# Space-Time Localization and Mapping

Minhaeng Lee, Charless C. Fowlkes
Dept. of Computer Science, University of California, Irvine
{minhaenl, fowlkes}@ics.uci.edu

## Abstract

*This paper addresses the problem of building a spatio-temporal model of the world from a stream of time-stamped data. Unlike traditional models for simultaneous localization and mapping (SLAM) and structure-from-motion (SfM) which focus on recovering a single rigid 3D model, we tackle the problem of mapping scenes in which dynamic components appear, move and disappear independently of each other over time. We introduce a simple generative probabilistic model of 4D structure which specifies location, spatial and temporal extent of rigid surface patches by local Gaussian mixtures. We fit this model to a time-stamped stream of input data using expectation-maximization to estimate the model structure parameters (mapping) and the alignment of the input data to the model (localization). By explicitly representing the temporal extent and observability of surfaces in a scene, our method yields superior localization and reconstruction relative to baselines that assume a static 3D scene. We carry out experiments on both synthetic RGB-D data streams as well as challenging real-world datasets, tracking scene dynamics in a human workspace over the course of several weeks.*

## 1. Introduction

A strategic question for scene understanding is how to leverage large repositories of images, video and other sensor data acquired over an extended period of time in order to analyze the content of a particular image. For static rigid scenes, a classic approach is to use visual SLAM, structure-from-motion, and multi-view stereo techniques to build up an explicit model of the scene geometry and appearance. These methods are well developed and have been scaled up to increasingly large problems in modeling outdoor and indoor scenes (see e.g., [1, 32, 10, 20, 9, 3]).

Such a geometric approach to scene understanding can make strong predictions about a novel test image including the camera pose (via feature matching and camera localization) and the appearance of points or surface patches projected into the image. However, reconstruction-based analysis typically neglects dynamic objects that change over time and are treated as outliers with respect to the estimation of a single rigid scene model. This problem becomes more acute as data is integrated over longer periods of time, during which an increasing proportion of objects in the scene may move non-rigidly. For example, people move on the time-scale of seconds while furnishings may shift on the time-scale of days and architecture and landscapes over years.

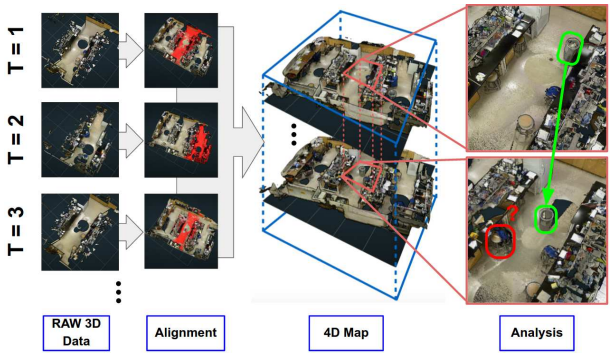



Figure 1. Raw measurements (pointclouds) captured at different times are aligned and probabilistically merged into a single 4D model that describes the spatio-temporal geometry of the scene. Model fitting reasons about occlusion, inferring complete surfaces even when they may not have been occluded at some times. The resulting integrated space-time map supports further analysis such as change detection and segmentation of dynamic objects.

In this paper, we investigate how the scope of such techniques can be extended by registering observations to a 4D reconstruction that explicitly represents geometric changes over time. We focus specifically on space-time mapping of indoor scenes where RGB-D sensors provide streams of high-quality geometric data and odometry over short time intervals and limited fields of view, but data acquired, e.g. on different days, is inconsistent due to changes in the scene. The key inferential challenge is thus distinguishing sensor noise and reconstruction errors from genuine changes in the scene geometry.

We describe a simple generative probabilistic model for 4D structure in which surface patches, specified by a spatial location, orientation and temporal extent, generate point and normal observations over time. These observations are

only recorded by the sensor if they fall within the spatio-temporal field of view of a measurement and are not occluded by other surfaces patches which exist at that same time. We assume the scene is static and rigid for the duration of each measurement and leave the problem of spatio-temporal grouping of surface patches into object tracks across time points as a post-process. Fitting the model to a time-stamped stream of measurements yields an estimate of the 4D scene structure as well as the location of the camera at each measurement time point. We term this problem *Space-Time SLAM* since it generalizes the standard offline RGB-D SLAM problem with a 4th temporal dimension[1].

The chief merits of this approach are in (1) providing robust pose estimation in the presence of changing scenes and (2) producing scene reconstructions that more accurately reflect the scene geometry at any given time point. In particular, by reasoning about occlusion, the model is capable of inferring amodal completion of surfaces which may be hidden by an object during some times but later revealed when the object moves. We quantify these benefits relative to baselines that lack an explicit temporal model using a synthetic dataset where ground-truth geometry and pose are known. We also perform comparisons using challenging data collected from several indoor workspaces over the course of several weeks. Finally, we demonstrate the utility of the recovered 4D model in segmenting dynamic objects by simply clustering surface patches based on their temporal extent.

## 2. Related work

**RGB-D SLAM** Mapping from images has a long history in the robotics and computer vision literature with many recent developments motivated by emerging applications in 3D modeling, augmented reality and autonomous vehicles. Large-scale structure from motion (e.g., [1, 32, 9]) when combined with mult-view stereo (e.g., [10]) can yield rich geometric models but dense correspondence is often difficult to establish from monocular imagery, particularly for untextured surfaces common in indoor scenes.

The availability of cheap RGB-D sensors has enabled rapid progress indoor mapping, where active stereo or ToF provides very dense 3D structure and allows correspondence and pose estimation to be carried out by rigidly aligning scene fragments, e.g., using iterative closest point (ICP), rather than sparse matching and projective structure estimation techniques used in monocular SLAM. Initial work by Henry et al. [15] demonstrated the value of RGB-D data in ICP-based pose estimation while the KinectFusion system of Richard *et al.* [29] demonstrated impressive online reconstruction.

[1]We assume the temporal coordinate of each measurement is given while a full generalization of SLAM would also estimate the 6+1 DOF camera space-time pose

More recent work, such as ElasticFusion [38], has focused on improving performance of online real-time reconstruction and odometry by active updating and loop closure. To improve accuracy and robustness of offline reconstructions, Choi *et al.* used stronger priors on reconstructed geometry while carrying out global pose-graph optimization [6]. Recognition of familiar objects has also been integrated with SLAM-based approaches using prior knowledge of 3D structure [33] and fusing 2D semantic understanding with 3D reconstruction [16, 37]

**3D Registration** A core component of contemporary SLAM approaches based on LiDAR or RGB-D sensors is estimating alignments between pointclouds from successive measurements. A traditional starting point is iterative closet point (ICP) [4] which refines a rigid alignment minimizing mean-square inter-point or point-to-surface distance. However, the RGB-D fragment alignment problem differs somewhat from the classic problem of aligning range scans (e.g., [2]) due to the narrow field of view which often lacks distinguishing geometric features.

Our approach is based on a family of methods that model geometry in terms of probability densities (rather than points, meshes or signed distance functions). Horaud *et al.* introduced expectation conditional maximization (ECM) method for rigid point registration [17]. This formulation is appealing as it avoids explicit point correspondences and naturally generalizes to multi-way registration [8] and non-rigid deformation models [27, 12]. Our model builds on the work of Evangelidis *et al.*, which uses an ECM-based formulation to align multiple point sets to a single underlying consensus model [8]. We augment this density model with a temporal dimension, occlusion reasoning, and a richer parameterization of local mixture components.

**Dynamic Scenes and 4D maps** Traditional SfM has primarily focused on recovering structure of a single rigid scene modeled by sparse keypoints. For dynamic scenes where correspondence is available in the form of extended keypoint tracks, multi-body SfM provides an approach to grouping tracks into subsets, each of which moves rigidly (e.g., [7, 40]) while non-rigid SfM (e.g., [5]) addresses recovery of non-rigid surfaces from such tracks. When correspondence is not available but smooth surfaces are densely sampled in space-time, surface tracking approaches can be used to fuse observations (e.g., [26, 28]). Here we focus on scenarios where the temporal sampling is too sparse to allow for effective surface or feature tracking.

Related work on the problem of geometry change detection from sparse imagery was investigated by [34] and [36], who detect geometric changes relative to an existing model using voxel-based appearance consistency to drive model updates. Change detection with viewability and oc-

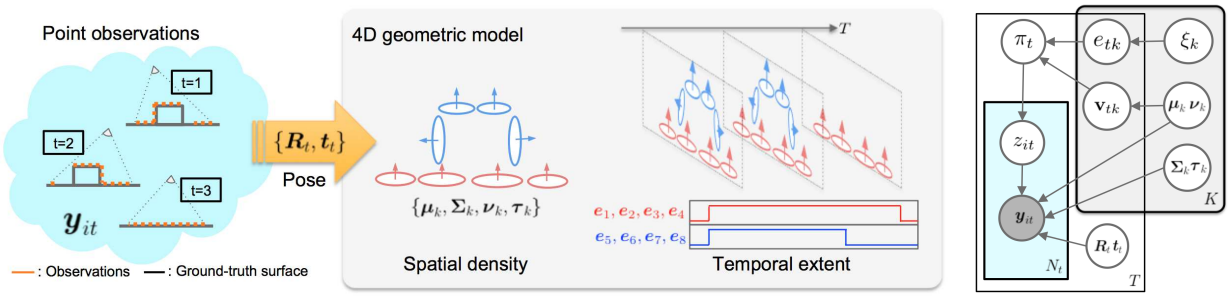



Figure 2. Observations are explained by a collection of surface patches that exist for some temporal extent and emit point observations into some unknown local coordinate system. We use a probabilistic mixture model (right) that represents parameters for $K$ patches producing $N_t$ point observations at each of $T$ time points. The prior probability of a patch emitting an observation at a given time $\boldsymbol{\pi}_t$ depends on whether the patch exists $\mathbf{e}_{tk}$ and is visible $\mathbf{v}_{tk}$, which in turn depends on the space-time geometry of the scene $(\boldsymbol{\xi}_k, \boldsymbol{\mu}_k, \boldsymbol{\nu}_k)$.

clusion was also explored by [11] for aligning observations to a construction job site plan. The work of [24] focuses on modeling dynamic appearance by grouping scene feature points into rectangular planes with an estimated temporal extent that captures, e.g., changing images on billboards. Finally, [23] used SfM to estimate geometry and warp images into a common viewpoint, enabling synthesis of time-lapse videos from unstructured photo collections. Closest in spirit to our approach is the work of Schindler and Dellaert [31] on "4D Cities", which utilizes bottom-up heuristics for grouping point observations from an SfM pipeline into building hypotheses and a probabilistic temporal model to infer the time interval during which buildings exist.

## 3. Space-time model fitting

We now describe our model for 4D geometry as a collection of surface patches with specified location, spatial and temporal extents. The model is fit to 3D point observations using a generative probabilistic approach inspired by the joint registration methods of Horaud *et al.* [17] and Georgios *et al.* [8].

### 3.1. Notation and Model Formulation

**Surface patches** We model the scene as a collection of $K$ surface patches. Each patch has a mean parameter $(\boldsymbol{\mu}_k, \boldsymbol{\nu}_k) \in \mathbb{R}^3 \times \mathbb{S}^2$ that describes the location and orientation. The spatial extent and roughness of the surface patch is described by corresponding variance parameters $(\boldsymbol{\Sigma}_k, \tau_k)$. Additionally, each patch $k$ has a specific temporal extent during which it exists in the scene specified by a time interval $[a_k, b_k]$. We denote the collection of shape parameters by $\mathcal{X} = \{(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \boldsymbol{\nu}_1, \tau_1), \ldots\}$ and temporal parameters by $\xi = \{(a_1, b_1), (a_2, b_2), \ldots\}$. We use the binary vector $\mathbf{e}_{tk} \in \{0, 1\}$ to indicate if patch $k$ *exists* at time $t$ so that $\mathbf{e}_{tk} = 1$ iff $t \in [a_k, b_k]$.

**Observations** The input data stream consists of observations of scene structure $\mathcal{Y} = \{\mathbf{y}_1 \ldots \mathbf{y}_T\}$ at $T$ discrete times. The observation at time $t$ consists of $N_t$ points with surface normals $\mathbf{y}_t = \{(\boldsymbol{l}_{it}, \boldsymbol{n}_{it}) \in \mathbb{R}^3 \times \mathbb{S}^2\}_{1 \le i \le N_t}$ where $\boldsymbol{l}_{it}$ and $\boldsymbol{n}_{it}$ to denote the location and surface normal associated with observation $\boldsymbol{y}_{it}$. In our experiments this data comes from a scan acquired by an RGB-D sensor but could come from other sources (e.g., ToF laser scanner or SfM reconstruction).

**Pose and Occlusion** Individual observations are assumed to be metric but are recorded in an arbitrary local coordinate system specified by unknown pose parameters $\Theta = \{\mathbf{R}_t, \mathbf{t}_t\}_{1 \le t \le T}$ which vary across time. We estimate a rigid transformation $\phi_t$ mapping each observation into a single global coordinate system. $\phi_t(\boldsymbol{l}_{it}, \boldsymbol{n}_{it}) = (\mathbf{R}_t \boldsymbol{l}_{it} + \mathbf{t}_t, \mathbf{R}_t \boldsymbol{n}_{it})$. A patch $k$ may not be visible at time $t$ due to the sensor placement relative to the scene structure. We use the variable $\mathbf{v}_{tk} \in \{0, 1\}$ indicate whether patch $k$ is visible at time $t$ which depends on the 4D model $\mathcal{X}$ and sensor parameters.

**Generating Observations from Patches** To generate observed data at time $t$ from scene $(\mathcal{X}, \xi)$ with a specified camera placement, we select a surface patch $k$ at random from those patches present at time $t$. If the patch is visible from the sensor position, then we sample a point location and normal from the patch density with parameters $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$. To allow for noise in the observations, we also include a background noise component whose distribution is uniform over the volume of a bounding box enclosing the scene model.

For a given time $t$, the probability of generating an observation $\boldsymbol{y}$ in local coordinates is modeled as a probabilistic mixture:

$$P(\boldsymbol{y}_t | \mathcal{X}, \Theta, \boldsymbol{\xi}) = \sum_{k=0}^{K} P(\phi_t(\boldsymbol{y}_t) | \mathcal{X}_k) P(k | \mathbf{e}_t, \mathbf{v}_t) \qquad (1)$$

The probability of generating an observation from patch $k$ is given by:

$$\pi_t(k) = P(k | \mathbf{e}_t, \mathbf{v}_t) = \begin{cases} \frac{1}{Z_t} p_0 & k = 0 \\ \frac{1}{Z_t} p_k e_{tk} \mathbf{v}_{tk} & k \in \{1 \ldots K\} \end{cases}$$

where $p_k$ is the time-independent intensity with which a patch $k$ generates observations, $p_0$ is the background noise intensity, and $Z_t$ is a normalizing constant.

Observations associated with a given patch are transformed into the global coordinate frame with location modeled by a Gaussian density and unit surface normal modeled by a von Mises Fisher (vMF) density

$$\begin{aligned} P(\phi_t(\boldsymbol{l},\boldsymbol{n})|\mathcal{X}_k) &= P(\phi_t(\boldsymbol{l})|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)P(\phi_t(\boldsymbol{n})|\boldsymbol{\nu}_k,\tau_k) \\ P(\phi_t(\boldsymbol{l})|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) &= \frac{1}{Z(\boldsymbol{\Sigma}_k)}\exp(-\frac{1}{2}||(\boldsymbol{R}_t\boldsymbol{l}+\boldsymbol{t}_t)-\boldsymbol{\mu}_k||^2_{\boldsymbol{\Sigma}_k}) \\ P(\phi_t(\boldsymbol{n})|\boldsymbol{\nu}_k,\tau_k) &= \frac{1}{Z(\tau_t)}\exp(\tau_k\boldsymbol{n}^T\boldsymbol{R}_t^T\boldsymbol{\nu}_k) \end{aligned}$$

Background noise points ($k=0$) are drawn from a uniform distribution over the observation volume $\boldsymbol{y}\sim\mathcal{U}(V\times\mathbb{S}^2)$.

### 3.2. Computing Visibility

To assign observations to surface patches, we need to compute visibility variables $\mathbf{v}_{tk}$ that indicate if patch $k$ is visible at time $t$. In order for a patch to be observed it must fall within the field of view of the sensor and must not be occluded by any other surface that existed at the observation time. For each time, we have one or more sets of camera parameters associated with gathering observations $\mathbf{y}_t$ which we write as $\{\mathbf{C}_{tu}\}_{1\le u\le F_t}$ where $F_t$ is the number of RGB-D frames used to build the observation.

Let $\phi_t^{-1}$ be the transformation specified by parameters $\Theta_t$ that transforms the global model $\mathcal{X}$ into the local coordinate frame used by observations at time $t$. We define the indicator function $FOV(\phi_t^{-1}(\boldsymbol{\mu}_k,\boldsymbol{\nu}_k),\mathbf{C}_{tu})$ to be 1 if patch $k$ was in the field of view of camera $\mathbf{C}_{tu}$.

To estimate occlusion, we use the hidden point removal algorithm introduced in [30] applied to the union of the observed points $\mathbf{y}_t$ and the set of transformed patch locations $\phi_t^{-1}(\mathcal{X})$ which are in the field of view. Let $OCL(\phi_t^{-1}(\boldsymbol{\mu}_k),\mathbf{y}_t,\mathbf{C}_{tu})$ be 1 if $\phi_t^{-1}(\boldsymbol{\mu}_k)$ is occluded by some part of $\mathbf{y}_t$ from camera viewpoint $\mathbf{C}_{tu}$ and 0 otherwise.

Combining these two components, we estimate that a patch $k$ should be visible at time $t$ if it is within the field of view and unoccluded in at least one camera view used to construct observation $\mathbf{y}_t$.

$$\mathbf{v}_{tk}=\begin{cases}1 & \exists u:[FOV(\phi_t^{-1}(\boldsymbol{\mu}_k,\boldsymbol{\nu}_k),\mathbf{C}_{tu})=1]\wedge \\ & \quad [OCL(\phi_t^{-1}(\boldsymbol{\mu}_k),\mathbf{y}_t,\mathbf{C}_{tu})=0] \\ 0 & \text{otherwise}\end{cases}$$

### 3.3. Model Parameter Estimation

To fit the model, we maximize the log-likelihood of observing $\mathcal{Y}$ given transformation parameters $\Theta$ and space-time geometry $\{\mathcal{X},\xi\}$ assuming independent point observations:

$$\max_{\mathcal{X},\Theta,\boldsymbol{\xi}}\log\prod_{it}P(\boldsymbol{y}_{it}|\mathcal{X},\Theta,\boldsymbol{\xi})P(\mathcal{X},\Theta,\boldsymbol{\xi})$$

We assume an uninformative priors on $\mathcal{X}$ and $\Theta$ and a prior on $\boldsymbol{\xi}$ that favors longer intervals (see below for details).

Let $z_{it}$ be a latent variable that denotes mixture assignments with $z_{it}=k$ when $\boldsymbol{y}_{it}$ is a point from patch $k$. We use expectation conditional maximization (ECM), which alternates between estimating expectations of $\mathcal{Z}$ and conditionally optimizing subsets of model parameters [25]. For a fixed setting of model parameters, ($\mathcal{X}$,$\Theta$,$\boldsymbol{\xi}$), the *E-step* estimates the probability that each observation $y_{it}$ came from surface patch $k$.

$$\alpha_{itk} \quad = \quad P(z_{it}=k\mid\mathcal{X},\Theta,\boldsymbol{\xi})$$

During the *M-step* we maximize the expected likelihood of subsets of parameters sequentially conditioned on the other parameters. Letting $s$ denote the iteration, we first update the alignment parameters, followed by the scene geometry and finally the temporal extent.

$$\begin{aligned} \Theta^{s+1} &= \arg\max_{\Theta}E_{\mathcal{Z}}\big[\log\big(P(\mathcal{Y},\mathcal{Z};\mathcal{X}^s,\Theta,\boldsymbol{\xi}^s)\big)\big] \\ \mathcal{X}^{s+1} &= \arg\max_{\mathcal{X}}E_{\mathcal{Z}}\big[\log\big(P(\mathcal{Y},\mathcal{Z};\mathcal{X},\Theta^{s+1},\boldsymbol{\xi}^s)\big)\big] \\ \boldsymbol{\xi}^{s+1} &= \arg\max_{\mathcal{V}}E_{\mathcal{Z}}\big[\log\big(P(\mathcal{Y},\mathcal{Z};\mathcal{X}^{s+1},\Theta^{s+1},\boldsymbol{\xi})\big)\big] \end{aligned}$$

We provide details for each parameter update below.

**Patch Assignment** Given the aligning transformation $\phi_t(\cdot)$ for time $t$ along with geometric, existence and visibility terms for $K$ patches, we compute the posterior probability that an observation is generated by a particular patch as:

$$\alpha_{itk} \quad = \quad \frac{P(\phi_t(\boldsymbol{y}_{ti})|\mathcal{X}_k)\pi_t(k)}{\sum_{j=1}^{K}P(\phi_t(\boldsymbol{y}_{ti})|\mathcal{X}_j)\pi_t(k)+\beta}$$

where $\pi_t(k)$ is the mixing weight of a patch $k$ at time $t$, $\beta$ is the weight of the background/outlier cluster which depends on $p_0$, and we set $\alpha_{it0}\propto\beta$ (see [17] for details).

**Alignment Parameters** Given the cluster assignment expectations for each observation, we would like to update the estimated transformation parameters $\mathbf{R}_t$ and $\mathbf{t}_t$ for $t$-th dataset. This amounts to a weighted least-squares problem with orthogonality constraint on $\mathbf{R}_t$. Following [17], we simplify this expression by first constructing a single "virtual point" $\mathbf{u}_{tk}$ per mixture component that integrates the interaction of all observed points with the patch.

$$\mathbf{u}_{tk}=\mathbf{w}_{tk}\sum_{i=1}^{N_t}\alpha_{itk}\boldsymbol{l}_{ti}\quad \mathbf{w}_{tk}=\left(\sum_{i=1}^{N_t}\alpha_{itk}\right)^{-1}$$

The optimal transformation can then be expressed as a weighted rigid alignment problem:

$$\arg\min_{\mathbf{R},\mathbf{t}} \frac{1}{2} \sum_{k=1}^{\mathbf{K}} \mathbf{w}_{tk} ||\mathbf{R}\mathbf{u}_k + \mathbf{t} - \boldsymbol{\mu}_k||^2_{\boldsymbol{\Sigma}_k}$$

When $\boldsymbol{\Sigma}_k$ is isotropic this can be solved efficiently using SVD (e.g., [18]). This provides a good initialization that can be further refined by projected gradient for anisotropic case and the additional linear term from the density over surface normals.

**Spatial Patch Parameters** Given the transformation parameters, we update mean and covariance for each Gaussian mixture component.

$$\boldsymbol{\mu}_k = \frac{\sum_{t=1}^{T}\sum_{i=1}^{N_t} \alpha_{itk}\phi_t(\boldsymbol{l}_{it})}{\sum_{t=1}^{T}\sum_{i=1}^{N_t} \alpha_{itk}}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{t=1}^{T}\sum_{i=1}^{N_t} \alpha_{itk}\big(\phi_t(\boldsymbol{l}_{it}) - \boldsymbol{\mu}_k\big)\big(\phi_t(\boldsymbol{l}_{it}) - \boldsymbol{\mu}_k\big)^\intercal}{\sum_{t=1}^{T}\sum_{i=1}^{N_t} \alpha_{itk}} + \epsilon^v \mathbf{I}$$

The variable $\epsilon^v$ is used to prevent the variance of a given cluster from collapsing ($\epsilon^v = 0.1^6$ in our experiments). The updates for vMF mean and concentration parameters for the surface normal follow a similar form [13]. In practice, we found that constraining the covariance to be isotropic works well for large K and yields more efficient optimization. Patch intensity priors $p_k$ are estimated as the assignment proportion $\sum_{it} \alpha_{itk}$ scaled by the proportion of observations in which the patch was visible and existed.

**Temporal Patch Parameters** To reliably estimate when each patch is present, we must make some stronger assumptions about the prior distribution over $\mathbf{e}_k$. We thus assume that a given patch exists for a single temporal interval $[a_k, b_k]$ during which the probability of the patch emitting an observation is uniform.

$$P(e_{tk} = 1) = \begin{cases} \gamma_{ab} & t \in [a_k, b_k] \\ \epsilon & \text{otherwise} \end{cases}$$

where $\epsilon$ is a small constant and $\gamma$ is chosen so the distribution integrates to 1 over the total observation interval $T$.

$$\gamma_{ab} = \frac{1 - \epsilon(T - (b - a))}{(b - a)},$$

To estimate $a_k, b_k$ we maximize the expected posterior probability over times where the cluster was visible:

$$[a_k, b_k] = \arg\max_{a,b} \sum_{t \in [a,b]} \mathbf{v}_{tk} \left[\sum_{i=1}^{N_t} \alpha_{itk} \log(\gamma_{ab})\right] + \sum_{t \notin [a,b]} \mathbf{v}_{tk} \left[\sum_{i}^{N_t} \alpha_{itk} \log(\epsilon)\right] + \log P^s(a, b)$$

where $\log P^s(a, b) = avg(\alpha_{..k}) \log(b - a + \epsilon^p)$ is a prior that encourages existence of patches for longer time spans. The prior is scaled using average value of $\alpha_{..k}$. In our experiments we use $0.05$ for $\epsilon$ and $0.01$ for $\epsilon^p$.

**Extension to Non-parametric Mixtures** In addition to standard mixture model fitting with fixed number of clusters $K$, we also considered a variant of our model using a Dirichlet Process (DP) prior over the cluster allocations [35, 22, 21]. This is appealing since it allows the model to naturally grow in complexity as more observations become available. We use collapsed Gibbs sampling to explore the space of the number of mixture components $K$ and weights $\pi$. Rather than performing full Bayesian inference, we interleaved rounds of sampling with conditional maximization to optimize alignment parameters. We observed empirically that starting from an initial state with few mixture components and refining the alignment while non-parametrically growing the number of components often resulted in better registration results (see experiments). We presume this may be because the early energy landscape with few mixture components has fewer local minima.

## 4. Space-Time Datasets

While there are a large number of published RGB-D and 3D scene datasets (e.g., [39]), previous work has focused on static scenes described at a single point in time (or collected at high-frame rate over a short interval). To validate our approach with more compelling temporally varying elements, we developed datasets based on both synthetic scenes with simulated sensors and real scans of human workspaces

**Synthetic Data** Synthetic data is easy to generate and provides perfect ground-truth which is useful for evaluating reconstruction accuracy. To emulate the noise characteristics of real scanning, we start from a 3D model and simulate acquisition of RGB-D data from a moving sensor and pass it through a standard SLAM pipeline to produce a 3D point cloud which constitutes observation at a single time point.

We use 3D room models provided by [14] and populate them with IKEA furniture models. Each item of furniture is present for randomly specified interval of time. We generate a virtual sensor trajectory by selecting several key points manually and synthesize a smooth path connecting the key points. Given a trajectory we render a sequence of RGB-D frames which are then fed into the ElasticFusion [38] pipeline to produce a simulated observed point-cloud. Back-projecting keypoints from the observation provides a ground-truth alignment with the world coordinate system and mapping between the simulated observed points and the object ids in the scene model. We generate 8 time point per scene producing scans with a million points (summarized in Table 1).

(a) Observations (b) Aligned observations from the same time point (c) Observations from all time points

Figure 3. 3D model pieces with 4 different types (a) for "Laboratory" dataset. (b) and (c) shows merged model with one time or space.

**Synthetic Data**

| Name | # regions | # times | # frames | # 3D points |
|---|---|---|---|---|
| Bedroom | 1 | 8 | 3.2k | 1.2M |
| Bathroom | 1 | 8 | 4k | 1.5M |

**Real Data**

| Name | # regions | # times | # frames | # 3D points |
|---|---|---|---|---|
| Laboratory | 4 | 8 | 3k | 1M |
| Copier room | 1 | 7 | 1.5k | 0.6M |
| Kitchen | 1 | 5 | 1.5k | 0.6M |

Table 1. Summary statistics of test datasets.

**Real Data** We also collected scans of 3 different indoor scenes (Laboratory, Kitchen, and Copier room) once a day over several weeks using a Kinect sensor. We chose these scenes since they contained a number of objects that naturally move from one day to the next by people passing through the room. To provide the best quality and consistency in scanning, we used a custom motorized tripod fixture to automated the scan path. For our "Laboratory" dataset, we collected 4 scans per time point in order to cover different overlapping parts of a larger room (see Figure 3). Each scan is processed individually using ElasticFusion [38] to produce a point cloud. Dataset statistics are summarized in Table 1.

To establish a high quality ground-truth alignment, we exploit the presence of the floor which is visible in all our recovered scans. We first segment the floor based on color and surface normal from each scan. We then constrain the search over alignments to only consider translations in the plane of the floor and rotations around the $z$ axis perpundicular to the floor. This pre-process greatly reduces the number local minima and, guided by a few hand-clicked correspondences, is sufficient for finding high quality alignments which can be further refined to improve accuracy.

Once the scans are aligned into a common coordinate frame, we segment and annotate the points in each scan with object instance labels such as "floor", "desk", "chair", etc. These instance labels are shared across time points, allowing us to identify observations at different times which correspond to the same underlying surface and provide a basis for benchmarking the ability of our model to correctly identify spatio-temporal extents (see Figure 6).

## 5. Experimental Evaluation

Figure 3 shows qualitative results of running our joint registration and reconstruction model on the Laboratory dataset depicting (a) individual scans, (b) reconstruction of a single time point consisting of multiple overlapping scans, and (c) all time point reconstructions superimposed in a single global coordinate system. Colors (b) and (c) indicate points belonging to different scans.

We quantitatively evaluate the method in terms of the accuracy of reconstruction and localization (alignment). In particular, we show that the existence and visibility terms are valuable even when aligning partially overlapping scans for static scenes. We then evaluate the accuracy of estimated existence time intervals. Lastly, we demonstrate the utility of this 4D representation in segmenting out dynamic objects in a scene.

**Spatial reconstruction accuracy** To evaluate metric reconstruction quality, we compare our method to two baselines which don't model temporal change. First, we consider running the ElasticFusion pipeline applied to data concatenated from all time points (**EF3D**). Second, we consider running ElasticFusion independently at each time point and subsequently align reconstructions from different times using the method of Evangelidis *et al.* [8] (**EF4D**). EF3D produces a single 3D reconstruction which is compared to all time points while EF4D and our model produce 4D reconstructions which change over time. We evaluate precision and recall (of those model points that were visible from the sensor) w.r.t. ground-truth surface across all time points for the synthetic 4D benchmark using a distance threshold of 1cm.

As Table 2 shows, the precision of a single 3D reconstruction (EF3D) is lower than the 4D models (EF4D,ours). Our model further improves precision over simply aligning individual time points by providing more robustness to dynamic objects. Our method also shows a substantial boost in recall over both baselines due to the ability of the model to fill in occluded regions with observations from a different time.

**Temporal reconstruction accuracy** We use the synthetic dataset for which the ground-truth duration of existence of each object is known and evaluate the accuracy ( Table 3). Since our model predicts existence of patches, we establish a correspondence, assigning points of each ground-truth object to a mixture component in our estimated model. This assignment imposes an upper-bound on the accuracy (i.e., since a mixture component may span two different objects whose temporal extent differs). We find that most incorrect

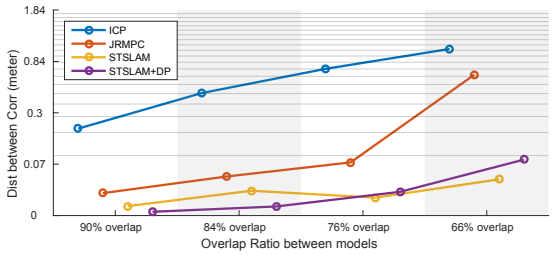



Figure 4. Registration accuracy of our method (ST-SLAM) compared to (ICP) [4] and joint registration (JRMPC) [8] measured by average closest point distance. Error axis is on a logarithmic scale. Explicitly inferring visibility improves robustness to partial overlap, as does using non-parametricly growing the number of mixture components during optimization (ST-SLAM+DP). Error in estimated rigid transformation parameters behaves similarly (see supplement).

predictions come from near such edges.

**Density visualization** Since our reconstruction is generative, we can also visualize it as a spatio-temporal probability density. In Figure 5, we display the estimated density of observations marginalized over all time as well as conditioned on specific time points alongside the corresponding scene. To aid visualization, we exclude the background scene component and rescale the colormap. As the figure shows, the estimated space-time density tracks the arrangement of furniture within the room.

**Robustness on partially overlapping observations** Previous joint registration methods [4, 19, 8] that align multiple pointclouds into a single consensus model often rely on a high degree of overlap between scans in order to achieve correct alignment. Since our approach explicitly models which surface patches are visible in a given scan, it can handle larger non-overlapping regions by allocating additional mixture components and explaining away the lack of data generated from those components in scans where they were not visible.

To demonstrate the value of estimating visibility, we carry out an experiment on the ground-truth Laboratory reconstruction by splitting a single time point into two pieces with controlled degree of overlap, ranging from 50% to 90%, apply a random rigid transformation and add Gaussian noise to point locations. We measure difference between the known transformation and the estimate, as well as mean distance between corresponding points, averaged over 10 trials.

As Figure 4 displays, all methods have higher registration error as the degree of overlap decreases. Since ICP [4] and the joint registration method JRMPC [8] do not infer which consensus points/mixtures are visible in a given observation, their performance degrades more rapidly as overlap decreases. Our model with a fixed number of mixtures (ST-SLAM) is more robust and using the DP mixture allocation (ST-SLAM+DP) yields more robust results, presumably due to annealing effects of starting with a small number of mixture components.

| | EF3D | EF4D | Our method (STSLAM) |
|---|---|---|---|
| Precision | 71.3% | 92.7% | 93.3% |
| Recall | 90.4% | 90.0% | 98.5% |

Table 2. Evaluation for reconstruction quality. Baselines EF3D merges all observations in a single 3D model; EF4D builds a separate model for each time point and then registers them. Our model of temporal extent and visibility improves both precision and recall. See Section 5 for details.

| | Baseline | STSLAM | STSLAM-DP | Upper-bound |
|---|---|---|---|---|
| Bedroom | 83.2% | 90.8% | 90.9% | 95.5% |
| Bedroom (NS) | 41.2% | 74.3% | 74.8% | 95.4% |
| Bathroom | 69.7% | 78.0% | 77.7% | 95.0% |
| Bathroom (NS) | 31.8% | 62.2% | 62.7% | 99.9% |

Table 3. Temporal reconstruction accuracy. Baseline assumes every object exists for all time points. Non-static (NS) evaluation excludes the static background component. Upper-bound indicates maximal achievable accuracy given the spatial quantization imposed by cluster assignment.

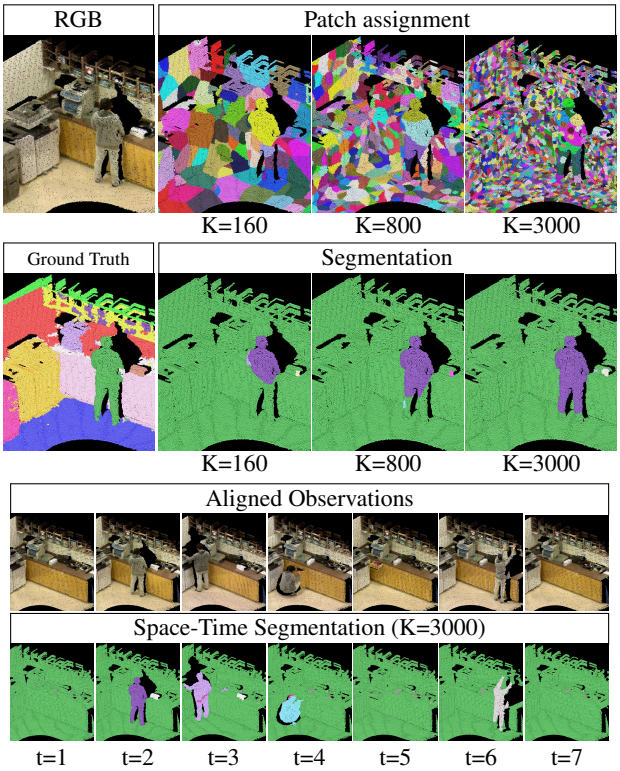
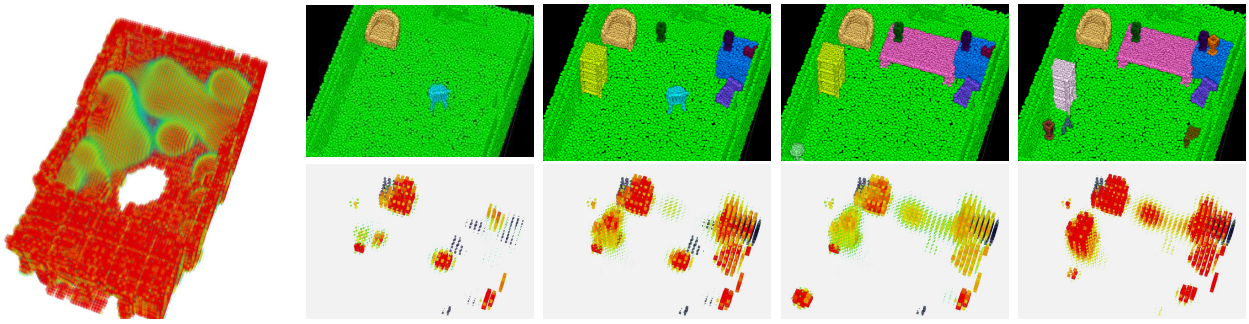


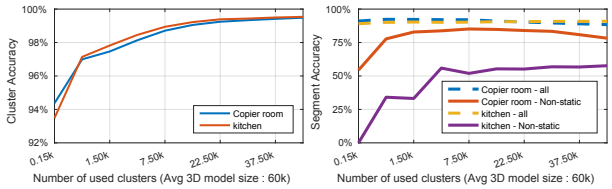

Figure 6. Segmentation results on the "Copier room" dataset showing grouping of surface patches with similar temporal extent. Segmentation accuracy depends on the number of surface patches (top). Segmentation across all space-time observations using the optimal cluster size discovers static and dynamic scene components (bottom).

**Segmentation of dynamic objects** The space-time geometry model provides a natural basis for performing ad-



(a) Marginal density $P(\mathbf{y})$ (b) $P(\mathbf{y}|t=1)$ (c) $P(\mathbf{y}|t=2)$ (d) $P(\mathbf{y}|t=3)$ (d) $P(\mathbf{y}|t=4)$

Figure 5. Visualization of 3D probability density predicted by the fit model. (a) shows density marginalized over time (b) - (e) display probability density conditioned on different observation times with the static component excluded for clarity.




(Under-segmentation accuracy) (Segmentation accuracy)

Figure 7. We measure the accuracy with which individual patches constitute an under-segmentation of the objects in the scene (left) and how well grouping patches by temporal extent recovers object segments (right) on real datasets "Copier room" and "Kitchen". Since the scene is dominated by static structure, we also separately plot the segmentation accuracy for the non-static components.

ditional inferences about a scene. In particular, the temporal coherence of a set of surface patches provides a strong indicator that those patches belong to the same surface. In Figure 6 we visualize segmentations into objects based on grouping those patches with a similar estimated temporal extent. Raw measurements are assigned the segment label for that cluster with the highest assignment probability ($\alpha_{itk}$). As shown in the upper figure panel, the segmentation accuracy is limited based on the size/number of surface patches fit to the scene. To produce good quality assignments we choose a number of mixture components that yields patches of an average small physical dimension (relative to the resolution of the raw point observations). The lower panel of Figure 6 shows such a segmentation with the static background component in green.

We consider two quantitative measures of segmentation accuracy. Let $U$, $V$, $S$ denote the surface patches, predicted segments, and ground-truth segments respectively. We characterize the degree of under-segmentation (i.e., how often a surface patch spans an object boundary) by the average percentage of a patch that is completely contained in some ground-truth segment.

$$Score_{useg} = \frac{1}{N_c}\sum_{i=1}^{N_c}\max_j \frac{|U_i \cap S_j|}{|S_j|}.$$

To measure the effectiveness of grouping patches by temporal extent, we compute the IoU of predicted and ground-truth segments.

$$Score_{seg} = \frac{1}{N_s}\sum_{i=1}^{N_s}\max_j \frac{|V_i \cap S_j|}{|V_i \cup S_j|}$$

In Figure 7 we plot these scores as a function of the number of mixture components. As might be expected, the under-segmentation error decreases rapidly as the number of clusters grow, allowing smaller patches that are less likely to span an object boundary. However, there is a tradeoff in segmentation accuracy of dynamic objects as the number of clusters goes beyond a certain point as the estimates of temporal extent become increasingly noisy with few observations per cluster.

## 6. Conclusion

We present a novel probabilistic formulation of space-time localization and mapping from RGB-D data streams which jointly estimates sensor pose and builds an explicit 4D map. We validated this approach on real and synthetic data, showing improved reconstruction and registration for dynamic scenes and demonstrate unique features of the model which allow estimation of the temporal extent of surface patches and segmentation into temporally coherent objects. In the future we hope to extend this approach to handle more dynamic and larger-scale scenes, replace our wide-FoV observations with individual RGB-D frames, and tackle the problem of inter-frame tracking of moving objects.

**Acknowledgements:** This work was supported by NSF grants IIS-1618806, IIS-1253538, and a hardware donation from NVIDIA.

## References


[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski. Building Rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] J. Aggarwal, B. Vemuri, Y. Chen, and G. Medioni. Range image understanding object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145 – 155, 1992.

[3] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large-scale indoor spaces. In *IEEE CVPR*, pages 1534–1543, 2016.

[4] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE TPAMI*, 14(2):239–256, Feb. 1992.

[5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *IEEE CVPR*, volume 2, pages 690–696, 2000.

[6] S. Choi, Q.-Y. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *IEEE CVPR*, 2015.

[7] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3):159–179, 1998.

[8] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis. A generative model for the joint registration of multiple point sets. In *ECCV*, 2014.

[9] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. Building Rome on a cloudless day. In *ECCV*, pages 368–381, 2010.

[10] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE TPAMI*, 2010.

[11] M. Golparvar-Fard, F. Pena-Mora, and S. Savarese. Monitoring changes of 3d building elements from unordered photo collections. In *IEEE ICCV Workshops*, pages 249–256, 2011.

[12] V. Golyanik, B. Taetz, G. Reis, and D. Stricker. Extended coherent point drift algorithm with correspondence priors and optimal subsampling. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9, 2016.

[13] S. Gopal and Y. Yang. Von Mises-Fisher clustering models. In *ICML*, pages 154–162, 2014.

[14] A. Handa, V. Patraucean, S. Stent, and R. Cipolla. Scenenet: An annotated model generator for indoor scene understanding. In *IEEE International Conference on Robotics and Automation, (ICRA)*, 2016.

[15] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *the 12th International Symposium on Experimental Robotics (ISER)*, 2010.

[16] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from RGB-D images. In *IEEE International Conference on Robotics and Automation, (ICRA)*, pages 2631–2638, 2014.

[17] R. Horaud, M. Yguel, G. Dewaele, F. Forbes, and J. Zhang. Rigid and articulated point registration with expectation conditional maximization. *IEEE TPAMI*, 33:587–602, 2010.

[18] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, 1987.

[19] B. Jian and B. Vemuri. Robust point set registration using Gaussian mixture models. *IEEE TPAMI*, 33(8):1633–1645, 2011.

[20] B. Klingner, D. Martin, and J. Roseborough. Street view motion-from-structure-from-motion. In *IEEE ICCV*, pages 953–960, 2013.

[21] D. Lin. Online learning of nonparametric mixture models via sequential variational approximation. In *NIPS*, pages 395–403. 2013.

[22] D. Lin and J. W. Fisher. Coupling nonparametric mixtures via latent dirichlet processes. In *NIPS*, pages 55–63. 2012.

[23] R. Martin-Brualla, D. Gallup, and S. M. Seitz. 3d time-lapse reconstruction from internet photos. In *IEEE ICCV*, 2015.

[24] K. Matzen and N. Snavely. Scene chronology. In *ECCV*, pages 615–630, 2014.

[25] X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267, 1993.

[26] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. J. Guibas, and H. Pottmann. Dynamic geometry registration. In *Symposium on geometry processing*, pages 173–182, 2007.

[27] A. Myronenko, X. Song, and . Carreira-Perpin. Non-rigid point set registration: Coherent point drift (CPD). In *NIPS*, 2006.

[28] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE CVPR*, pages 343–352, 2015.

[29] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE ISMAR*, October 2011.

[30] K. Sagi, T. Ayellet, and B. Ronen. Direct visibility of point sets. In *ACM SIGGRAPH*, 2007.

[31] G. Schindler and F. Dellaert. Probabilistic temporal inference on reconstructed 3d scenes. In *IEEE CVPR*, pages 1410–1417, 2010.

[32] N. Snavely, I. Simon, M. Goesele, R. Szeliski, and S. M. Seitz. Scene reconstruction and visualization from community photo collections. *Proceedings of the IEEE*, 98(8):1370–1390, 2010.

[33] H. Strasdat, R. A. Newcombe, R. F. Salas-Moreno, P. H. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. *IEEE CVPR*, 00, 2013.

[34] A. Taneja, L. Ballan, and M. Pollefeys. Image based detection of geometric changes in urban environments. In *IEEE ICCV*, pages 2336–2343, 2011.

[35] Y. W. Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. 2011.

[36] A. O. Ulusoy and J. L. Mundy. Image-based 4-d reconstruction using 3-d change detection. In *ECCV*, pages 31–45, 2014.

[37] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kähler, D. W. Murray,

S. Izadi, P. Perez, and P. H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[38] T. Whelan, S. Leutenegger, R. S. Moreno, B. Glocker, and A. Davison. Elasticfusion: Dense SLAM without a pose graph. In *Proceedings of Robotics: Science and Systems*, 2015.

[39] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *IEEE ICCV*, 2013.

[40] J. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE TPAMI*, 30(5):865–877, 2008.