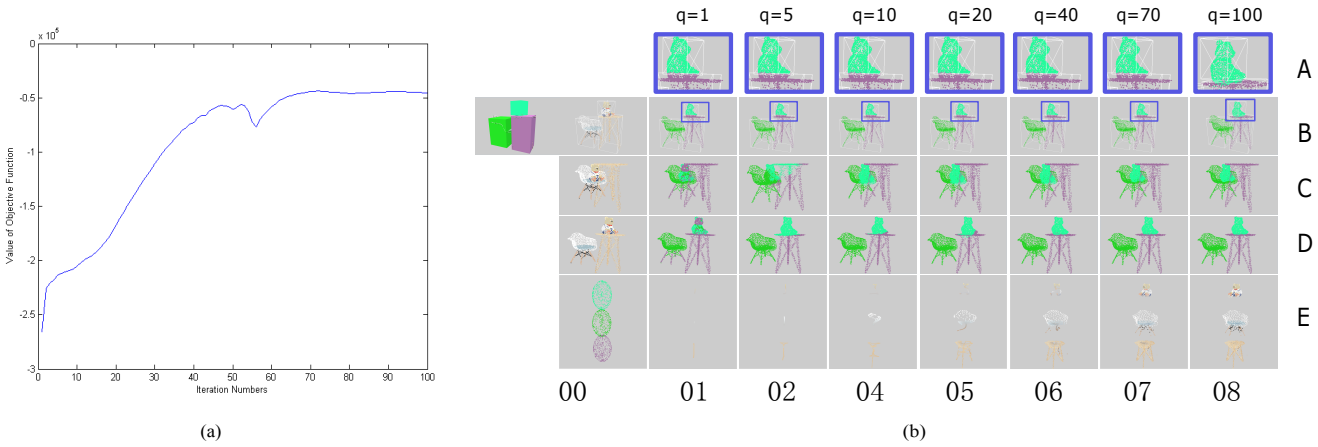
5. Experiment and Discussion

In this section, we will show a series of experimental results including evaluation for co-segmentation and joint registration on synthetic data for quantitative analysis, investigation on the robustness of our method on point completeness and amount of user interaction, and testing on one group of real data.

Synthetic Data Collection. We generate a group of synthetic datasets (synthetic point sets) for quantitatively evaluate our algorithm. For each dataset, we model a 3D scene using object models from 3D Warehouse. We convert the mesh model of the scene into a point set using the Poisson sampling method [CCS12]. Then we manually move the objects according to their functions and generate multiple point sets.

5.1. Evaluation for Co-segmentation on Synthetic Data

From the perspective of co-segmentation, we quantitatively evaluate our algorithm on two group of synthetic data of indoor scenes. To estimate the power of the proposed algorithm, the interaction of placing boxes is only performed at one point set. No fur-



(a) (b)

Figure 5: *This figure shows an example of our algorithm converging with increasing number of iterations. (a) shows the curve of objective function - iteration number. The objective value is calculated according to (7). Note that the curve is not monotonic increasing, which makes it difficult to set a stop condition based on our objective.(b) shows three input point set as examples out of 12. Column "00" shows the input point sets and the initial Gaussian centroids, among which "B00" has two images the left one is the input layout (boxes) which is only placed in the first point set. The column "01"-"08" shows result of segmentation (in row "B"-"D") and Gaussian centroids (in row "E") at different iteration numbers q. The q is shown at top of each column. The row "A" shows highlighted areas of "B01"-"B08".*

Table 1: *Mean and standard deviation of IOU scores on two synthetic datasets. JRCS-Basic is our basic formulation. JRCS-Bilateral is our bilateral formulation with point color as feature. PointNet is the semantic segmentation of [QSMG17].*

Datasets	Study Room	Office Desk
JRCS-Basic	0.808±0.032	0.831 ± 0.027
JRCS-Bilateral	0.876±0.012	0.829 ± 0.028
PointNet	0.402±0.032	0.439 ± 0.049

ther interaction is required. For numerical estimation, we calculate the intersection over union (IOU) scores for the inducing segmentation against the ground-truth segmentation. We compare our results with the state of art semantic segmentation method, PointNet [QSMG17], which trains a network using a large-scale database. Table 1 shows the numeric result and Figure 6 shows visual result of three input point set including the one that is equiped with input layout. For the object class that is not annotated in the training data, PointNet [QSMG17] treats it as a special class of "clutter". This is why we have different ground truth for our method and PointNet. As shown in Figure 6, we have "GT" as groundtruth used to evaluate our method and "GT for PointNet" as groundtruth used to evaluate PointNet. Comparing our method to PointNet is not an exact fair comparison in following aspects:

1. Our method allows user interaction and PointNet is fully automatic in the test phase.
2. Our synthetic data is quite different from the data in Stanford 3D semantic parsing dataset [ASZ*16] which is used to train the semantic segmentation network of PointNet.
3. Our method outputs object-level segmentation without semantic label, while PointNet outputs semantic labels.

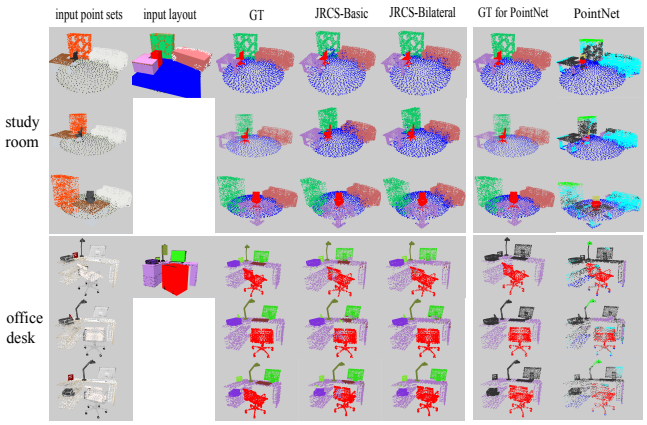


Figure 6: *Segmentation evaluations on two groups of synthetic data (study room and office desk). Three examples of point set have been shown from each group.*

However, by comparison we can see that the generalization ability of current learning based point set segmentation method is still far from enough to be used as tool to prepare data and build dataset. Semantic segmentation method is limited to certain set of object classes (13 classes for PointNet) and cannot be used to carry on our task.

5.2. Evaluation for Joint Registration on Synthetic Data

From the perspective of joint registration, we first evaluate the result by transferring the point cloud of objects to each input point set

Table 2: *Registration errors of the three groups of synthetic data in Figure 6. The errors are measured in meter.*

Method@Dataset	Maximum	Median	Minimum
Basic@Study Room	0.441	0.085	0.027
Bilateral@Study Room	0.139	0.052	1.31e-05
Basic@Office Desk	0.309	0.0408	5.82e-03
Bilateral@Office Desk	0.222	0.0574	8.33e-03

based on result $\{\phi_{mn}\}$ and calculating the average distance from a point to its true correspondent point for each input point set. We use this average distance as fitness error to evaluate the registration quality respect to each input set.

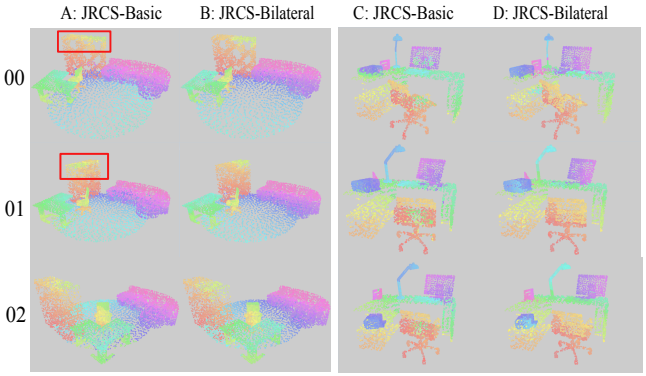


Figure 7: *This figure shows some visual examples of joint registration. The result is illustrated by color-coding the point-wise correspondences.*

Table 2 shows the result of this evaluation. In Table 2 the Maximum, Median and Minimum of the fitness error across input sets are reported. We find that even the input set with high IOU scores in segmentation can result in high fitness error. We believe this is due to the symmetric and near-symmetric objects in the scene. For symmetric objects, even the registration is correct the distance from a point to its true correspondent point can be high, since the rotation in registration result can be different from the one we use to generate this synthetic data. For near-symmetric objects, the registration often gets stuck in a local optimal and results in a high IOU score but a high fitness error.

Therefore, we show some example of visual result for joint registration in Figure 7. By comparing Figure 7"A01" and Figure 7"A02", we can see that the registration of the round carpet is correct but due to its symmetry its point-wise correspondences are not the same with identity transformation. By comparing Figure 7"A00" and Figure 7"A01", (the parts that are highlighted by red rectangle) we can see that registration of the shelf is not correct and it is stucked at a local minimum that maps left part to right part.

We then compare our method (JRCS-Basic) with [EKHP14](JRMPC) on the synthetic point sets released by [EKHP14]. These data contains four point sets of Stanford Bunny with different noise and outliers. From the experiment

Table 3: *This table shows the RMSE of joint registration on 4 point sets of Stanford Bunny. JRMPC is the method of [EKHP14]. JRCS-Basic is our basic formulation*

Point Sets	View 2	View 3	View 4
JRMPC	0.1604	0.1719	0.1838
JRCS-Basic	0.0822	0.1570	0.2301

result shown in Table 3 and Figure 8, we can see that when dealing with one object, our method have similar result with [EKHP14].

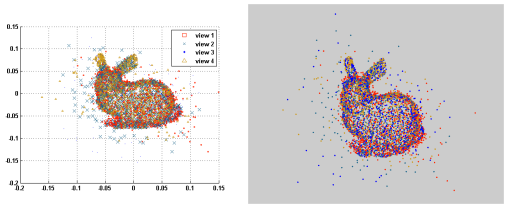


Figure 8: *This figure shows the visual result of joint registration on 4 point sets of Stanford Bunny by JRMPC ([EKHP14] at left) and JRCS-Basic (our basic formulation at right)*

5.3. Investigation on Interaction

For parameter initialization and object shape constraint, we only need the user to input layout (boxes) in one of the input point sets. However, our algorithm sometimes gets stuck at local minimum on handling non-local motion of objects. In such challenging cases, we require more user input to further guide the optimization. In this subsection, we show an example of such challenging case and investigate on the amount of interaction that is needed to improve the result. Figure 9 shows how the IOU score increases along with the amount of interaction. In this experiment, we use JRCS-Basic. From this experiment, We can see this from Figure 9, the curve of Minimum IOU is not monotonically increasing with the amount of manual input, which means more interaction does not guarantee improvement of the segmentation results in all point sets. When the initial correspondences in most point sets are far from correct our method loses its ability to transfer the information among different point sets. The further interaction only improves the segmentation in the point set which the user adds layout into and barely improves the segmentation in other point sets. From Figure 10, we can see that actually quite a lot more interaction is needed for the overall segmentation result to be visually satisfying.

5.4. Investigation on Influence of Point Incompleteness

In previous evaluation on synthetic data, we use data that the objects are completely covered by the sampled points. In this subsection, we investigate how the point set incompleteness affect the result of our algorithm. To test this, we pick a group of point sets, and gradually remove certain percentage ($0\% - 30\%$) of points from each point set. In order to simulate the point incompleteness caused by occlusion using a simple method, we generate the incomplete point sets with incompleteness of $p\%$ as follows:

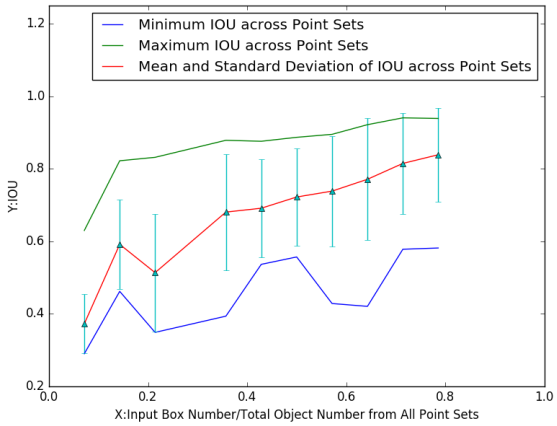


Figure 9: *This figure shows an example of how the amount of interaction affect the IOU score of co-segmentation, the X axis is ratio:* $x = \frac{Input\ Box\ Number}{Total\ Object\ Number}$. $x = 1.0$ *means that the user input one box for each object in all point sets*

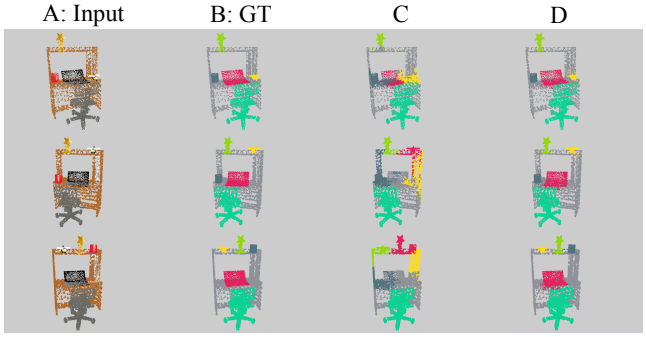


Figure 10: *This figure shows 3 out of 16 point sets as examples for experiment on amount of interaction. The column A shows the input point sets, the column B shows the groundtruth segmentation. The column C shows the visual result when 1 point set is equipped with manual input layout. The column D shows the visual result when 11 out of 16 point sets are equipped with manual input layout.*

1. We randomly pick one point from each complete point set.
2. For one point set, sort all points ascending according to their euclidean distance to the picked point.
3. Remove the first $p\%$ points from the point set to generate a point set with incompleteness of $p\%$.

Figure 11 shows how the IOU score is affected with the increasing point set incompleteness in this experiment. Figure 12 shows some example of visual result in this experiment. The results of $p = \{0.0, 14.0, 30.0\}$ are shown. In Figure 12(A09-E09), we can see that for some object in the scene half of the points are already removed. From the result we can see that even with serious incompleteness on some of the objects our algorithm converge to a relative good result.

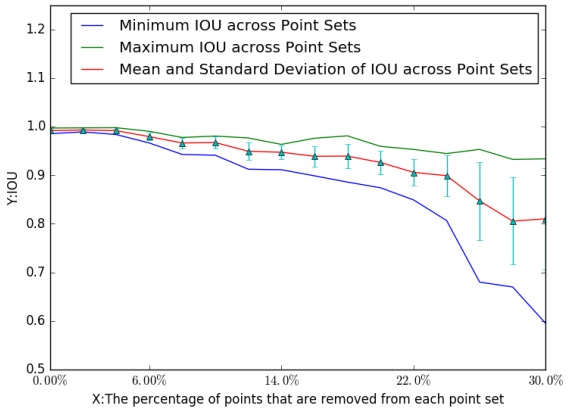


Figure 11: *This figure shows how the data incompleteness affect the IOU score of co-segmentation. The data used in this experiment is partially shown in Figure 12.*

5.5. Test On Real Data

To capture real data we employ the voxel hashing method [NZIS13] and use plane fitting to remove walls and floors. We then transfer the meshes into point sets using a Poisson sampling process [CCS12]. Figure 13 shows a scanned point set. We can see that, there are noised and blurred color, shape distortion, partial scanning and outliers in real data. Figure 14 shows the segmentation and registration results on a group of scanned point sets We uses JRCS-Bilateral in this test and Figure 14(d) shows the only point set that is equipped with layout in this test. From Figure 14(e), we can see that all input point sets are partitioned into objects. In Figure 14(g), we align the point sets all together respecting to each of the objects. There are four objects in the scene, so there are four different aligned result in Figure 14(g). The light blue rectangle highlights the object that is used to align the point sets. We can verify that the objects from each input set are aligned together by the result transformation.

5.6. Conclusions

For the challenging problem of point set joint registration and co-segmentation, we come up with a formulation simultaneously modeling the two entangled sub-problems. For the difficult initialization and optimization of this formulation, we provide a strategy that lean on a few manual inputs. In the evaluation, our algorithm have some success on both synthetic and real data. The practical issue holding us back is the time performance of our current implementation, which prevents us from going over more initialization and optimization strategies. For a group of 11 point sets with about 9K points in each point set, our current implementation will take

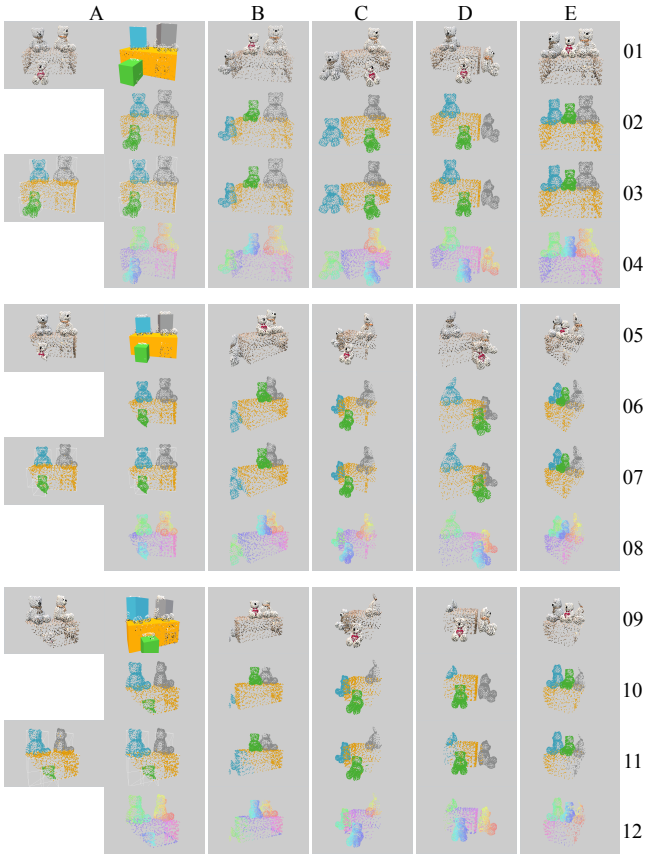


Figure 12: *This figure shows some of the visual result of experiments on incompleteness. This figure shows results at 3 different level of incompleteness which are* 0.0% *at row 01-04,* 14% *at row 05-08 and* 30% *at row 09-12. Each column shows the information of the same point set. Row 01,05,09 shows the inputs. For the A column the input is not only the point set but also a layout for initialization. Row 02,06,10 are ground-truth of segmentation. Row 03,04,10 are results of segmentation. For the A column, the intial segmentation and final segmentation are both shown. For the rest column, the final segmentation are shown. Row 04,08,12 shows the point-wise correspondences of joint registration by color-coding*

about 110 minutes to run 100 iteration. With a parallelized implementation, we can probably explore more potentials of our algorithm and try it on a scene of a larger scale by drawing experience from [CP16] and [EKT*16].

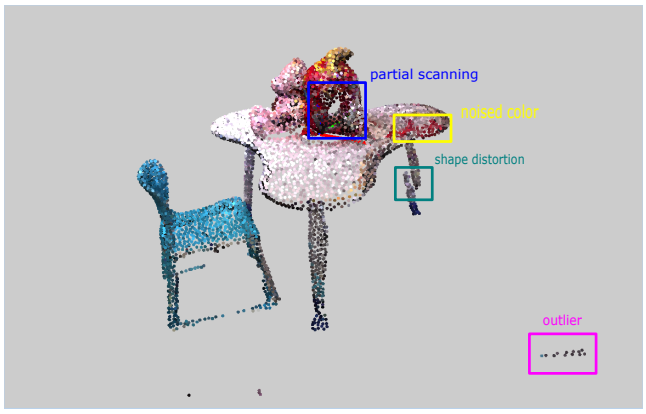


Figure 13: *This figure highlights the common challenges on real data.*

References

[ASZ*16] ARMENI I., SENER O., ZAMIR A. R., JIANG H., BRILAKIS I., FISCHER M., SAVARESE S.: 3D semantic parsing of large-scale indoor spaces. IEEE Computer Society, pp. 1534–1543. doi:doi.ieeecomputersociety.org/10.1109/CVPR.2016.170. 7

[CB07] CHEN H., BHANU B.: 3D free-form object recognition in range images using local surface patches. *Pattern Recogn. Lett. 28*, 10 (July 2007), 1252–1262. doi:10.1016/j.patrec.2007.02.009. 3

[CCS12] CORSINI M., CIGNONI P., SCOPIGNO R.: Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Trans. on Visualization and Computer Graphics 18*, 6 (June 2012), 914–924. URL: http://dx.doi.org/10.1109/TVCG.2012.34, doi:10.1109/TVCG.2012.34. 6, 9, 11

[CLW*14] CHEN K., LAI Y.-K., WU Y.-X., MARTIN R., HU S.-M.: Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 208:1–208:12. URL: http://doi.acm.org/10.1145/2661229.2661239, doi:10.1145/2661229.2661239. 1

[CP16] CAMPBELL D., PETERSSON L.: Gogma: Globally-optimal gaussian mixture alignment. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 5685–5694. doi:10.1109/CVPR.2016.613. 2, 10

[DNZ*17] DAI A., NIESSNER M., ZOLLHÖFER M., IZADI S., THEOBALT C.: BundleFusion: Real-time globally consistent 3D reconstruction using online surface re-integration. *ACM Trans. Graph. 36*, 4 (2017). URL: http://doi.acm.org/10.1145/3072959.3126814, doi:10.1145/3072959.3126814. 2

[DSS12] DEMA M. A., SARI-SARRAF H.: 3D scene generation by learning from examples. In *IEEE International Symposium on Multimedia* (Dec 2012), pp. 58–64. doi:10.1109/ISM.2012.19. 1

[EKHP14] EVANGELIDIS G. D., KOUNADES-BASTIAN D., HORAUD R., PSARAKIS E. Z.: A generative model for the joint registration of multiple point sets. In *ECCV* (2014), pp. 109–122. URL: https://doi.org/10.1007/978-3-319-10584-0_8, doi:10.1007/978-3-319-10584-0_8. 2, 3, 4, 5, 8

[EKT*16] ECKART B., KIM K., TROCCOLI A., KELLY A., KAUTZ J.: Accelerated generative models for 3D point cloud data. In *CVPR* (June 2016), pp. 5497–5505. doi:10.1109/CVPR.2016.593. 10

[FRS*12] FISHER M., RITCHIE D., SAVVA M., FUNKHOUSER T., HANRAHAN P.: Example-based synthesis of 3d object arrangements. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 135:1–135:11. URL: http://doi.acm.org/10.1145/2366145.2366154, doi:10.1145/2366145.2366154. 1

[FSL*15] FISHER M., SAVVA M., LI Y., HANRAHAN P., NIESSNER M.: Activity-centric scene synthesis for functional 3D scene modeling. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 179:1–179:13. URL: http://doi.acm.org/10.1145/2816795.2818057, doi:10.1145/2816795.2818057. 1

[IKH*11] IZADI S., KIM D., HILLIGES O., MOLYNEAUX D., NEWCOMBE R., KOHLI P., SHOTTON J., HODGES S., FREEMAN D.,

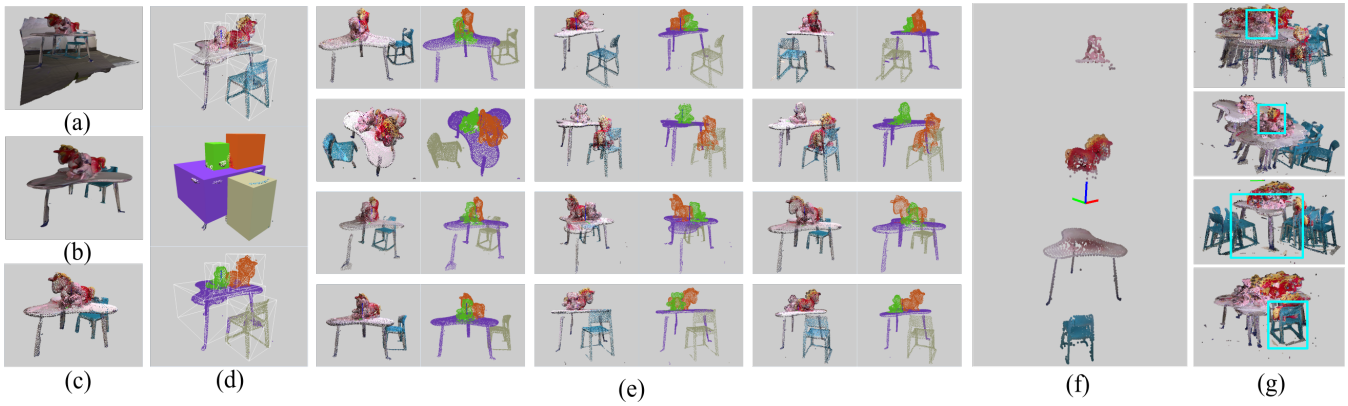


Figure 14: *Segmentation and registration on real data. (a) Scanned mesh using method in [NZIS13]. (b) Remove walls and floors by plane fitting. (c) Sampled point set using [CCS12]. (d) With roughly placed boxes on only one point set, the points are initially segmented in this one point set. Note that parts of the chair legs are segmented to the table due to the rough box placement by users. (e) Pairs of input point sets and corresponding segmentation results. (f) The final Gaussian centroids for the five objects in the scene. (g) Verification of the registration result by aligning all point sets with respect to each object. The light blue rectangle highlights the object that is aligned together. Except the aligned object, the other objects are placed quite messy since they came from different point sets and have different arrangement relative to the aligned object.*

DAVISON A., FITZGIBBON A.: 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* (New York, NY, USA, 2011), UIST '11, ACM, pp. 559–568. URL: http://doi.acm.org/10.1145/2047196.2047270, doi:10.1145/2047196.2047270. 2

[JGSC15] JIA Z., GALLAGHER A. C., SAXENA A., CHEN T.: 3d reasoning from blocks to stability. *IEEE Trans. on Pattern Analysis and Machine Intelligence 37*, 5 (May 2015), 905–918. doi:10.1109/TPAMI.2014.2359435. 2

[JV11] JIAN B., VEMURI B. C.: Robust point set registration using gaussian mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence 33*, 8 (Aug 2011), 1633–1645. doi:10.1109/TPAMI.2010.223. 2

[LPRD16] LI Y., PALURI M., REHG J. M., DOLLAR P.: Unsupervised learning of edges. In *CVPR* (June 2016), pp. 1619–1627. doi:10.1109/CVPR.2016.179. 3

[LZW*15] LIU Z., ZHANG Y., WU W., LIU K., SUN Z.: Model-driven indoor scenes modeling from a single image. In *Proceedings of the 41st Graphics Interface Conference, Halifax, NS, Canada, June 3-5, 2015* (2015), pp. 25–32. URL: http://dl.acm.org/citation.cfm?id=2788896. 1

[MS10] MYRONENKO A., SONG X.: Point set registration: Coherent point drift. *IEEE Trans. on Pattern Analysis and Machine Intelligence 32*, 12 (Dec 2010), 2262–2275. doi:10.1109/TPAMI.2010.46. 2

[MSL*11] MERRELL P., SCHKUFZA E., LI Z., AGRAWALA M., KOLTUN V.: Interactive furniture layout using interior design guidelines. *ACM Trans. Graph. 30*, 4 (July 2011), 87:1–87:10. URL: http://doi.acm.org/10.1145/2010324.1964982, doi:10.1145/2010324.1964982. 1

[NXS12] NAN L., XIE K., SHARF A.: A search-classify approach for cluttered indoor scene understanding. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 137:1–137:10. URL: http://doi.acm.org/10.1145/2366145.2366156, doi:10.1145/2366145.2366156. 1

[NZIS13] NIEßNER M., ZOLLHÖFER M., IZADI S., STAMMINGER M.: Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 169:1–169:11. URL: http://doi.acm.org/10.1145/2508363.2508374, doi:10.1145/2508363.2508374. 2, 9, 11

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: PointNet: Deep learning on point sets for 3d classification and segmentation. In *CVPR* (2017). 7

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: GrabCut: interactive foreground extraction using iterated graph cuts. vol. 23, pp. 309–314. URL: http://doi.acm.org/10.1145/1015706.1015720, doi:10.1145/1015706.1015720. 2

[RMBK06] ROTHER C., MINKA T. P., BLAKE A., KOLMOGOROV V.: Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR* (2006), pp. 993–1000. URL: https://doi.org/10.1109/CVPR.2006.91, doi:10.1109/CVPR.2006.91. 2

[TS10] TOMBARI F., STEFANO L. D.: Object recognition in 3d scenes with occlusions and clutter by hough voting. In *Fourth Pacific-Rim Symposium on Image and Video Technology* (Nov 2010), pp. 349–355. doi:10.1109/PSIVT.2010.65. 3

[TSS16] TANIAI T., SINHA S. N., SATO Y.: Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR* (2016), pp. 4246–4255. URL: https://doi.org/10.1109/CVPR.2016.460, doi:10.1109/CVPR.2016.460. 2, 3

[XCF*13] XU K., CHEN K., FU H., SUN W.-L., HU S.-M.: Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models. *ACM Trans. Graph. 32*, 4 (July 2013), 123:1–123:15. URL: http://doi.acm.org/10.1145/2461912.2461968, doi:10.1145/2461912.2461968. 1

[XHS*15] XU K., HUANG H., SHI Y., LI H., LONG P., CAICHEN J., SUN W., CHEN B.: Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 177:1–177:14. URL: http://doi.acm.org/10.1145/2816795.2818075, doi:10.1145/2816795.2818075. 2, 3