

Interactive Point Set Joint Registration and Co-segmentation

Paper 1049

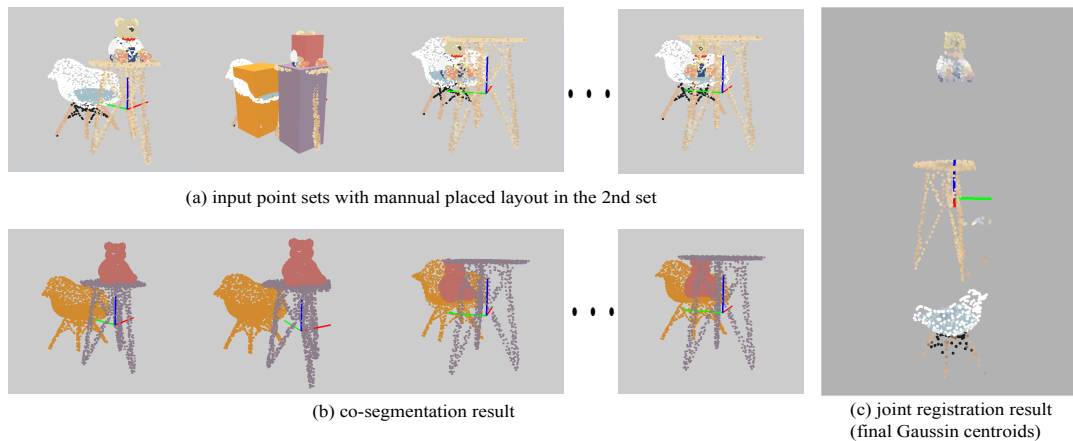


Figure 1: Given a series of point sets and user-placed object layout boxes in one point set (a), our method jointly segment and register objects in the point sets to generate object-level co-segmentation (b) and registration to object models represented as GMM (c).

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 the 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 to intuitively control the optimization process. The experiment results on a group of synthetic and scanned point clouds demonstrate that our method is helpful and effective for the 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 object models, resulting in scenes with object segmentation and annotation naturally available, or to manually segment and annotate existing 3D scenes. This procedure can be tedious and time-consuming, despite many efforts of improving the interaction experience [MSL*11, XCF*13]. Another way is to automatically generate scenes from 3D shape

models according to images [LZW*15, CLW*14]. In these methods, a retrieval procedure is usually needed and inevitably limit the result to a certain set of 3D models, without producing the actual 3D shapes that appear in the input images.

Generating scene models directly from captured point clouds will significantly facilitate dataset construction and increase the dataset variety. 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 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, DNZ*16]. Thus, a generic object-level segmentation and modeling method from scanned 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 the block-based stability [JGSC15], or the motion consistency of rigid objects [XHS*15]. While a robot is employed to do proactive push and movement tracking is used to verify and iteratively improve the object-level segmentation result [XHS*15], it remains significantly challenging to recover the motion consistency in a *non-invasive way*. (cxj: still not comfortable with the word "invasive"... This word is mainly used in cancer or medical procedure. Or provide more detailed explanation on "non-invasive".) (cxj: Do we "recover" the motion consistency or "use" the motion consistency to do co-segmentation and registration?) (hsy: "Invasive" is the original word used in [XHS*15].)

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 along the time, we hope to segment the objects in a scene from the scanned point clouds 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 the correspondences of objects in different point sets due to the occlusions by human bodies in the first scheme or sparse sampling on times in the second scheme. In the first scheme, additional 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 data leads us to the problem of coupled joint registration and co-segmentation.

In this problem, registration and segmentation are entangled in each other. On the one hand, the segmentation 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 depends on the segmentation to break the problem into a series of rigid joint registration instead of a joint registration with non-coherent point drift. Non-coherent point drift means that a pair of points is 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 this pair of points actually belong to different objects. This makes the big difference from non-rigid registration problems where point motions are smooth everywhere. We employ a group of Gaussian mixture models and each of these Gaussian mixture models represents a potential object. This model unentangles the registration and segmentation in the way that the segmentation can be done by evaluating the probability of points belongs to the Gaussian mixture models and the registration can be done by evaluating rigid registration against each Gaussian 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 point sets.
2. We propose a generative model to simultaneously solve the joint registration and co-segmentation of point sets.
3. We design a tool for joint registration and co-segmentation based on the generative model and release it to public.

2. Related Work

In this section we describe a series of previous work on point set processing and how we draw experiences from these methods.

Point Set Registration with GMM Representation. Gaussian mixture models are widely used as the representation 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 non-rigid 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 Gaussian mixture model 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, all point sets are treated equally as realizations of a Gaussian mixture model and the registration is cast into a clustering problem [EKBHP14]. 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 [EKBHP14] to simultaneously handle joint registration and co-segmentation. The difference between our method and non-rigid registration is that we can handle the non-coherent point drift by simply estimating independent transformation for each segmentation. (cxj: The challenging part is the non-coherent point drifts in different objects in point sets which brings a large number of unknown discrete labels in the formulation.) (hsy: There is no such discrete variables in our formulation.)

Image segmentation and co-segmentation Due to the challenges in automatic segmentation problems, many interactive methods have been proposed to leverage human's interaction on high-level hints and computational abilities of computers. An influential work for interactive image segmentation is GrabCut [RKB04]. It uses two Gaussian mixture models, one for foreground and the other for background. To initialize these two Gaussian mixture models, a rectangle is manually placed to contain the foreground. Our design of interaction draws on the experience from [RKB04]. The difference is that our interaction is designed for 3D space and can handle multi-object segmentation rather than foreground-background segmentation. [TSS16] jointly recovers co-segmentation and dense per-pixel correspondences in two images. Its co-segmentation is limited to foreground-background segmentation. Our work solves a similar problem for multiple 3D point sets. (cxj: There are many image co-segmentation papers. If you want to discuss it, you should discuss more.)

Segmentation from Motion. Object motion, as a strong hint for object segmentation, is widely used in many works. [XHS*15] employs a robot to do proactive push and track the motion to learn object segmentation. [LPRD16] exploits the motion in a video and uses the motion edges as the training data to learn an edge detector for images. These methods lean on the motion that is continuous in time and can be tracked. In comparison, our method can handle motion that is not continuous in 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 problem in point sets. Our method should be classified as one of the correspondence grouping method. Comparing to previous methods that uses (cxj: use what?)(hsy: you added "that uses" here I'm not sure) [TS10,CB07], our method simultaneously solves the problem for multiple target models in multiple scenes.

3. Method Overview

In this section, we introduce our formulation of the joint registration and co-segmentation problem for point sets. Tabel 1 lists all the symbols used in our formulation. 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 different objects to M different point sets.

3.1. Basic Formulation

For robustness, we do not model the point sets as a simple composition of transformed 3D points in each object model. Instead, we model the point sets as realizations 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

$$\underbrace{\{\{\mathbf{x}_1, \Sigma_1\}, \dots, \{\mathbf{x}_{K_1}, \Sigma_{K_1}\}\}}_{O_1}, \underbrace{\{\{\mathbf{x}_{K_1+1}, \Sigma_{K_1+1}\}, \dots, \{\mathbf{x}_{K_1+K_2}, \Sigma_{K_1+K_2}\}\}}_{O_2}, \dots, \underbrace{\{\{\mathbf{x}_{K_S+1}, \Sigma_{K_S+1}\}, \dots, \{\mathbf{x}_{K_S+K_n}, \Sigma_{K_S+K_n}\}\}}_{O_n} \quad (1)$$

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. Each object model 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 $\mathbf{x}_k \in O_n$. Hence, for each point \mathbf{v}_{mi} in a point set \mathbf{V}_m , given object models n and their rigid transformations $\{\phi_{mn}\}$ to the point sets, we can write

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

which treat the i^{th} observed point \mathbf{v}_{mi} from the m^{th} point set as a sample point generated by a large Gaussian mixture model that represent N objects all together.

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

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

In these parameters, the $\{\phi_{mn}\}_{m=1, n=1}^{M, N}$ are the transformations for joint registration problem. $\{\mathbf{x}_k, \Sigma_k\}_{k=1}^{K_{all}}$ are the Gaussian models which are predefined to be one of the N objects. $\{p_k\}_{k=1}^{K_{all}}$ are weights for these Gaussian models. After we estimate these parameters, we can assign each point in all input point sets to one of the Gaussian models. 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 sets 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 the EM framework of Expectation-Maximization. In particular, we bring in hidden variables as:

$$\mathcal{Z} = \{z_{mi} | m = 1 \dots M, i = 1 \dots L_m\}, \quad (4)$$

such that $z_{mi} = k (k = 1, 2, \dots, K_{all})$ assigns the observed point \mathbf{v}_{mi} to the k^{th} component of the Gaussian mixture model. We aim to

Table 1: Table of symbols used in the paper.

Symbol	Definition
M	The number of input point sets.
\mathbf{V}_m	The m^{th} input point set.
\mathcal{V}	The input point sets $\{\mathbf{V}_m\}_{m=1}^M$.
L_m	The number of points of the m^{th} point set \mathbf{V}_m .
\mathbf{v}_{mi}	The i^{th} point of \mathbf{V}_m .
\mathbf{f}_{mi}	The point-wise feature vector of \mathbf{v}_{mi} .
z_{mi}	The latent parameter for \mathbf{v}_{mi} . $z_{mi} = k$ means \mathbf{v}_{mi} is generated by k^{th} Gaussian.
\mathcal{Z}	$\mathcal{Z} = \{z_{mi} m = 1 \dots M, i = 1 \dots L_m\}$.
N	The number of objects in the scene.
K_n	The number of Gaussian models for the n^{th} object.
K_S	The sum of $\{K_1, K_2, \dots, K_{n-1}\}$ $K_S = \sum_{i=1}^{n-1} K_i$
K_{all}	The total number of Gaussian models of all objects. $K_{all} = \sum_{n=1}^N K_n$.
p_k	The weight of k^{th} Gaussian. $\sum_{k=1}^{K_{all}} p_k = 1$.
\mathbf{x}_k	The centroid of k^{th} Gaussian model.
\mathbf{x}_k^v	The centroid of k^{th} Gaussian for point position.
\mathbf{x}_k^f	The centroid of k^{th} Gaussian for point feature.
Σ_k	The covariance matrix of k^{th} Gaussian model.
σ_k	$\Sigma_k = \sigma_k^2 \mathbf{I}$, where \mathbf{I} is an identity matrix.
σ_k^v	Gaussian covariance parameter for point position
σ_k^f	Gaussian covariance parameter for point feature
ϕ_{mn}	Rigid transformation from object O_n to point set \mathbf{V}_m .

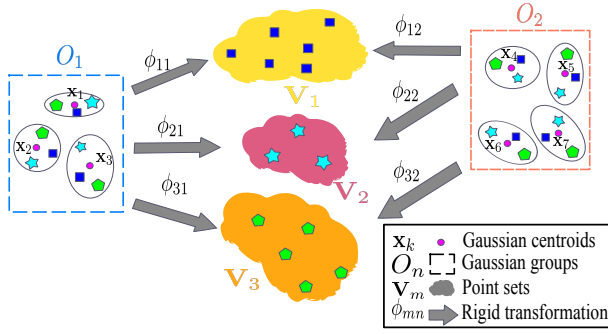


Figure 2: Illustration of formulation.

maximize the expected complete-data log-likelihood:

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

This formulation can be seen as an adaption of the joint registration formulation in [EKBHP14], upon which we separate Gaussian models into groups to express multiple objects. The latent parameter \mathcal{Z} that assigns observed points to Gaussian models can naturally indicate the object level segmentation. Under the assumption that the input points are independent and identically distributed, we can rewrite the objective defined in Eq. (5) into:

$$\Theta = \arg \max_{\Theta} \sum_{mik} \alpha_{mik} (\ln p_k + \ln P(\mathbf{v}_{mi} | z_{mi} = k; \Theta)), \quad (6)$$

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:

$$\Theta = \arg \max_{\Theta} \sum_{mik} \alpha_{mik} (\|\mathbf{v}_{mi} - \phi_{mn}(\mathbf{x}_k)\|_{\Sigma_k}^2 + \ln |\Sigma_k| - 2 \ln p_k), \quad (7)$$

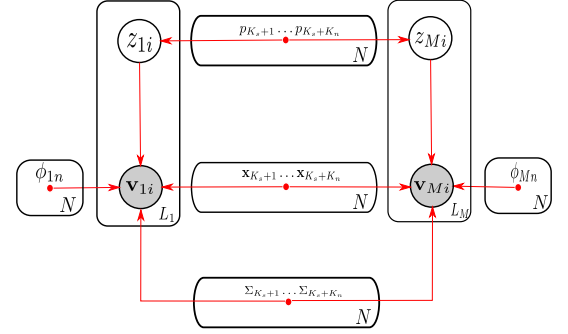
where the $|\cdot|$ denotes the determinant and $\|\mathbf{x}\|_{\Lambda}^2 = \mathbf{x}^T \mathbf{A}^{-1} \mathbf{x}$. It is predefined that \mathbf{x}_k is one of the Gaussian centroids used to represent n^{th} object, which is why we apply transformation ϕ_{mn} on to the \mathbf{x}_k . For the convenience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

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.

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

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

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



where Λ_{mn} is $K_n \times K_n$ diagonal matrix with elements $\lambda_{mnk} = \frac{1}{\sigma_k} \sqrt{\sum_{i=1}^{L_m} \alpha_{mik}}$, L_m is the number of point for the m^{th} input point set, $\mathbf{X}_n = [\mathbf{x}_{K_S+1}, \mathbf{x}_{K_S+2}, \dots, \mathbf{x}_{K_S+K_n}]$ where $K_S = \sum_{i=1}^{n-1} K_i$ 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 $\mathbf{W}_{mn} = [\mathbf{w}_{m(K_S+1)}, \mathbf{w}_{m(K_S+2)}, \dots, \mathbf{w}_{mk}, \dots, \mathbf{w}_{m(K_S+K_n)}]$ where $K_S = \sum_{i=1}^{n-1} K_i$, in which \mathbf{w}_{mk} is a weighted point as

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

This problem has a similar solution with [EKBHP14]. 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 group of \mathbf{x}_k corresponding to multiple objects in our Gaussian models. The optimal can be given by:

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

$$\mathbf{t}_{mn}^* = \frac{1}{tr(\Lambda_{mn}^2)} (\mathbf{W}_{mn} - \mathbf{R}_{mn} \mathbf{X}_n) \Lambda_{mn}^2 \mathbf{e} \quad (12)$$

where $[\mathbf{U}_{mn}, \mathbf{S}, \mathbf{V}_{mn}] = \text{svd}(\mathbf{W}_{mn} \Lambda_{mn} \mathbf{P}_{mn} \Lambda_{mn} \mathbf{X}_n^T)$ 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. $\mathbf{C}_{mn} = \text{diag}(1, 1, |\mathbf{U}_{mn}| |\mathbf{V}_{mn}|)$.

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

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

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

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

$$p_k^* = \frac{\sum_{i=1}^{L_m} \alpha_{mik}}{M} \quad (15)$$

$$y_{mi}^* = \arg \max_n \sum_{k=\sum_{s=1}^{n-1} K_S+1}^{\sum_{s=1}^n K_S} \alpha_{mik} \quad (16)$$

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

3.2. Bilateral Formulation

When considering point-wise features, we can add bilateral terms into the generative model.

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

where \mathbf{f}_{mi} is the feature vector for point \mathbf{v}_{mi} and \mathbf{x}_k^f is the feature vector for k^{th} point in object model. As shown in the formulation, there is no transformation applied onto \mathbf{x}_k^f , which means that this formulation is only suitable to the features that is rotation and translation invariant. For example, the point color vector (for all the result in this paper we use RGB color as feature vector) $[red_{mi}, green_{mi}, blue_{mi}]$ is a suitable feature for this formulation. In this formulation $\mathcal{N}(\mathbf{v}_{mi} | \phi_{mn}(\mathbf{x}_k^v), \sigma_{v_k})$ is the spatial term and $\mathcal{N}(\mathbf{f}_{mi} | \mathbf{x}_k^f, \sigma_k^f)$ 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.

$$\alpha_{mik} = \frac{p_k P_v(\mathbf{v}_{mi}, \phi_{mn}(\mathbf{x}_k^v), \sigma_{v_k}) P_f(\mathbf{f}_{mi}, \mathbf{x}_k^f, \sigma_k^f)}{\sum_{s=1}^{K_{all}} p_s P_v(\mathbf{v}_{mi}, \phi_{mn}(\mathbf{x}_s^v), \sigma_{v_s}) P_f(\mathbf{f}_{mi}, \mathbf{x}_s^f, \sigma_s^f)} \quad (18)$$

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 (11) and (12).

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

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

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

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

4. Initialization and Optimization

In this section, we summarize the entire framework for our algorithm and explain how do we initialize the parameters and guide the optimization with interaction. Based on our formulation in Sec. 3, our algorithm can be summarized as in Algorithm 1.

Algorithm 1 Joint Registration and Co-segmentation (JRCS)

Input:

$\{\mathbf{V}_m\}$: M 3D point sets

Θ^0 : Initial parameters

$\{\beta_{ik}\}_m$: layout based 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 a bilateral formulation);

4. Alter α_{mik}^q with $\{\beta_{ik}\}_m$ according to Eq. (22);

5. M-step-a: use $\alpha_{mik}^q, \mathbf{x}_k^{q-1}$ to estimate $\{\mathbf{R}_{mn}^q\}$ and $\{\mathbf{t}_{mn}^q\}$ according to Eqs. (11)(12);

6. M-step-b: use $\alpha_{mik}^q, \{\mathbf{R}_{mn}^q\}$ and $\{\mathbf{t}_{mn}^q\}$ to update other parameters for Gaussian models according to Eqs. (13)(14)(15)(16) (or use Eqs. (19)(20) for a bilateral formulation);

7. $q \leftarrow q + 1$

8. **until** Convergence (cxj: what is the convergence conditions?)

9. **return** Θ^q

4.1. Interaction

Unfortunately, there are a large number of parameters that can not be easily initialized in our formulation. In this subsection we first introduce our design of interaction, which is intuitive for users to input the semantic prior this way. We then explain how we can easily initialize those parameters for our optimization based on the manual input.

As demonstrated in Figure 3, we let user choose one of the point sets and place boxes in it to indicate the layout for this point set. From this, we can easily initialize the total number of objects N and determine $\{K_n\}$ which is the numbers of Gaussian mixture models used to represent each object. These two parameters are difficult to

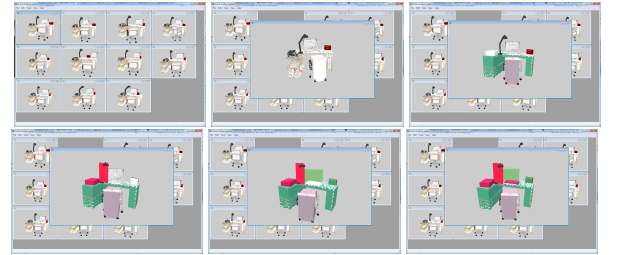


Figure 3: From the first to the sixth, the nine images show the procedure of interaction: the user pick one point set and place boxes in it to indicate the layout for this point set. The box in white is the box currently under editing. The boxes in other colors are boxes placed to represent object layouts. One color represents one object. The interaction allows multiple boxes to represents same object.(e.g. the desk is represented by three boxes in same color)

be initialized without semantic prior, but with the input of the users we can naturally initialize the N as the number of different color

label and the K_n as

$$K_n = \frac{V_n}{\sum V_n} K_{all}, \quad (21)$$

in which the V_n represent the total volume of the boxes in the n^{th} color and the K_{all} is initialized as $K_{all} = \frac{\text{median}(L_m)}{2}$ and $\{L_m\}$ are point numbers of M input point sets. This is an empirical choice borrowed from [EKBHP14].

The expectation maximization framework is easily converged to a local optimal. To cope with this problem we further use this layout (boxes) from interaction as a soft constraint to guide the optimization and constrain the shape of generated object model. Such constraint is enforced by simply altering the posterior probability α_{mik} as

$$\alpha_{mik}^* = \frac{\alpha_{mik} \beta_{mik}}{\sum_{i,k} \alpha_{mik} \beta_{mik}} \quad (22)$$

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

$$\beta_{mik} = \begin{cases} 1 & \mathbf{v}_{mi} \in B_n \\ \exp(-\frac{\min_{\mathbf{v}_{mj}} \|\mathbf{v}_{mi} - \mathbf{v}_{mj}\|_2^2}{L}) & \mathbf{v}_{mi} \notin B_n \text{ and } \mathbf{v}_{mj} \in B_n \end{cases} \quad (23)$$

where the B_n is a point set that is enclosed by the boxes used to represent the layout of n^{th} object. The k^{th} Gaussian model is pre-defined to be one of the Gaussians used to represent n^{th} object. $\min_{\mathbf{v}_{mj}} \|\mathbf{v}_{mi} - \mathbf{v}_{mj}\|_2^2$ is actually the squared euclidean distance from point \mathbf{v}_{mi} to the point set B_n , as we define the distance from a point to a point set as the minimum distance from the point to any point inside the point set. L here is a constant number with $L = 2r^2$, and r is the median of the radius of input point sets. The radius of a point set is half of length of diagonal line of its axis-aligned bounding box. This alteration on posterior probability is only done with

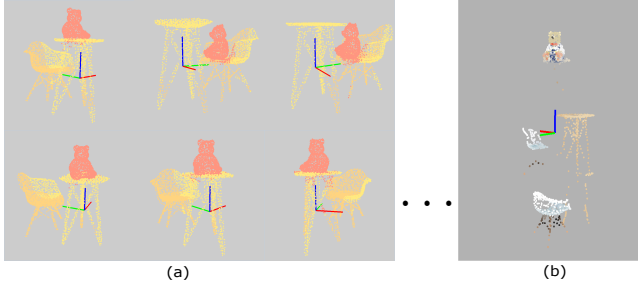


Figure 4: This figure shows an example result when converges to a local optimal. (a) is the result of segmentation of this local optimal. (b) is the final centroids of latent model. It shows that from top to down the 2nd and 3rd object model both include part of the table and part of the chair.

the probability related to the m^{th} point set that have the manual input layout (the boxes) in it. This alteration can help prevent the optimization from converging to a local optimal as in Figure 4. The result from the Figure 4 has the same input and initialization with the result from Figure 1, but it does not use the posterior alteration as a soft constraint.

4.2. Initialization

Initialization of Θ : We start with determining the total number of Gaussian model K_{all} as we explained in Sec. 4.1. We set $p_k = \frac{1}{\sum K_n}$, which means each Gaussian has the same weight at the beginning. We separate the total K_{all} Gaussian models into N groups to represent N objects. Each group has K_n Gaussian models based on Eq. (21). We implement this by recording the start and end indices of the N objects. In other words, we record $\{0, K_1, K_1 + 1, K_1 + K_2, \dots, \sum_{n=1}^{N-1} K_n + 1, \sum_{n=1}^N K_n\}$. $\{\mathbf{x}_k\}_n$ are Gaussian centroids of n^{th} group and they are initialized as a 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 center of the n^{th} sphere is $\mathbf{c}_n = (0, 0, z_n)$, where $z_n \in \{-(N-1)r, -(N-3)r, \dots, (N-1)r\}$. (cxj: What is N? In total, there would be $2N - 1$ centers?) (hsy: No, The common difference is $2r$ here, it is $\{-2r, 0, 2r\}$ when $N = 3$ and $\{-3r, -r, r, 3r\}$ when $N = 4$) This means that the object models are vertically arranged in latent space as shown in Figure 1(c) and in Figure 4(b). We choose vertical arrangement for groups of object merely for the convenience of visualization. (cxj: So you can also set the spheres horizontally or slantly or arbitrarily for registration only?) We choose spheres as the initial shape so that we can initialize all the \mathbf{R}_{mn} to identity matrix. For the \mathbf{t}_{mn} we initialize them as $\mathbf{t}_{mn} = -\mathbf{c}_n$ so that all the object model starting with position at origin point when they are transformed to the space of each input set. However, if the m^{th} input point set has the manually placed layout, we treat the associated \mathbf{t}_{mn} differently. For this case we have:

$$\mathbf{t}_{mn} = \frac{\sum_{\mathbf{v}_{mi} \in B_n} \mathbf{v}_{mi}}{N(B_n)} - \mathbf{c}_n \quad (24)$$

where $N(B_n)$ here is the number of element in B_n and B_n is the point set that is enclosed by the manually input layout (box) indicating arrangement of the n^{th} object. (cxj: You have many boxes, B_n is which one?. The number n is also confusing while you use n for sphere center index.) (hsy: n is always indices for object and B_n is the box for n^{th} object)

4.3. Hot Intervention Mechanism

(cxj: I am curious where do you get this word? Any reference?) Our current implementation of optimization is quite slow (fails to converge in an interactive-rate time) especially when the numbers of points inside the input point sets are large and it is possible for our optimization to stuck in a local optimal, requiring the guide from the manual input. As a compensation for these drawbacks, we implement a hot intervention mechanism, allowing the manual input to take effect during the optimization process. Theoretically, this is possible due to the i.i.d assumption, under which the calculation of posterior probability is independent for each input point set. Even after the optimization is started, we can still allow the user to add more layouts (cxj: what do you mean by 'add layouts'?) for other point sets and the program can do the same alteration as (22) in the next iteration. Figure 5 shows how the users can use the hot intervention mechanism within our tool.

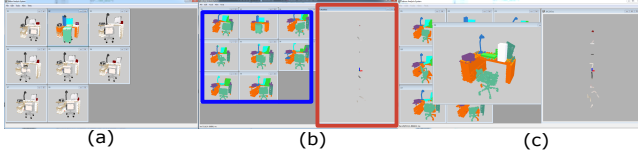


Figure 5: This figure shows the hot intervention mechanism. (a) The input point sets with manually placed layout in the 2nd point set. (b) For each iteration, the instant segmentation results for all point sets are shown in the blue region while the object models (the shape of the centroids of the Gaussian models) are shown in the red region. (c) The user picks another input point set and adds more boxes targeting the incorrect segmentation to further guide the optimization when the optimization is running. (cxj: highlight the newly added box. Do not show the entire interface.. just show the data.)

Table 2: IOU scores on three 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 [QSMG16]

Datasets	Child Shelf	Office Desk	Living Room
JRCS Basic	-	-	-
JRCS Bilateral	-	-	-
PointNet	-	-	-

5. Experiment and Discussion

In this section, we will show a series of experimental results including a) evaluation for co-segmentation and joint registration on synthetic data in which we compare our method to previous methods b) investigation on the robustness of our method on point completeness and amount of user interaction c) testing on one group of real data and verifying the result visually.

In the experiments, the synthetic datasets are generated as follows:

1. Acquire meshes for objects from 3D Warehouse.
2. Turn the meshes into point sets with sampling method of [CCS12]
3. Manually place and move the point sets of objects into different point sets of scenes.

The real dataset (as shown in 10) is acquired as follows:

1. Scan the scenes with method of [IKH*11]
2. Remove walls and floors with plane fitting
3. Turn the meshes of scenes into point sets with sampling method of [CCS12]

5.1. Evaluation for Co-segmentation on Synthetic Data

From the perspective of co-segmentation, we quantitatively evaluate our algorithm on three group of synthetic data of indoor scenes. To estimate the power of the algorithm, we only input layout for one point set in each group for initialization and do not add further interaction. For numerical estimation, we calculate the intersection over

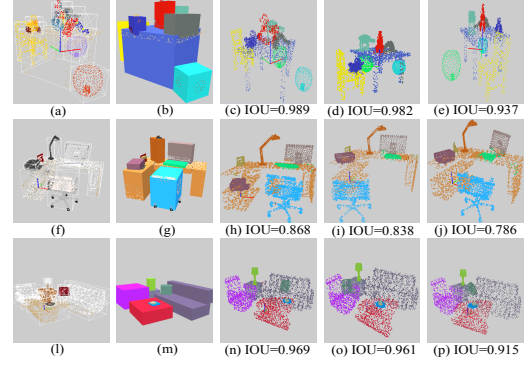


Figure 6: Segmentation evaluations on three groups of synthetic data (From top to bottom: child table, office desk, living room). Each group of data have 13 point sets. Columns from left to right: (a) Examples of point sets for each group of data. The groundtruth bounding boxes of objects in the 3D model are shown in white. (b) Manually placed boxes (cxj: on one point set); Three segmentation results with (c) maximum IOU scores, (d) median IOU scores and (e) minimum IOU scores in the groups.

union (IOU) scores for the result segmentation against ground-truth segmentation. In order to compare with the semantic segmentation method of PointNet [QSMG16], we produce the "Living Room" with objects inside 13 object classes of PointNet [QSMG16]. Table 2 shows the numeric result and Figure 6 shows some samples of visual result. From the evaluation, we want to discuss one interesting observation:

For all three groups, the point set with highest IOU score is not the same as the point set equipped with manually placed layout. In other words, the point sets from Figure 6(c) are not the same point sets from Figure 6(b). In the first group of data (dataset of child table in Figure 6), the segmentation result of the point set with layout is even the second worst in the sense of IOU score. We believe this is because that the manually placed layout is not accurate respecting to point-wise segmentation. At early iterations of the optimization, the alteration in (22) can serve as a soft constraint to help constraining the shape of object, but in the final iterations the alteration will obstruct the further improvement of segmentation for the correspondent point set.

5.2. Evaluation for Joint Registration on Synthetic Data

From the perspective of joint registration, we first compare our method with [EKBHP14] on the synthetic point sets released by [EKBHP14]. These data contains four point sets of Stanford Bunny with different noise and outliers. From the experiment result shown in Table 3 and Figure 7, we can see that when dealing with one object, our method have similar result with [EKBHP14].

We then evaluate the result by transferring the point cloud of objects to each input point set based on result $\{\phi_{mn}\}$ and calculating the average distance from a point to its true correspondent point for

Table 3: This table shows the RMSE of joint registration on 4 point sets of Stanford Bunny. JRMPC is the method of [EKBHP14]. 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

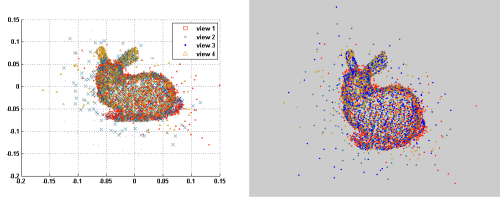


Figure 7: This figure shows the visual result of joint registration on 4 point sets of Stanford Bunny by JRMPC([EKBHP14] at left) and JRCS Basic(our basic formulation at right)

each input point set. We use this average distance as fitness error to evaluate the registration quality respect to each input set.

$$\mathbf{t}_{mn} = \frac{\sum_{\mathbf{v}_{mi} \in B_n} \mathbf{v}_{mi}}{N(B_n)} - \mathbf{c}_n \quad (25)$$

Table 4 shows the result of this evaluation. For this evaluation we want to discuss that:

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.

5.3. Investigation on Interaction

a) with respect to improvement on the point set with interaction (prove that the interaction is only a initialization). b) with respect to number of interactions

Table 4: Registration errors of the three groups of synthetic data in Figure 6. The errors are measured in meter. (cxj: 3 cm is actually large error...)

Dataset	Maximum Error	Median Error	Minimum Error
Child Shelf	0.0715	0.0112	4.91e-005
Office Desk	0.189	0.0618	0.00518
Living Room	0.132	0.0563	0.0301

5.4. Investigation on Influence of Point Incompleteness

In previous evaluation on synthetic data, we use data that the objects are evenly and completely covered by the sampled points. In this subsection, we investigate how the point set incompleteness affect result of our algorithm. To do this, we pick a group of point sets that can converge well (IOU > 99% for each point set) when the point sets are complete. We then gradually remove certain percentage (0% – 30%) of points from each point sets. The Figure 8 shows how the IOU score is affected with the increased point set incompleteness.

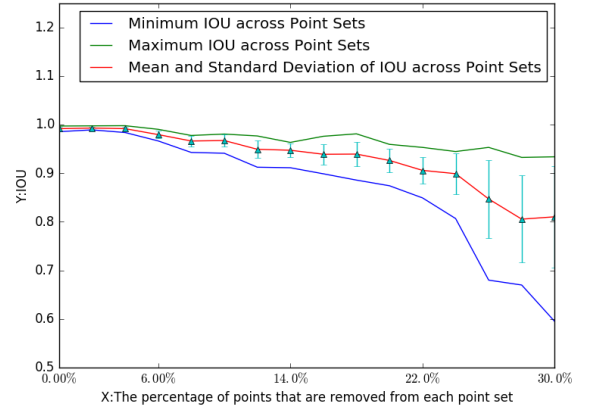


Figure 8: This figure shows how the data incompleteness affect the IOU score of co-segmentation

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 9 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 10 shows the segmentation and registration results on a group of scanned point sets. From Figure 10(e), we can see that all input point sets are partitioned into objects. From the Figure 10(g), we can verify that the object from each input set are aligned together by the result transformation. (cxj: More discussions about the robustness of our method according to reviewer's comments..) **Different user interaction**

5.6. Limitations and Future Work

(hsy: Need to add discussion on failure cases) The biggest problem holding us back is the time performance of our current implementation. Due to i.i.d. assumption most calculation of our algorithm can actually be parallelized. We plan to implement a new version on GPU cluster so that we can explore more potentials of our algorithm. For example, to integrate semantic feature vectors (generated by neural networks) into it and try it on a scene of a larger scale like [CPI6]. As advocated in the recent work of [EKT*16], it may

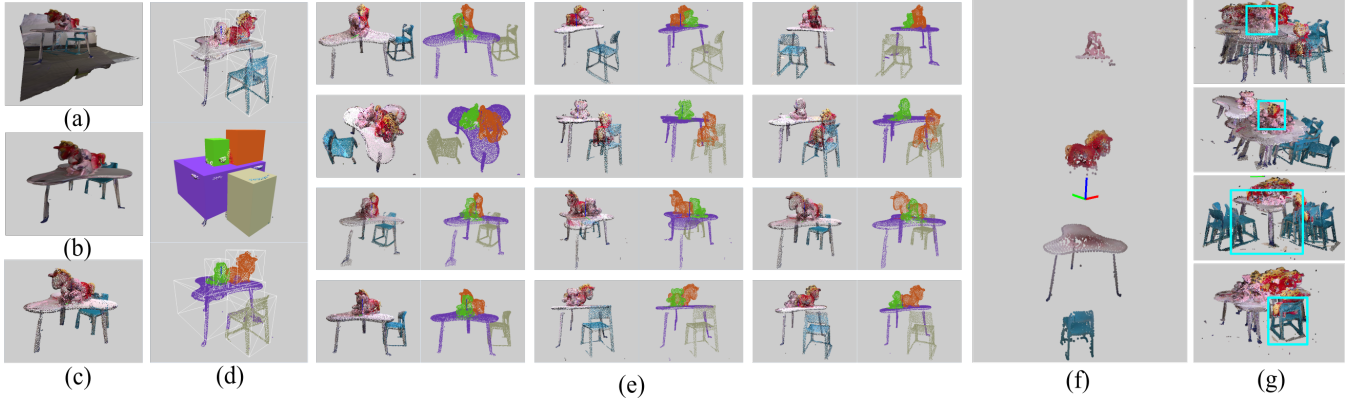


Figure 10: Segmentation and registration on real data. (a) Scanned mesh using method in [NZIS13]. (b) Remove walls and floors by plane fitting. (c) Sampled point set using [CCS12]. (d) With roughly placed boxes on only one point set, the points are initially segmented in this one point set. Note that 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. (cxj: Again, i think (g) is really messy and confusing...)

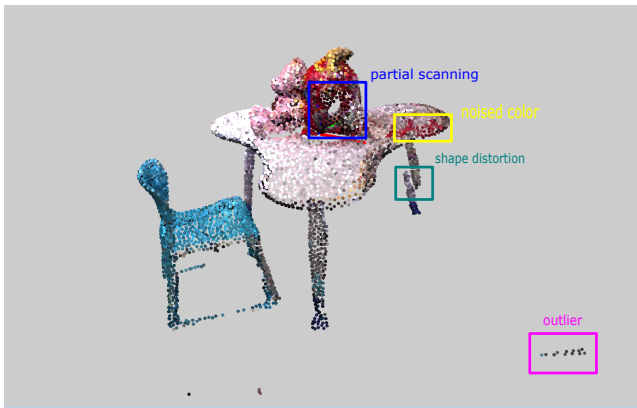


Figure 9: This figure highlights the common challenges on real data.

be a good idea to do the joint registration and co-segmentation with hierarchical GMM representation when for scenes in larger scales.

References

- [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. URL: <http://dx.doi.org/10.1016/j.patrec.2007.02.009>, 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 Transactions 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. 7, 8, 9
- [CLW*14] CHEN K., LAI Y.-K., WU Y.-X., MARTIN R., HU S.-M.: Automatic semantic modeling of indoor scenes from low-quality rgbd 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, 8
- [DNZ*16] DAI A., NIESSNER M., ZOLLHÖFER M., IZADI S., THEOBALT C.: Bundlefusion: Real-time globally consistent 3d reconstruction using online surface re-integration. *arXiv preprint arXiv:1604.01093* (2016). 2
- [DSS12] DEMA M. A., SARI-SARRAF H.: 3d scene generation by learning from examples. In *Multimedia (ISM), 2012 IEEE International Symposium on* (Dec 2012), pp. 58–64. doi:10.1109/ISM.2012.19. 1
- [EKBHP14] EVANGELIDIS G. D., KOUNADES-BASTIAN D., HORAUD R., PSARAKIS E. Z.: *A Generative Model for the Joint Registration of Multiple Point Sets*. Springer International Publishing, Cham, 2014, pp. 109–122. URL: http://dx.doi.org/10.1007/978-3-319-10584-0_8, doi:10.1007/978-3-319-10584-0_8. 2, 4, 6, 7, 8
- [EKT*16] ECKART B., KIM K., TROCCOLI A., KELLY A., KAUTZ J.: Accelerated generative models for 3d point cloud data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016), pp. 5497–5505. doi:10.1109/CVPR.2016.593. 8
- [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., 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, 7
- [JGSC15] JIA Z., GALLAGHER A. C., SAXENA A., CHEN T.: 3d reasoning from blocks to stability. *IEEE Transactions 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 Transactions 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (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 *Graphics Interface Conference* (2015). 1
- [MS10] MYRONENKO A., SONG X.: Point set registration: Coherent point drift. *IEEE Transactions 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. 8, 9
- [QSMG16] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593* (2016). 7
- [RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: "grabcut": Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), SIGGRAPH '04, ACM, pp. 309–314. URL: <http://doi.acm.org/10.1145/1186562.1015720>, doi:10.1145/1186562.1015720. 2
- [TS10] TOMBARI F., STEFANO L. D.: Object recognition in 3d scenes with occlusions and clutter by hough voting. In *2010 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016). 2
- [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