Handling Perceptual Clutter for Robot Vision with Partial Model-Based Interpretations

Grace Tsai and Benjamin Kuipers

Abstract— **For a robot to act in the world, it needs to build and maintain a simple and concise model of that world, from which it can derive safe opportunities for action and hazards to avoid. Unfortunately, the world itself is infinitely complex, containing aspects ("clutter") that are not well described, or even well approximated, by the simple model. An adequate explanatory model must therefore explicitly delineate the clutter that it does not attempt to explain. As the robot searches for the best model to explain its observations, it faces a three-way trade-off among the coverage of the model, the degree of accuracy with which the model explains the observations, and the simplicity of the model. We present a likelihood function that addresses this trade-off. We demonstrate and evaluate this likelihood function in the context of a mobile robot doing visual scene understanding. Our experimental results on a corpus of RGB-D videos of cluttered indoor environments demonstrate that this method is capable of creating a simple and concise planar model of the major structures (ground plane and walls) in the environment, while separating out for later analysis segments of clutter represented by 3D point clouds.**

I. INTRODUCTION

An indoor navigating robot must perceive its local environment in order to act. Visual perception has become popular because it captures a lot of information at low cost. When using vision as a sensor for a robot, the input to visual perception is a temporally continuous stream of images, not simply a single image or a collection of images. The output of visual scene understanding has to be a concise interpretation of the local environment that is useful for the robot to make plans. Moreover, visual processing must be an on-line and efficient process as oppose to a batch process so that the robot's interpretation can be updated as visual observations become available.

Many Visual SLAM methods [1], [11], [9] have been proposed to construct a 3D point cloud of the local environment in real-time. A more concise, large-granularity model that would be useful to a robot in planning must then be constructed from the point cloud. Methods [4], [2], [3] have been proposed to extract planar models from a 3D point cloud. However, due to the need for multiple iterations through the entire set of images, these methods are off-line and computationally intensive, making them inapplicable to real-time visual scene understanding for robots.

Our previous work is an efficient on-line method for scene understanding. We presented a concise planar representation, the Planar Semantic Model [**?**], to represent the geometric structure of the local indoor environment. We treat the scene understanding problem as a dynamic and incremental process because the view of the local environment changes while the robot travels within it. We presented an efficient on-line generate-and-test method with Bayesian filtering to construct the PSM from a stream of monocular images [14]. However, our previous method assumes the indoor environment is empty and the PSM is capable of modeling everything in the environment. If the environment is not empty, observations that are not part of the planar structures may mislead the Bayesian filter to converge to the wrong hypothesis.

The focus of this paper is to handle clutter in visual scene understanding. We define "clutter" as regions in the local environment that cannot be represented by the given model. Since clutter is unstructured by the given the model, clutter is represented by 3D point clouds, where each point cloud corresponds to a 2D segment in the image space. The interpretation of the local environment consists of the model and clutter. The model "explains" a subset of the visual observations, and clutter is the collection of observations that are not explained by the model.

There are existing works on indoor scene understanding that interpret the scene with a planar model and clutter. Hedau *et al.* [5] and Wang *et al.* [16] model clutter using a classifier that links the image features to clutter. Dependence on prior training for clutter is difficult to generalize to different indoor environments since the appearance of clutter is highly variable. Lee *et al.* [10] and Taylor *et al.* [12] treat clutter as observations that are not explainable by the planar model geometrically, which is our definition of clutter, and find a model that fits most of the observations. However, these methods may be difficult to apply to real-time applications because evaluations at pixel level or evaluation on a large set of hypotheses are involved. Moreover, since these methods handle only single image, temporally coherent interpretation of the scene may be difficult to achieve if each frame is independently processed.

This paper builds on top of the on-line generate-and-test framework [14] to incrementally construct an interpretation of the local environment with a PSM model and clutter. Specifically, we propose a likelihood function that allows a good PSM hypothesis to explain only a subset of the observed features. The likelihood function tests the hypothesis based on a three-way trade-off among coverage, accuracy, and simplicity of the model. Our experimental results on a variety of RGB-D videos on cluttered indoor environments demonstrate that our method is capable of interpreting the

Both authors are with the Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, U.S.A.

This work has taken place in the Intelligent Robotics Lab in the Computer Science and Engineering Division of the University of Michigan. Research of the Intelligent Robotics lab is supported in part by grants from the National Science Foundation (CPS-0931474, IIS-1111494, and IIS-1252987).

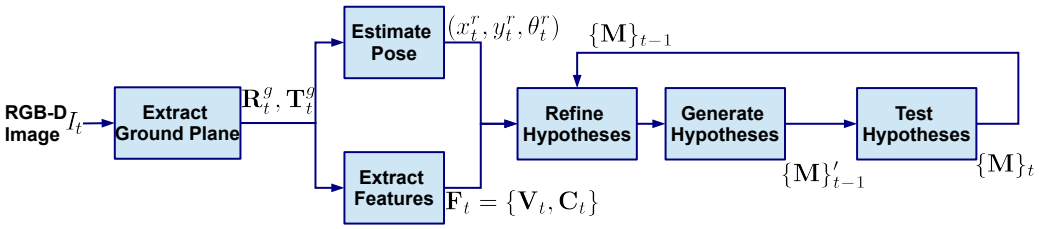


Fig. 1. Framework. Given the current RGB-D frame t, we first find the relation between the camera and the local 3D world coordinate, and then estimate the 3dof robot pose in the 3D world coordinate. At the same time, we extract useful features $\mathbf{F}_t$ (e.g. vertical-plane features $\mathbf{V}_t$ and point-set features $\mathbf{C}_t$) from frame t. We then follow the on-line generate-and-test framework for indoor scene understanding [14]. The main contribution of this paper is the proposed likelihood function for testing the hypotheses that allows a good hypothesis to explain only a subset of the features $\mathbf{F}_t$. In addition, we demonstrated a method to generate and refine the hypotheses using the vertical-plane features $\mathbf{V}_t$.

environment with a PSM model and separate out for later analysis segments of clutter represented by 3D point clouds.

Our work directly addressed a central dilemma in model-based scene interpretation. On one hand, it is worth describing much of the environment using a model that is as simple as possible, and identifying the parts of the environment that require more sophisticated models, instead of having one highly-complex model to describe the whole environment. On the other hand, when testing multiple model hypotheses, a hypothesis is preferred if it explains as many observations as possible and only leaves a small portion of the observations unexplained. Our proposed likelihood function makes explicit the trade-off between explanation coverage, accuracy, and simplicity. While this paper demonstrates a method to extract a geometric structure of the environment and separate out clutter, the same approach can be applied to further interpret clutter with object models.

II. METHOD

This paper demonstrates a method that incrementally and dynamically construct an interpretation of the local environment with a concise model (Section II-C) and clutter, parts of the environment that cannot be represented by the model, using a stream of RGB-D images. Figure 1 illustrates our framework. At each frame t, we first extract the ground plane G and compute a transformation $[R_t^g, T_t^g]$ that transforms the 3D points from the image coordinate to the local 3D coordinate (Section II-A). Points that lie on the ground plane are now considered as "explained" by the ground G. The transformation $[R_t^g, T_t^g]$ captures 3dof of the full 6dof camera pose, the pan and roll angles and the height of the camera with respect to the ground plane at frame t. We call the remaining 3dof the *robot pose*, which is represented as $(x_t^r, y_t^r, \theta_t^r)$ on the 3D world coordinate. The robot pose is estimated by aligning RGB image features and the 3D non-ground-plane points between frame $t-1$ and t (Section II-B).

At the same time, we extract a set of features $\mathbf{F}_t$ from the 3D points that are not explained by the ground G. While there are other research focusing on grouping a RGB-D image or a set of 3D points into primitive shapes [7], in this paper, we group the 3D points into a set of vertical-plane features $\mathbf{V}_t$ and a set of point-set features $\mathbf{C}_t$. Vertical-plane features suggest the existence of walls, while point-set features suggest the existence of clutter.

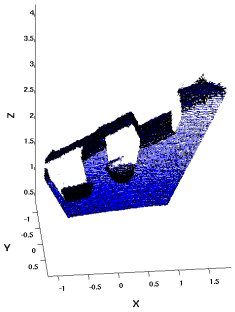
Given the robot pose and the features $\mathbf{F}_t = \{\mathbf{V}_t, \mathbf{C}_t\}$, the on-line generate-and-test framework first uses the vertical-plane features $\mathbf{V}_t$ to refine the precision of the existing set of PSM hypotheses $\{\mathbf{M}\}_{t-1}$ (Section II-E). Then, a set of new hypotheses are generated and added to the hypothesis set $\{\mathbf{M}\}'_{t-1}$ by transforming existing hypotheses into children hypotheses describing the same environment with more details based on the vertical-plane features $\mathbf{V}_t$. (Section II-F) Finally, we use a Bayesian filter to test the hypotheses. While our previous work requires a good hypothesis explains all the observed features, in this paper, we proposed a new likelihood function that allows a good hypothesis to explain only a subset of the features. (Section II-G) The output interpretation of the environment is the PSM hypothesis with the maximum posterior probability and clutter, 3D points that cannot be explained by that PSM hypothesis.

A. Extract Ground Plane

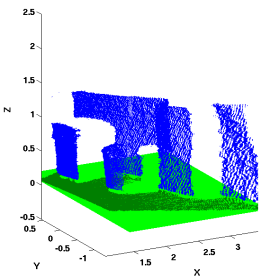
At each frame, we extract the 3D ground plane using both RGB image and depth information. Figure 2 illustrates how the ground-plane is extracted. First, we collect the pixels where their local surface normals differ from the approximated normal vector $\mathbf{N}_{appx}$ of the ground plane [1] within ϕ_{ground}. The local surface normal of each pixel is computed using the efficient algorithm proposed by Holz *et al.* [7]. From those pixels, we use RANSAC to fit the dominant plane that has a normal vector within ϕ_{ground} of the approximated normal $\mathbf{N}_{appx}$. From the inlier pixels, we perform a morphological close operation on the RGB image to locate a smooth and bounded region for the ground plane.

Once the ground plane is extracted, we compute the transformation $[\mathbf{R}_t^g, \mathbf{T}_t^g]$ between the camera coordinate and the local 3D coordinate, where the x-y plane is the ground-plane and the z-axis is pointing up. The origin of the local 3D coordinate is set to the projected location of the camera center on the ground plane. Mathematically, $\mathbf{R}_t^g$ is the rotation matrix that rotates the normal vector of the ground plane to $[0, 0, 1]$, and $\mathbf{T}_t^g = [0, 0, h_t]$ is determined by the distance h_t between the camera center and the ground-plane. In this paper, we also define the world coordinate to be the local 3D coordinate of the first frame. The transformation

[1]For a front-facing camera, $\mathbf{N}_{appx} = [0, 1, 0]$ in the image coordinate, and in our experiments, we set $\phi_{ground} = \frac{\pi}{6}$.



(a) Collect candidate ground points



(b) Fit plane using RANSAC

(c) Obtain inlier points

(d) Smooth ground-plane region

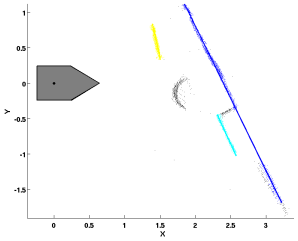
Fig. 2. Extract ground plane from a single RGB-D image. (Best viewed in color.) (a) From the input RGB-D image, collect candidate ground-plane points (blue) with local surface normal close to $[0, 1, 0]$ in the image coordinate. Notice that since the depth information is noisy, candidate points may not lie on the ground plane, and some ground-plane points may not be identified as candidate points. (b) (c) Use RANSAC to fit the dominant plane from the candidate points. Inlier points (pixels) are marked as green. From the ground-plane equation, we determine the transformation between the image coordinate and the local 3D coordinate. In (b), points are transformed to the local 3D coordinate. (d) Perform a morphological close operation on the image to obtain a smooth ground-plane region.

between the local 3D coordinate and the world coordinate can be inferred by the robot pose (Section II-B).

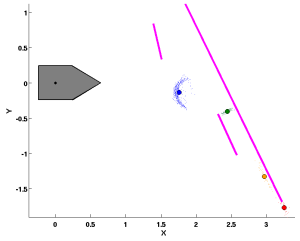
B. Estimate Pose

Given the ground-plane and the transformation $[\mathbf{R}_t^g, \mathbf{T}_t^g]$ from the camera coordinate to the local 3D coordinate, the *robot pose* $(x_t^r, y_t^r, \theta_t^r)$ captures the remaining three degrees of freedom of the camera pose in the 3D world coordinate. We start by estimating the pose change between frame t and $t-1$ by aligning sparse feature correspondences between the two RGB-D images. We extract Harris corner features in the RGB image at frame $t-1$, and obtain their corresponding image locations at frame t using KLT tracking. Then, we find a rigid-body transformation $[\mathbf{R}_t^p, \mathbf{T}_t^p]$ that aligns the 3D locations of the correspondences in the two frames. Note that since we only have three degrees of freedom, $\mathbf{R}_t^p$ is a rotation matrix along the z-axis and $\mathbf{T}_t^p$ is a translation vector on the x-y plane. With this initial estimate of the robot pose [2], we use Iterative Closest Point (ICP) algorithm to refine $[\mathbf{R}_t^p, \mathbf{T}_t^p]$ from all the 3D points that are not on the ground plane. Finally, the robot pose in the world coordinate can be computed from the pose change.

[2]If there are not enough sparse feature correspondences to compute the initial estimate, we assume the robot is moving at a constant motion and use the pose change between frame $t-1$ and $t-2$ as our initial estimate.



(a) Extract Vertical-Plane Features



(b) Extract Point-Set Features

Fig. 3. Extract features from the RGB-D frame shown in Figure 2. (Best viewed in color.) Once the ground-plane is extracted, we extract features from the points that are not from the ground plane. (a) We first extract vertical-plane features by fitting line segments to the points in the ground-plane map. Points are colored according to the vertical-plane features that they generated, and points that cannot be grouped into line segments are marked as black. (b) We extract point-set features by clustering the remaining points based on their Euclidean distances. In general, vertical-plane features suggests the existence of walls, and point-set features suggest the existence of clutter. However, some vertical-plane features may come from clutter region (e.g. the blue trash can), and some (red and orange) point-set features may be noise from a wall plane.

C. Planar Semantic Model

We use the Planar Semantic Model (PSM) proposed in our previous work [14] to represent the structure of an indoor environment. In this paper, we use the following notations to represent a PSM M:

$$\begin{aligned} M &= \{G, W_1, W_2, W_3, ..., W_n\} \\ W_i &= \langle \alpha_i, d_i, S_1^i, S_2^i, ..., S_{m_i}^i \rangle \\ S_j^i &= \langle x_{1,j}^i, y_{1,j}^i, u_{1,j}^i, x_{2,j}^i, y_{2,j}^i, u_{2,j}^i \rangle \end{aligned} \quad . \tag{1}$$

The PSM is defined on a ground-plane map, a 2D slice of the world coordinate along the ground-plane. PSM consists of the ground plane G and a set of walls $\{W_1, W_2, W_3, ..., W_n\}$. The location of a wall W_i is specified by a line on the ground-plane map parametrized by (α_i, d_i), where $\alpha_i \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$ is the orientation of the line and $d_i \in \mathbb{R}$ is the directed distance from the origin to the line. A wall W_i consists of a set of wall segments $\{S_1^i, S_2^i, ..., S_{m_i}^i\}$ delimiting where the wall is present and where there is an opening. A wall segment S_j^i is represented by two endpoints. An endpoint is defined by its location $(x_{\bullet,j}^i, y_{\bullet,j}^i)$ on the ground-plane map and its type $u_{\bullet,j}^i$ representing different levels of understanding of the bound of the wall segment.

D. Extract Features

After extracting the ground-plane, we group the points (pixels) from the current RGB-D frame into a set of vertical-plane features $\mathbf{V} = \{v_j | j = 1, ..., n_v\}$ and a set of point-set features $\mathbf{C} = \{c_j | j = 1, ..., n_c\}$. Figure 3 is an example of the features. A vertical-plane feature v is a plane segment that is perpendicular to the ground-plane, and a point-set feature is a cluster of 3D points that cannot be grouped into vertical planar segments. Vertical-plane features suggest the existence of walls, and point-set features suggest the existence of clutter. However, not all the vertical-plane features belong to the PSM. For example, a box on the ground also generates several vertical-plane features. On the opposite, a point-set

feature that is close to a wall may simply be a small instance sticking out from the wall or noise. For example, a door knob may become a point-set feature. In this paper, we use the vertical-plane features to generate and refine the hypotheses, and we use all features to test the hypotheses.

Similar to a PSM wall, a vertical-plane feature v corresponds to a line segment in the ground-plane map:

$$v = \langle \alpha^v, d^v, x_1^v, y_1^v, x_2^v, y_2^v \rangle \tag{2}$$

where $\alpha^v \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$ and $d^v \in \mathbb{R}$. This is the same line parametrization used to represent a wall plane, and (x_1^v, y_1^v) and (x_2^v, y_2^v) are two end-points of the line that denotes where the vertical-plane is visible. We first extract vertical-plane features from the 3D points that are not on the ground plane. (Points are represented in the 3D world coordinate.) We project the points onto the ground-plane map, and use J-linkage [13] to fit a set of line segments. Each line segment forms a vertical-plane feature. We remove the points that form the vertical-plane features and cluster the remaining points based on their 3D Euclidean distances. A cluster that consists of more than 100 points forms a point-set feature. Mathematically, a point-set feature c is represented by

$$c = \langle x^c, y^c, \mathbf{P}^c \rangle \tag{3}$$

where $\mathbf{P}^c$ is the set of 3D points in the cluster, and (x^c, y^c) is the ground-plane map projection of the 3D mean location of the points. For points that failed to form features are considered as noise and are thus, discarded.

E. Refine Hypotheses

We use the information from the current frame to refine the precision of each hypothesis. The generic Extended Kalman Filter (EKF) is used to estimate the parameters of each wall and the location of each occluding endpoint [14]. [3]

Vertical-plane features V_t are potential measurements for the walls that are visible in frame t. We associate the visible walls to the vertical-plane features based on the similarity of their corresponding lines on the ground-plane map. If a wall w_i is associated to a vertical-plane feature v_j, we use EKF to update the wall parameters (α_i, d_i) based on the location of the vertical-plane feature (α_j^v, d_j^v).

Once the wall parameters are updated, we refine the location of each occluding endpoint. Potential measurements for an occluding endpoint are the end-points of the vertical-plane features that have similar parameters with its corresponding wall. We project these end-points onto the line of the corresponding wall and find the nearest projected point to the endpoint. If the distance between the nearest point and the endpoint is less than 0.2 meters, the projected point is the measurement for that endpoint. An endpoint is updated using EKF, if its measurement is available.

[3]Occluding endpoints is a type of endpoint that is associated to only one observed wall. [14]

F. Generate Hypotheses

We combine vertical-plane features $\mathbf{V}_t$ to generate PSM hypotheses. In the first frame, a set of simple hypotheses are generated. A simple hypothesis is either a PSM with two parallel walls or with at most three walls where each wall intersects with its adjacent walls. These simple hypotheses are generated by combining vertical-plane features with certain constraints [15]. At every 10 frames, we transfer each existing hypothesis into a set of child hypotheses describing the same environment in more details. These child hypotheses are essential for incrementally modeling the environment as a robot travels. For example, when the robot is at a long corridor, the wall at the end of the corridor or openings along the side walls, may not be sensible from a distance, and thus, hypotheses capturing these details can only be generated when the robot is close to them. Given an existing hypothesis, its children hypotheses are generated by adding openings to the walls [14] or by combining with a simple hypothesis generated at the current frame.

G. Test Hypotheses

Hypotheses are tested using a recursive Bayesian filtering framework [15], [14]. Starting from a uniform prior, at each frame t, we compute the likelihood function $p(\mathbf{F}_t|M_i)$ of each hypothesis M_i to update its posterior probability $p(M_i|\mathbf{F}_{1:t})$. In our previous work, we assume that a hypothesis explains all the observed features in the current frame. This assumption works well in empty environments because all the observed features actually lie on the walls and ground planes. However, in a cluttered environment, features may be part of clutter which cannot be explained by the PSM, as shown in Figure 3. Thus, in this paper, we design a likelihood function that allows a good hypothesis to explain only a subset of the observed features $\mathbf{F}_t = \{\mathbf{V}_t, \mathbf{C}_t\}$.

The likelihood of hypothesis M_i consists of three terms,

$$p(\mathbf{F}_t|M_i) = p_c(\mathbf{F}_t|M_i)p_a(\mathbf{F}_t|M_i)p_s(\mathbf{F}_t|M_i). \tag{4}$$

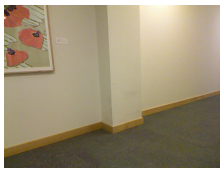
These terms describe a three-way trade-off among the feature coverage by the hypothesis $p_c(\mathbf{F}_t|M_i)$, the accuracy of the explained features $p_a(\mathbf{F}_t|M_i)$, and the simplicity of the hypothesis $p_s(\mathbf{F}_t|M_i)$.

The coverage term $p_c(\mathbf{F}_t|M_i)$ measures the number of features that M_i explains, regardless of the accuracy of the explanations. Formally,

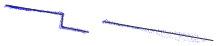
$$p_c(\mathbf{F}_t|M_i) = \omega_v \frac{|\mathbf{V}_t^i|}{|\mathbf{V}_t|} + \omega_c \frac{|\mathbf{C}_t^i|}{|\mathbf{C}_t|} \tag{5}$$

where $\mathbf{V}_t^i \subseteq \mathbf{V}_t$ are the vertical-plane features, and $\mathbf{C}_t^i \subseteq \mathbf{C}_t$ are the point-set features that are explained by M_i. (One can define their own metric to determine whether a feature is explained or not. Our metric is presented in the appendix.) ω_v and ω_c are the importances of explaining vertical-plane features and point-set features, respectively ($\omega_v + \omega_c = 1$). [4]

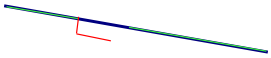
[4]In indoor environment, a vertical-plane feature is more likely to be part of a wall plane, while a point-set feature is more likely to be clutter. Thus, we set $\omega_v > \omega_c$. In our experiments, we set $\omega_v = 0.7$ and $\omega_c = 0.3$.



(a) Frame t

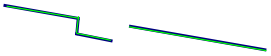


(b) Vertical-plane features extracted from the observation of frame t



(c) M_1: simple 1-wall model that perfectly explains some features

(d) M_2: simple 1-wall model that moderately explains all features



(e) M_3: complex 4-wall-segment model that perfectly explains all features

Hypothesis	M_1	M_2	M_3
coverage $p_c(\mathbf{F}_t\|M_i)$	0.50	1.00	1.00
accuracy $p_a(\mathbf{F}_t\|M_i)$ ($\sigma = 0.2$)	1.00	0.88	1.00
accuracy $p_a(\mathbf{F}_t\|M_i)$ ($\sigma = 0.05$)	1.00	0.14	1.00
simplicity $p_s(\mathbf{F}_t\|M_i)$ ($\gamma = 0$)	0.50	0.50	0.50
simplicity $p_s(\mathbf{F}_t\|M_i)$ ($\gamma = 1$)	0.88	0.88	0.27
Likelihood $p(\mathbf{F}_t\|M_i) = p_c(\mathbf{F}_t\|M_i)p_a(\mathbf{F}_t\|M_i)p_s(\mathbf{F}_t\|M_i)$			
CASE 1: $\sigma = 0.20$, $\gamma = 0$	0.25	0.44	**0.50**
CASE 2: $\sigma = 0.20$, $\gamma = 1$	0.44	**0.77**	0.27
CASE 3: $\sigma = 0.05$, $\gamma = 0$	0.25	0.07	**0.50**
CASE 4: $\sigma = 0.05$, $\gamma = 1$	**0.44**	0.12	0.27

(f) Three-Way trade-off among likelihood factors

Fig. 4. Example for the three-way trade-off among the likelihood factors. (Best viewed in color.) Assume we extract four vertical-plane features from frame t (a), as shown in (b), and three valid hypotheses (thick blue lines) are generated: (c) Hypothesis M_1 models the environment with one wall that perfectly explains only two of the features (green), leaving the other two unexplained (red); (d) Hypothesis M_2 models the environment with one wall that explains all the features but explain them poorly (yellow); (e) Hypothesis M_3 models the environment with four wall segments that explain all four features perfectly. As shown in (f), the likelihood function is a three-way trade-off among feature coverage of the hypothesis, accuracy of the explained features, and simplicity of the hypothesis. Depending on the parameters in each term, the likelihood function can prefer any of the hypotheses. If σ is small, the accuracy term is important since the Gaussian function in $p_a(\mathbf{F}_t|M_i)$ gives large penalties to poor explanations. Contrary, if σ is large, the accuracy term becomes less important because the Gaussian function is less discriminative between poor and good explanations. In this example, $p_a(\mathbf{F}_t|M_2)$ dramatically increases when σ increases. The decay rate γ controls the preference towards simpler hypotheses (Fig. 5). When $\gamma = 0$, there is no preference for the simplicity of the hypothesis. As γ increases, the likelihood function will increases its preference towards simpler hypotheses. In CASE 1, coverage is the most important factor, and since M_3 has a slightly better accuracy than M_2, M_3 has the highest likelihood. In contrast to CASE 1, CASE 2 consider both coverage and simplicity important, and thus, M_2 has a higher likelihood than the complex hypothesis M_3. In CASE 3, where only coverage and accuracy are considered, M_3 is the best because it perfectly explains all the features while M_1 and M_2 have issues with coverage and accuracy, respectively. In all three cases, M_1 is not preferred, because it has a lower coverage. However, if considering all three factors (CASE 4), M_1 is the best.

The accuracy term $p_a(\mathbf{F}_t|M_i)$ measures the accuracy of the features explained by M_i. Since different hypotheses may explain different subsets of the features $\mathbf{F}_t$, we compute the weighted RMS error $error(\mathbf{F}_t, M_i)$ of the features that hypothesis M_i explains,

$$error(\mathbf{F}_t, M_i) = \sqrt{\frac{\omega_v \sum_{v_j \in \mathbf{V}_t^i} \varepsilon_p(v_j, M_i)^2 + \omega_c \sum_{c_j \in \mathbf{C}_t^i} \varepsilon_c(c_j, M_i)^2}{\omega_v |\mathbf{V}_t^i| + \omega_c |\mathbf{C}_t^i|}} \quad (6)$$

$\varepsilon_p(v_j, M_i)$ and $\varepsilon_c(c_j, M_i)$ are the error of M_i explaining feature v_j and c_j. (One can define their own error metric. Our metric is presented in the appendix.) The RMS error is modeled by a Gaussian distribution with zero mean and σ^2 variance [5]. Mathematically,

$$p_a(\mathbf{F}_t|M_i) \propto \begin{cases} 0 & \text{if } |\mathbf{V}_t^i| + |\mathbf{C}_t^i| = 0 \\ \exp \frac{-error(\mathbf{F}_t, M_i)^2}{2\sigma^2} & \text{otherwise} \end{cases}. \quad (7)$$

Note that both the coverage and the accuracy term require a hypothesis to explain at least one feature in order to obtain a non-zero likelihood. In other words, the Bayesian filter will filter out a hypothesis that explains nothing in view.

The simplicity term $p_s(\mathbf{F}_t|M_i)$ measures the amount of information that M_i used to explain the features. This is a regularization term that prevents the Bayesian filter from over-fitting the features. $p_s(\mathbf{F}_t|M_i)$ is modeled by a generalized logistic function where the growth rate γ is negative,

$$p_s(\mathbf{F}_t|M_i) = \frac{1}{1 + exp\gamma(|M_i|_t - n_{max_\gamma})}. \quad (8)$$

[5]In our experiments, $\sigma^2 = 0.04$. This value is selected to account for the noise of the sensor and the accumulated error for pose estimation.

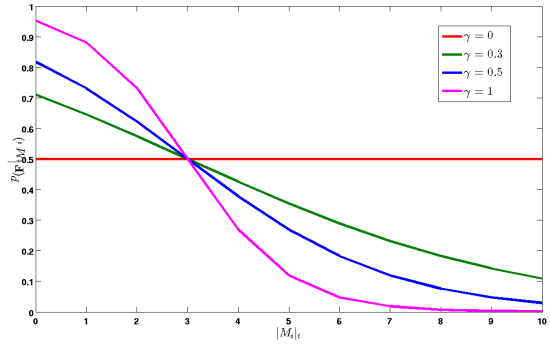
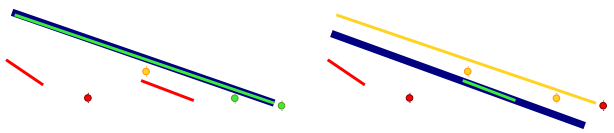


Fig. 5. The simplicity term $p_s(\mathbf{F}_t|M_i)$ of the likelihood function with respect to different γ with a fixed $n_{max_\gamma} = 3$. (Best viewed in color.) The horizontal axis $|M_i|_t$ is the number of walls in view and the vertical axis is $p_s(\mathbf{F}_t|M_i)$. In the extreme case, where $\gamma = 0$, the likelihood function has no preference towards simpler hypotheses. As γ increases, the Bayesian filter is more likely to converge to a simpler hypothesis.

$|M_i|_t$ is the number of walls that are visible in frame t and n_{max_γ} is where the maximum decay occurs. [6]

Figure 4 demonstrates the trade-offs among the three terms (coverage $p_c(\mathbf{F}_t|M)$, accuracy $p_a(\mathbf{F}_t|M)$, and simplicity $p_s(\mathbf{F}_t|M)$) in the likelihood function. The importance of each term is controlled by two parameters, the variance of the Gaussian function σ^2 in the accuracy term $p_a(\mathbf{F}_t|M)$ and the decay rate γ in the simplicity term $p_s(\mathbf{F}_t|M)$. The variance σ^2 mainly controls the importance of the accuracy term. A large variance σ^2 in the accuracy term means that the difference between good and poor explanations is low, and

[6]In our experiments, $\gamma = 0.3$ and $n_{max_\gamma} = 10$ to ensure the likelihood for having one to five walls are similar because theses are the common number of walls that are visible in a frame based on the field of view of our depth camera (57° horizontally).



(a) M_4 — A good hypothesis (b) M_5 — A mediocre hypothesis

	Full Explanation		Partial Explanation	
Hypothesis	M_4	M_5	M_4	M_5
coverage p_c	1	1	0.46	0.94
accuracy p_a	0.12	0.25	0.95	0.29
simplicity p_s	same			
Likelihood	0.12	0.25	0.44	0.31

(c) Likelihood comparisons with and without clutter concept

Fig. 6. The environment in Fig. 2 and 3 demonstrates the importance of allowing a hypothesis to explain only a subset of the features. (Best viewed in color) Both M_4 and M_5 use one wall (blue thick line) to represent the environment, so the simplicity term for both hypotheses are the same. If we force both hypotheses to explain all the features, then M_5 has a higher likelihood than M_4 because it has a lower RMS error. ($\omega_v = 0.7$ and $\sigma = 0.2$) If we allow the hypotheses to explain only a subset of the features, M_4 is most likely to have a higher likelihood. Green features are perfectly explained (0 error) by the hypothesis, yellow features are explained with errors, and red features are not explained. See text for more details.

thus, the likelihood function is more reflective on the feature coverage. Contrary, a small variance σ^2 means that the penalty for poor explanation is high, and thus, the accuracy of the explanation is highly important. The decay rate γ controls the preferences toward having a simpler model as shown in Fig. 5. In the extreme case, where $\gamma = 0$, the simplicity term will be the same for all hypotheses and thus, the likelihood will not prefer simpler hypotheses. As γ increases, the likelihood function will start preferring a simpler hypothesis with reasonable coverage and accuracy.

On one hand, the parameters in the likelihood function allows the user to set their preferences based on their applications. If the purpose of exploration and mapping is to determine free-space for safe navigation, then one set of parameter values might be preferred, while if the purpose is to create an architectural CAD model of the environment, a different set of values may be preferable. On the other hand, the parameters may seem to be sensitive to the hypothesis that the Bayesian filter converges to. In fact, for the purpose of navigation, it is reasonable to converge to any of the three hypotheses. Both M_3 and M_1 specify the same part of the environment as free-space, but they specify the free-space in a different way. M_2 is less precise in specifying the boundaries of free-space, and the accuracy of its explanation $p_a(\mathbf{F}_t|M_2)$ requires a robot to be more cautious about the boundaries. In other words, the accuracy term provides a confidence measurement of free-space for navigation algorithms [8]. In a simple empty environment like this, most hypotheses are reasonable. However, in a more complex or cluttered environment, many bizarre hypotheses will be generated and the likelihood function allows us to converge to a reasonable hypothesis (see Section III).

Figure 6 is an example that shows why it is important to allow a good hypothesis to explain only a subset of the observed features. If we require a hypothesis to explain all the observed features ($p_c(\mathbf{F}_t|M_4) = p_c(\mathbf{F}_t|M_5) = 1$), then the mediocre hypothesis M_5 dominates M_4 because it has lower RMS error. However, if a hypothesis can distinguish between clutter and non-clutter features, then the higher-quality partial explanation M_4 can dominate the nearly complete but lower-quality hypothesis M_5.

III. RESULTS

We evaluated our approach using four RGBD video datasets in various indoor environments. [7] The videos were collected by a Kinect sensor mounted on a wheeled device with a front pointed direction. The relative poses between the camera and the ground plane are fixed within each video, but different among different videos. In Dataset CORNER and LAB, the robot traveled about 1.5 meters in a very cluttered corner. In Dataset INTERSECTION, the robot made a right turn around an empty L-intersection. In Dataset CORRIDOR, the robot traveled about 5 meters in a long corridor with objects on the sides.

Figure 8 shows the results on these videos. Our method converges to a reasonable hypothesis to describe each environment in 3D. Once we obtain the best hypothesis, we separate out clutter, observations that were not explained by the hypothesis. At each frame, each hypothesis has its own partition of explained and unexplained features. The unexplained observations are the 3D points that contribute to these unexplained features at each frame. We further cluster these unexplained observations into a set of 3D regions based on their Euclidean distances. In most cases, a cluttered region is either an object or a pile of objects, but in some situations (INTERSECTION), the cluttered region may be part of the building, such as a pillar along the wall.

To evaluate our method quantitatively, for every 10 frames, we manually labeled the ground truth classification of the planes (i.e. the walls, ground and ceiling), and ground-truth classification of clutter in the projected image space. We define the *Plane Accuracy* of a hypothesis being the percentage of pixels with the correct plane classification, and the *Scene Accuracy* of a hypothesis being the percentage of pixels with the correct scene interpretation (PSM + clutter). When computing these accuracies, only non-ceiling pixels with valid depth data are considered, because ceiling is not modeled in PSM and pixels with invalid depth data are not used to compute the likelihood. The average accuracies of the maximum a posteriori hypotheses are reported in Fig. 7. To illustrate the importance of partial explanation, we ran an experiment without the concept of clutter ($p_c(\mathbf{F}_t|M_i) = 1$ and $p_a(\mathbf{F}_t|M_i)$ is computed by all the features). Among all datasets, only INTERSECTION converges to the correct hypothesis, and the overall *Plane Accuracy* accuracy without clutter concept is 67.11%. Thus, allowing partial explanation is a key to handle cluttered environments.

Besides locating free-space for navigation, our output interpretation is an important step towards reasoning about

[7] Publicly available at `http://www.eecs.umich.edu/~gstsai/release/Umich_indoor_corridor_2014_dataset.html`.

Dataset	CORNER	LAB	CORRIDOR	INTERSECTION	Overall
Clutterness	34.84%	20.80%	9.50%	10.22%	16.50%
Plane Accuracy	98.18 %	99.31 %	97.83 %	98.25 %	98.49 %
Scene Accuracy	92.82 %	95.10 %	91.76 %	98.16 %	94.83 %

Fig. 7. Quantitative evaluation. The average accuracies of the maximum a posteriori hypotheses at the evaluated frames are reported. *Plane Accuracy* measures the accuracy of the hypothesized PSM model, and *Scene Accuracy* measures the accuracy of the whole interpretation (PSM + clutter).

objects in the local environment. Our interpretation segments out regions that may contain objects for object reasoning methods. A clutter region is a smaller problem for object reasoning compare to the whole scene. For example, we applied a functionality-based object classification [6] to the clutter region with the chairs in LAB and determined that this region is "sittable" among other functionality classes ("table-like", "cup-like", and "layable").

IV. CONCLUSION

We addressed the dilemma of model-based scene interpretation. On one hand, it is valuable to interpret the environment with a simple model and separate out the regions that cannot be described by a simple model — clutter, as oppose to having a fine-grained model that models all the details in the environment. On the other hand, while testing multiple model hypotheses, it is more preferable to select a hypothesis that explains more observations and has less unexplainable observations. We proposed a likelihood function handles the dilemma. The likelihood function make explicit a three-way trade-off among coverage of the observed features, accuracy of the explanation, and simplicity of the hypothesis.

We demonstrated the likelihood function in the context of an on-line generate-and-test framework to visual scene understanding [14]. Our experimental results on a variety of RGB-D videos demonstrated that our method is capable of interpreting cluttered environment by a simple model (PSM) and clutter. Our output interpretation not only provides information of free-space for navigation, but separates out segments of clutter represented by 3D point clouds that require further analysis with more complex models. Our approach can be applied to further interpret clutter with other models (e.g. object models).

APPENDIX

Explaining vertical-plane features: A vertical-plane feature v is explained by hypothesis M if v can be explained by a wall segment S_j^i in M. Feature v is explained by M if the error $\varepsilon_p(o^p, M)$ is less than a threshold ϵ. ($\epsilon = 0.1$ in our experiments.) The error metric $\varepsilon_p(v, M)$ is computed as followed. First, we find the wall plane $W_{match} \in M$ that best matches v by computing the displacement $dist(v, W_i)$ between v and each wall W_i based on their corresponding lines in the ground-plane map. For efficiency, we only consider a wall that has a similar angle to the vertical-plane feature. Two angles are similar, if $\min(|\alpha_i - \alpha^v|, \pi - |\alpha_i - \alpha^v|) < 0.0873$. If no walls are within the angle constraints, v is not explained by M. Each wall has a coordinate for computing the displacement. The origin of the coordinate is at the weighted center of the line of v and the line of wall W_i, and the x-axis is along the weighted average direction of the two lines. The weight of each line is proportional to its length. The displacement $dist(v, W_i)$ of the two lines is defined as the maximum difference along the y-axis. W_{match} is the wall with minimum $dist(v, W_i)$. If the entire v lies within a single wall segment of W_{match}, $\varepsilon_v(v, M) = dist(v, W_{match})$. Otherwise, v is not explained by M.

Explaining point-set features: We define an error metric $\varepsilon_c(c, M)$ to be the shortest distance of the point-set feature (x^c, y^c) to a wall W_i in M. Feature c is not explained by M if the error is larger than a threshold ϵ. (This is the same threshold ϵ for explaining vertical-plane features.) Feature c is not explained if the projected location of the feature does not lie within the bound of any wall segments of W_i. Moreover, we take into consideration the distribution of the points $\mathbf{P}^c$ in c. Only if 70% of the points are within ϵ of distance to wall W_i, point-set feature c is explained.

REFERENCES

[1] A. J. Davison. Real-time simultaneous localization and mapping with a single camera. *ICCV*, 2003.
[2] A. Flint, D. Murray, and I. Reid. Manhattan scene understanding using monocular, stereo, and 3d features. *ICCV*, 2011.
[3] A. Furlan, S. Miller, D. G. Sorrenti, L. Fei-Fei, and S. Savarese. Free your camera: 3d indoor scene understanding from arbitrary camera motion. *BMVC*, 2013.
[4] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Reconstructing building interiors from images. *ICCV*, 2009.
[5] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. *ICCV*, 2009.
[6] L. Hinkle and E. Olson. Predicting object functionality using physical simulations. *IROS*, 2013.
[7] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke. Real-time plane segmentation using RGB-D cameras. *RoboCup 2011: Robot Soccer World Cup XV*, 2012.
[8] C. J. Jong Jin Park and B. Kuipers. Robot navigation with model predictive equilibrium point control. *IROS*, 2012.
[9] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. *ISMAR*, 2007.
[10] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. *CVPR*, 2009.
[11] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. *ICRA*, 2006.
[12] C. J. Taylor and A. Cowley. Parsing indoor scenes using RGB-D imagery. *RSS*, 2012.
[13] R. Toldo and A. Fusiello. Robust multiple structure estimation with J-linkage. *ECCV*, 2008.
[14] G. Tsai and B. Kuipers. Dynamic visual understanding of the local environment for an indoor navigating robot. *IROS*, 2012.
[15] G. Tsai, C. Xu, J. Liu, and B. Kuipers. Real-time indoor scene understanding using Bayesian filtering with motion cues. *ICCV*, 2011.
[16] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. *ECCV*, 2010.

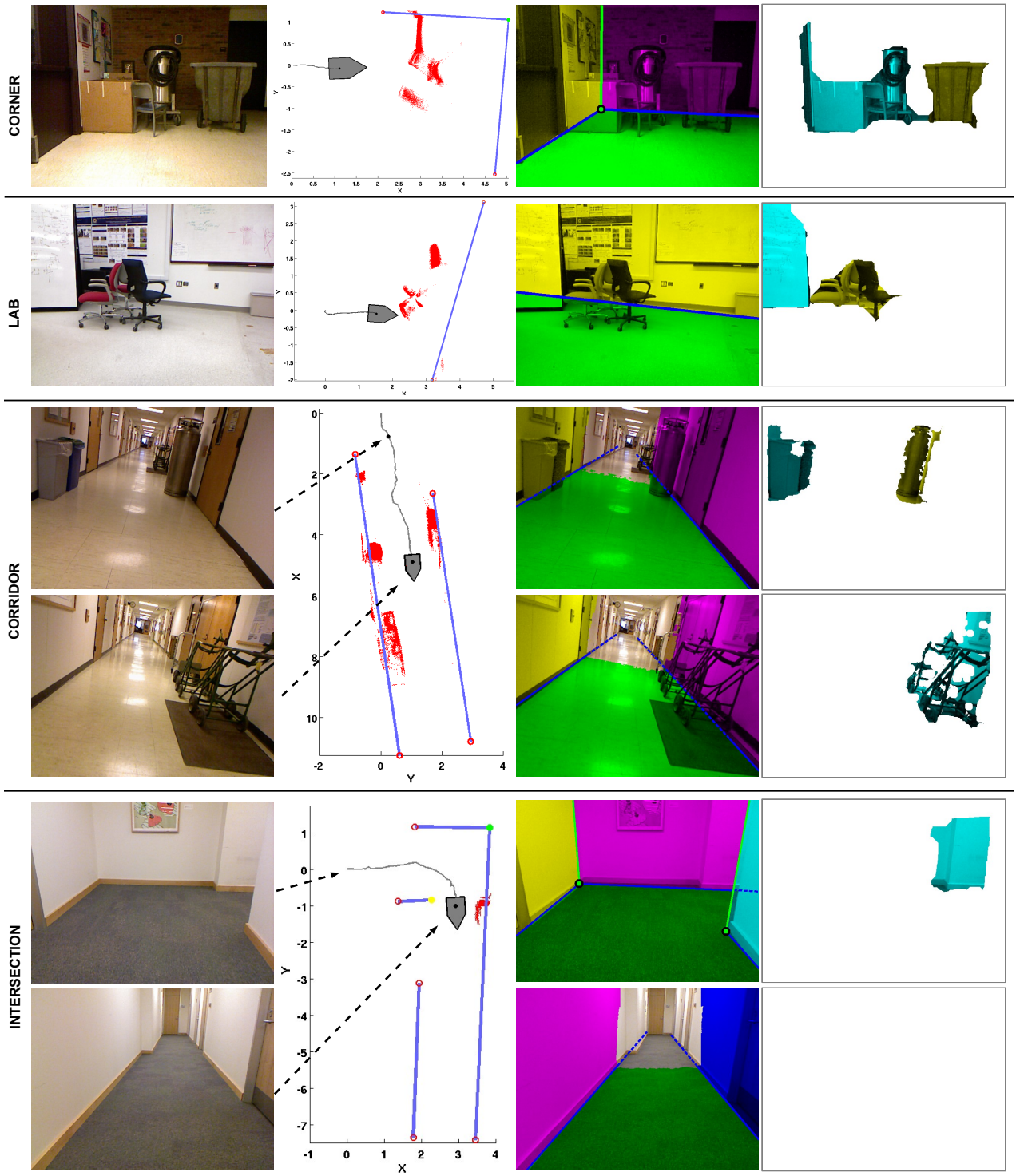


Fig. 8. Evaluation of our approach on interpreting indoor environment by a Planar Semantic Model and clutter. (Best viewed in color.) The first column is the image in our dataset. (The depth images are not shown in this figure.) The second column is the 3D interpretation obtained by our method. The interpretation is visualized in the ground-plane map. For the PSM, the blue lines are the wall planes, and the big dots are the endpoints. The endpoints are color coded based on their types (green: dihedral; yellow: occluding; red: indefinite)[14]. Clutter, observations that are not explained by the model, is represented by a 3D point cloud, which is visualized by the tiny red dots. Notice that some of the clutter points are caused by errors in the pose estimation. The pose estimation is less accurate in CORRIDOR because a large portion of the pixels have unreliable depth measurements due to distances. The third column is the image projection of the PSM. The ground is green and each wall is shown in a different color. The part of the images that are not painted are too far away from the robot to obtain a reliable depth data, so the robot will start modeling those regions as it gets closer. The fourth column shows the clutter in the image. We cluster the clutter points in 3D into regions, and each cluster corresponds to a 2D segment in the image space. These segments are shown in different colors. In general, each clutter region consists of an object or a pile of objects, but in INTERSECTION, the clutter region is actually a pillar along the wall. Our interpretation does not model the objects, since PSM can only represent the ground-plane and walls. However, if object models are given, our method can be applied to factor each clutter region into known objects and points that remains unexplained.