2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Probabilistic Plane Extraction and Modeling for Active Visual-Inertial Mapping

Mitchell Usayiwevu[1], Fouad Sukkar[1,2] and Teresa Vidal-Calleja[1,2]

***Abstract*— This paper presents an active visual-inertial mapping framework with points and planes. The key aspect of the proposed framework is a novel probabilistic plane extraction with its associated model for estimation. The approach allows the extraction of plane parameters and their uncertainties based on a modified version of PlaneRCNN [1]. The extracted probabilistic plane features are fused with point features in order to increase the robustness of the estimation system in texture-less environments, where algorithms based on points alone would struggle. A visual-inertial framework based on Iterative Extended Kalman filter (IEKF) is used to demonstrate the approach. The IEKF equations are customized through a measurement extrapolation method, which enables the estimation to handle the delay introduced by the neural network inference time systematically. The system is encompassed within an active mapping framework, based on Informative Path Planning to find the most informative path for minimizing map uncertainty in visual-inertial systems. The results from the conducted experiments with a stereo/IMU system mounted on a robotic arm show that introducing planar features to the map, in order to complement the point features in the state estimation, improves robustness in texture-less environments.**

I. Introduction

Robotic mapping has been a highly active research area in robotics for over four decades. Formally, mapping is the problem of integrating the information gathered with a robot's sensors into a given representation of the world [2]. This gives the robot knowledge of what the environment it is working in looks like and enables autonomous operations.

Active mapping is a subclass of the mapping problem that focuses on finding trajectories to use during the map building. Typically active mapping aims to search for the best possible representation of an environment by minimizing the mapping uncertainty [3]. The availability of low cost and low weight IMUs and cameras, coupled with the rich information and frequent sensor data they offer has led to an increase in the use of visual-inertial systems for active mapping tasks. As a result, we address the active mapping problem in the context of *Visual-Inertial Odometry* (VIO). Traditional VIO estimation algorithms ordinarily work by extracting points from the camera data, which are used as features in the map. However, the use of point features alone poses limitations in that the systems are incapable of handling low texture environments. A common work-around to this problem found in the literature, is to use planar features to complement the points in the map, and increase robustness of the estimation. Man-made environments are usually rich in static planar structures. This planar information can act as an effective regularizer for landmark locations, improving estimation accuracy.

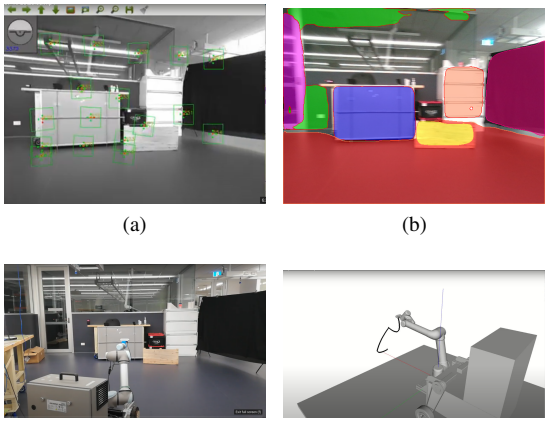

Fig. 1: A 6-DOF UR5 arm executing an informative trajectory from our planner that minimizes mapping uncertainty. The scene with the UR5 arm is shown in (c) and (d) is the 3D executed trajectory. (a) shows the point features while (b) shows the plane features used by the estimation algorithm to increase robustness in low texture scenes.

In this paper, we are interested in active mapping with both points and planes, with the plane features extracted by a deep learning front-end. Deep learning methods learn good representations of the input images at multiple levels, which results in robust and effective features that are not affected by varying lighting. Most deep learning-based plane extraction methods however do not output uncertainty, which is key for VIO estimation frameworks and, even more important, for an active mapping framework that relies on uncertainty propagation. Therefore, we proposed a modified version of the state-of-the-art plane extraction neural network, PlaneRCNN [1]. It is modified to output not only the parameter estimates of the planes in the environment but also the uncertainty associated with estimation.

This probabilistic plane extraction method is encompassed within an *Informative Path Planning* (IPP) framework to actively find the continuous optimal trajectory for reducing mapping uncertainty[4]. A modified version of ROVIO [5], a VIO framework based on the IEKF, is used for state estimation with point and planar features. We customize the standard IEKF equations to enable the estimation to handle the delay introduced by the neural network inference time. A summary of the contributions of this paper is as follows:

- A probabilistic plane extraction method and subsequent

[1] Authors are with the UTS Robotics Institute, University of Technology Sydney, 2007, Ultimo, NSW, Australia and [2] the Australian Cobotics Centre (ITTC for Collaborative Robotics in Advanced Manufacturing). Corresponding author Mitchell.Usayiwevu@student.uts.edu.au.

This research is supported by the UTS International Research Scholarship and the ITT Centre for Collaborative Robotics in Advanced Manufacturing funded by ARC (Project ID: IC200100001).

model for estimation that produces plane parameters and their associated uncertainties.
- A visual-inertial active mapping framework that finds the continuous optimal path to minimize uncertainty on points and delayed planar features for robustness.

This approach is evaluated in an experimental set-up with a stereo camera/ Inertial Measurement Unit (IMU) system mounted on a UR5 in low texture scenes as shown in Fig. 1.

II. Related Work

Due to the extensive use of cameras and IMUs in robotics, there has been an increase in visual-inertial based state estimation algorithms. These include VIO [6] and Visual-Inertial Simultaneous Localization and Mapping (VI-SLAM) [7], which fuse visual data with inertial measurements for both, localisation and mapping purposes.

Considering planes, work in [8] show that including plane features to the visual SLAM problem improves the accuracy of the state estimation and mapping in texture-less environments. Popular visual SLAM algorithms like DVO-RGB-D SLAM [9], ORB-RGB-D SLAM [10] and ORB-SLAM3 [11] fail in such scenarios with low-texture images. In [12], the authors propose a robust and lightweight monocular visual-inertial odometry system using multiplane priors. The planes extracted from the point cloud and are used for fast localization and for stabilizing the depths under small-translation movements. Ultimately, this improves the accuracy of the odometry framework. Authors in [13] jointly use point and planar landmarks to do SLAM. In a similar way, [14] presents a SLAM method that jointly estimates camera position and planar landmarks in the map within a linear Kalman filter framework using RGB-D data. A major drawback of their work is that they do not include rotational motion as part of their state vector in an attempt to linearize the SLAM problem, while we consider rotations in our work.

Works in [15], [16] and [17] are closely related to our work. These approaches integrate traditional VIO with planes extracted from deep learning algorithms. In [15] a CNN is used to extract planar features from RGB images which are used in VIO for estimation. Authors of [16] leverage the geometry of planes for improved robustness and accuracy in challenging and dynamic environments. The biggest distinction between these works and ours is that our neural network produces plane parameters and their associated uncertainties, which allows us to integrate the planar features in a direct manner with appropriate noise learnt from the data and the model itself.

In addition, we are interested in active mapping where the literature mainly focuses on active SLAM since the robotic localization and mapping problems are inherently linked. The majority of active SLAM algorithms are information gain based. To select an action(s), an information gain-based planner compares the corresponding expected information gain for each of the available actions. The planner will select the action that maximizes the information gain [18], [19], [20]. In [21], the authors solve a similar problem with the aim of maximizing the map accuracy during exploration. They solve the problem using an information gain maximization method. Although these works are similar to ours in that we also use IPP, our work additionally considers visual-inertial integration which incorporates probabilistic planar features to improve the robustness of the mapping task.

III. Overview

Let us consider a visual-inertial system carrying out a mapping task in the environment $\mathcal{E}$ using a probabilistic estimation framework. Given the IMU and camera measurements, in the form of point features and plane centroid location, plane normals and their associated uncertainties acquired during the execution of a path $\pi_{0:k}$ between times $t = 0$ and $t = t_k$, the state $\mathbf{x}_k$ and covariance matrix $\mathbf{P}_k$ of the system at time $t = t_k$ are estimated with an IEKF-based VIO framework which uses points and planes. An IPP framework takes the current estimates and forecasts planes and points estimates using continuous trajectories to account for IMU readings with a sampling based planner, in order to maximize mapping information gain.

IV. Probabilistic PlaneRCNN front-end

As mentioned above, our active mapping framework requires the correct treatment of uncertainties from the mapped features. We opted to use PlaneRCNN, however, this learning-based plane extraction approach does not output uncertainties inherently. In this section, we present the probabilistic variant custom developed to cater for our framework.

PlaneRCNN [1] is a deep neural network architecture that can detect an arbitrary number of plane instances in an image, giving the associated plane parameters, segmentation mask and depth-map for each image. Since an estimate of uncertainty on the plane parameters is a pre-requisite for our active mapping framework, we proposed a modification of PlaneRCNN following the work by Loquercio *et al.* in [22] on a general framework for uncertainty estimation in deep learning. This modification allows us to estimate the uncertainty of the predicted plane normals which is derived from two sources; model uncertainty and data uncertainty. The model uncertainty arises from imbalances in the training data distribution and the data uncertainty is generated by the noisy measurements from the sensors.

Forward propagation of the sensor noise through PlaneRCNN is done via a technique known as *Assumed Density Filtering* ADF [23]. This is achieved by transforming PlaneRCNN into a Bayesian belief network, i.e replacing all activations including the inputs, and outputs with probability distributions as shown in Fig. 2. Given the parameters of the previous activation distribution $\mathtt{q}(\mathtt{z}^{(i-1)})$, the ADF approximation results in a recursive formula to compute the activation mean and uncertainty, $(\mu^{(i)}, \mathtt{v}^{(i)})$:

$$\mu^i = \mathbb{E}_{\mathtt{q}(\mathtt{z}^{(i-1)})}[\mathtt{f}^{(i)}(\mathtt{z}^{(i-1)})] \tag{1}$$

$$\mathtt{v}^i = \mathbb{V}_{\mathtt{q}(\mathtt{z}^{(i-1)})}[\mathtt{f}^{(i)}(\mathtt{z}^{(i-1)})] \tag{2}$$

where $\mathbb{E}$ and $\mathbb{V}$ denote the expectation and variance of the distribution and $\mathtt{f}^{(i)}$ is the i-th layer function which transforms the activation $\mathtt{z}^{(i-1)}$ into a distribution. We refer

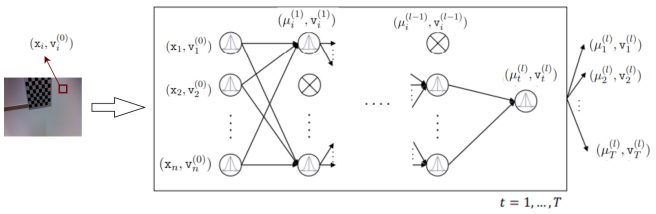

Fig. 2: The transformation of PlaneRCNN into a Bayesian belief network An ensemble of T such networks is created by enabling dropout at test time. Figure is adapted from [22].

the reader to [22] and [24] for a derivation of the formulae. The model uncertainty is represented by placing a distribution over the network's weights, $\boldsymbol{\omega}$. Monte-Carlo based approaches approximate this distribution by using dropouts at test time. The model uncertainty is therefore given by:

$$\sigma_{model} = \frac{1}{T}\sum_{i=1}^{T}(\mathtt{y}_t - \overline{\mathtt{y}})^2 \tag{3}$$

where $\{\mathtt{y}_t\}_{t=1}^{T}$ is a set of T sampled outputs for weights instances $\boldsymbol{\omega}^t \sim \mathtt{q}(\boldsymbol{\omega}; \boldsymbol{\Phi})$ and $\overline{\mathtt{y}} = \frac{1}{T}\sum_t \mathtt{y}_t$.

Note that the plane representation of PlaneRCNN comprises three parameters which are equal to the product of the plane normal and the plane offset $\mathbf{n}d$. Thus, the final plane parameters prediction $\mathbf{n}d$ and the associated total uncertainty $\boldsymbol{\iota}$ of the network output $\mathtt{y}$ for an input sample $\mathtt{x}$ corrupted by noise $\mathtt{v}^{(0)}$ is:

$$\mathbf{n}d = \frac{1}{T}\sum_{i=1}^{T}\boldsymbol{\mu}_t^{(l)} \tag{4}$$

$$\boldsymbol{\Sigma}_\iota = Var_{p(\mathtt{y}|\mathtt{x})}(\mathtt{y}) = \frac{1}{T}\sum_{i=1}^{T}\mathtt{v}_t^{(l)} + \frac{1}{T}\sum_{i=1}^{T}(\boldsymbol{\mu}_t^{(l)} - \overline{\boldsymbol{\mu}})^2.$$

V. Estimation with Probabilistic PlaneRCNN

A. Parameterization of planes in the measurement model

The plane normal is $\mathbf{n}$ with components (n_x, n_y, n_z) and the plane offset is d. As the plane normal is a unit vector, the plane offset can be extracted directly by finding the norm of $\mathbf{n}d$. Let us define a function $h(\cdot)$ that maps the plane normal from PlaneRCNN to a vector of spherical coordinates made up of the elevation (ψ) and azimuth (α) angles of the plane normal, and the inverse of the depth parameter, ρ (this is the inverse of the offset of plane centroid from the camera d) as follows:

$$\begin{bmatrix}\psi\\ \alpha\\ \rho\end{bmatrix} = h(\mathbf{n}, d) = \begin{bmatrix}\arctan(n_y/n_x)\\ \arctan(n_z/\sqrt{{n_x}^2 + {n_y}^2})\\ 1/d\end{bmatrix}. \tag{5}$$

A plane in space can be described completely by 3 independent variables so it has 3 DOF. Thus, the proposed spherical parameterization is a minimal representation and it is favourable because the angles do not need to be normalized after the update step in any filter based estimator, in our case IEKF.

Let us consider an inertial-aided system mapping point and planes whose state can be estimated by a probabilistic estimation framework, and the state is modelled as a multivariate Gaussian distribution $\mathcal{N}(\mathbf{x}, \mathbf{P})$. The state

$$\begin{aligned}\mathbf{x} = (\mathbf{r},\ \mathbf{v},\ \mathbf{q},\ \mathbf{b}_f, \mathbf{b}_w, \mathbf{p_c},\ \mathbf{q_c},\ \boldsymbol{\nu}_0, ...\boldsymbol{\nu}_Q,\ \beta_0, ...\beta_Q,\\ \psi_0, ...\psi_M,\ \alpha_0, ...\alpha_M,\ \rho_0, ...\rho_M),\end{aligned} \tag{6}$$

where, $\mathbf{r} \in \mathbb{R}^3$ is the position of the IMU in the world frame $\mathcal{W}$, $\mathbf{v} \in \mathbb{R}^3$ is the velocity of the IMU in $\mathcal{W}$, $\mathbf{q} \in \mathbb{SO}(3)$ is the orientation (parameterized as a unit quaternion and maps from $\mathcal{W}$ to the inertial frame $\mathcal{I}$), $\mathbf{b}_f$ is additive bias on accelerometer, $\mathbf{b}_w$ is additive bias on gyroscope, $\mathbf{p_c}$ and $\mathbf{q_c}$ are the linear and rotational part of extrinsic parameters between the IMU and camera, $\boldsymbol{\nu}_i$ is the bearing vector and β_i is the distance parameter to the point landmark i for the $Q+1$ point landmarks and ψ,α and ρ represent the spherical coordinated for the $M+1$ planes.

The spherical parameterization is applied to PlaneRCNN plane parameters $(n_x, n_y, n_z) * d$ transformed from $\mathcal{W}$ to the camera reference frame $\mathcal{C}$ according to:

$$\Omega(\mathbf{x}) \begin{cases}\mathbf{n}_c & = ((\mathbf{q}_{\mathcal{WI}} \otimes \mathbf{q}_{\mathcal{IC}}) \otimes \mathbf{n}_{\mathcal{W}} \otimes (\mathbf{q}_{\mathcal{WI}} \otimes \mathbf{q}_{\mathcal{IC}})^*)\\ d_c & = |\mathbf{n}_{\mathcal{W}}| d_{\mathcal{W}} - (\mathbf{n}_{\mathcal{W}} \cdot (\mathbf{q_c} \otimes \mathbf{r} \otimes \mathbf{q_c}^*) + \mathbf{p_c}),\end{cases} \tag{7}$$

where, $\cdot$ is the dot product operator, $\otimes$ is the quaternion product operator, $\mathbf{q}_{\mathcal{WI}}$ represents the rotation from $\mathcal{W}$ to $\mathcal{I}$, $\mathbf{q}_{\mathcal{IC}}$ represents the rotation from $\mathcal{I}$ to $\mathcal{C}$, $\mathbf{n}_{\mathcal{W}}$ is the plane normal in $\mathcal{W}$ and $\mathbf{n}_{\mathcal{C}}$ is the plane normal in $\mathcal{C}$.

We are interested in the complete measurement model $\mathbf{z}$, i.e., the composition of the coordinate frame transformation from $\mathcal{W}$ to $\mathcal{C}$, $\Omega(\cdot)$ and the transformation to spherical parameterization for plane normals $h(\cdot)$. Thus, the composite measurement model for planes is written as follows;

$$\mathbf{z} = h(\Omega(\mathbf{x})) + \boldsymbol{\iota}, \tag{8}$$

where $\boldsymbol{\iota} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\iota}})$ is the plane parameter uncertainty from probabilistic PlaneRCNN as per (4). In the Kalman Filter-based state estimation framework used (IEKF following [5]), a linearization of the non-linear composite measurement model is required for state and uncertainty propagation. We simply compute the measurement model Jacobian $\mathbf{H}$ by applying the chain rule of derivatives to the composite measurement model.

B. Conversion of uncertainty to new parameterization

The plane normals are initialized with the plane normal noise $\boldsymbol{\iota}$ (defined by the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\iota}}$) from PlaneRCNN as the initial covariance in the estimation filter. However, the plane normal uncertainty parameters from PlaneRCNN are parameterized in Cartesian coordinates. Since we apply a coordinate transformation to spherical planes parameterization to the plane normals, we multiply the plane normal uncertainty parameters with the Jacobians of the coordinate transformation.

C. Delayed Measurements from PlaneRCNN

There is some delay in the plane measurements introduced by the neural network inference time for each image. On average, the inference time per image is $1s$. Fusing these

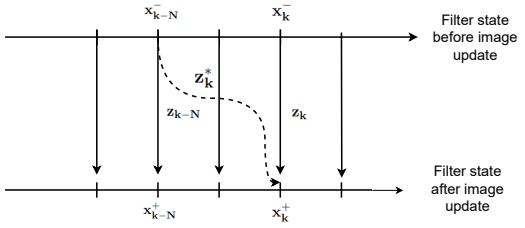

Fig. 3: System that incorporates N sample delay to the filtering state. Adapted from [25]

delayed measurements in a filter-based estimator is non-trivial and we implement it by using a measurement extrapolation method (see Fig 3) proposed by [25]. Let us modify the IEKF equations from ROVIO framework [5] for a non-linear discrete systems observed by undelayed measurements with additive Gaussian noise. To account for the delayed planar measurements, a second output equation for the plane measurement model in Eqn. 8 is formulated according to [25], as follows;

$$\mathbf{z}^*_k = h(\Omega(\mathbf{x}_{k-N})) + \boldsymbol{\iota}, \qquad \boldsymbol{\iota} \sim \mathcal{N}(0,\ \boldsymbol{\Sigma}^*_{\boldsymbol{\iota}}), \tag{9}$$

where N is the number of image data updates between the time where the planar measurements were obtained and the current filter state. The state at $k-N, \mathbf{x}_{k-N}$ is stored and it is used to calculate the residual that would have been obtained from time $k-N$ in the update at time k,

$$\mathbf{y}_k := \mathbf{y}_{k-N} = \mathbf{z}^*_k - h(\Omega(\mathbf{x}_k))\,. \tag{10}$$

This is equivalent to extrapolating the measurement $\mathbf{z}^*_k$ to a present measurement $\mathbf{z}^{int}_k$. $\mathbf{z}^*_k$ is fused at time k as follows:

$$\mathbf{z}^{int}_k = \mathbf{z}^*_k + h(\Omega(\mathbf{x}_k)) - h(\Omega(\mathbf{x}_{k-N})) \tag{11}$$

The new extrapolated measurements are used to derive the revised IEKF update equations:

$$\begin{aligned}\tilde{\mathbf{x}}^+_k = \prod_{i=0}^{N-1}(\mathbf{I} - \mathbf{K}_{k-i}\mathbf{H}_{k-i})\mathbf{F}_{k-i-1}\mathbf{x}^+_{k-N} + \\ \Lambda_1(\epsilon_{k-N}, ...\epsilon_{k-1}) + \Lambda_2(\iota_{k-N}, ...\iota_{k-1}),\end{aligned} \tag{12}$$

where $\tilde{\mathbf{x}}^+_k$ is the estimation error and Λ_1 and Λ_2 are functions of the noise sequence $\boldsymbol{\epsilon}$ and $\boldsymbol{\iota}$. The state after update $\mathbf{x}^+_k = \mathbf{x}_k + \tilde{\mathbf{x}}^+_k$. The state update equation is corrected as follows;

$$\begin{aligned}\mathbf{P}^+_{\mathbf{k}} &= \mathbf{P}_{\mathbf{k}} - \mathbf{K}_k\mathbf{H}^*_{k-N}\mathbf{P}_{k-N}\mathbf{M}^\top_* \\ \mathbf{K}_k &= \mathbf{M}_*\mathbf{P}_{k-N}\mathbf{H}^{*\top}_{k-N}[\mathbf{H}^*_{k-N}\mathbf{P}_{k-N}\mathbf{H}^{*\top}_{k-N} + \boldsymbol{\Sigma}^*_{\boldsymbol{\iota}_k}]^{-1} \\ \mathbf{M}_* &= \prod_{i=0}^{N-1}(\mathbf{I} - \mathbf{K}_{k-i}\mathbf{H}_{k-i})\mathbf{F}_{k-i-1}.\end{aligned} \tag{13}$$

The reader is referred to [25] for the full derivation of these equations. The value of N is calculated as the ratio of the image update rate to the inference rate of PlaneRCNN,

$$N = \text{image update rate/PlaneRCNN delay rate.} \tag{14}$$

D. Estimating Correspondences

The data association problem is about finding the correspondences between the measurements and the map features [26]. To decide whether the predicted measurement $h(\Omega(\mathbf{x}^-_{k-N}))$ and observed delayed plane measurement $\mathbf{z}_k$ belong to the same landmark, chi-square χ^2 distribution test is used as follows:

$$\begin{aligned}d^2_{\text{Maha}} = (\mathbf{z}_k - h(\Omega(\mathbf{x}^-_{k-N})))^\top[\mathbf{H}_{k-N}\mathbf{P}^-_{k-N}\mathbf{H}^\top_{k-N} + \boldsymbol{\Sigma}_{\boldsymbol{\iota}_k}]^{-1} \\ (\mathbf{z}_k - h(\Omega(\mathbf{x}^-_{k-N}))) < \chi^2_{n_d,\alpha}.\end{aligned} \tag{15}$$

where α is the confidence level and n_d is the degree of freedom which is 3 for our plane measurements. For our specific case, we chose a confidence level of 2σ.

VI. Active Mapping

An IPP framework is used to find the continuous optimal trajectory $\pi^*_{k:k+L}$, in the space of all trajectories Ψ for reducing mapping uncertainty,

$$\pi^*_{k:k+L} = \underset{\pi_{k:k+L}}{\text{argmin}}\ (\text{tr}(\mathbf{P}_{\mathbf{m}_{k+L}})), \tag{16}$$

where $L \in \mathbb{N}$ is the planning horizon, $\pi_{k:k+L}$ is the path from $t = t_k$ to $t = t_{k+L}$ and $\mathbf{P}_{\mathbf{m}}$ is the covariance matrix associated with the map, which we get by marginalizing the joint posterior covariance of the full state from the IEKF update step $\mathbf{P}^+_{\mathbf{k}}$ from Eqn 13. Rapidly-exploring random tree (RRT)* [27] is the sampling-based motion planning algorithm used to generate a set of nodes that are evaluated for information content in our framework.

Because the IMU requires continuous trajectories that are twice-differentiable in position and once differentiable in orientation to generate coherent measurements, we employ a tailor made interpolation method based on GP regression, which is presented in our previous work [4]. The interpolation method applies linear operators to the GP kernel function to infer not only continuous position trajectories, but also velocities and accelerations. Furthermore, linear functionals are used to enable velocity and acceleration constraints to be added in the GP model as part of the measurement vector.

The continuous trajectory and estimated map are used for simulating measurements which are used to compute the joint full state expected covariance. This joint expected covariance matrix is marginalized and the section associated with the map is used to determine the information content in a particular candidate trajectory based on the utility function I[$\cdot$] used to evaluate the expected reduction in the mapping uncertainty is defined as follows;

$$\text{I}(\boldsymbol{\pi}_{k:k+1}) = \text{tr}(\mathbf{P}_{\mathbf{m}_{k+1}}) - \text{tr}(\mathbf{P}_{\mathbf{m}_k}). \tag{17}$$

VII. Experiments and Results

In this section we discuss the evaluation and validation experiments for the proposed deep neural network. Hardware experiments are carried out in a texture-less scene to show the improved robustness of our proposed framework that uses both point and plane features in estimation as opposed to using only point features.

(a) (b) (c) (d) (e) (f)

Fig. 4: Evaluation of the probabilistic PlaneRCNN plane masks on the Machine Hall sequence 1 EuRoC Dataset and the UTS Mechatronics Lab Dataset.

A. Plane extraction capabilities

We assess qualitatively the plane extraction performance of the probabilistic PlaneRCNN. Three different setups are considered, two of which contain datasets recorded in our Labs at the UTS RI, and the third one is from the Machine Hall 1 sequence from the EuRoC datasets. The results of these experiments are shown in Fig. 4. The environment used in Fig 4a and 4b is the easiest since it has planar objects that fill up the entire image and contains no clutter, followed by Fig 4c and 4d which has some clutter and reflective glass surfaces and lastly Fig 4e and 4f are the hardest setup which is mostly cluttered with non-planar objects and reflective surfaces. For Fig 4a to Fig 4d, PlaneRCNN is able to accurately extract plane surfaces and with Fig 4e and Fig 4f, the performance the plane extraction though not as good as the previous four setups shown, is still acceptable. This experiment shows that the modification done to PlaneRCNN, of replacing all activations including the inputs, and outputs with probability distributions, does not affect the network's ability to extract planar feature from the environment.

B. Plane normals and uncertainties

The qualitative and quantitative assessment of the plane normals extracted by the probabilistic PlaneRCNN is carried out in this section. We compare the plane normal parameters extracted by PlaneRCNN against the plane normal parameters extracted by another model fitting algorithm using RANSAC. We then compare the difference in angle between the 2 plane normal vectors extracted by the two algorithms.

Dataset	Average angle between normals [radians]	% of RANSAC parameters within $2-\sigma$ bounds
Machine Hall	0.1558	72
UTS Dataset 2	0.1138	76
UTS Dataset 1	0.0592	75

TABLE I: Validating plane normals and uncertainties extracted by the probabilistic PlaneRCNN with RANSAC plane normals.

The 2^{nd} column of Table I shows the average over 30 plane normals considered for each of the three datasets discussed in Sec. VII-A. The average angle difference between PlaneRCNN normals and RANSAC normals for all the datasets considered is below 0.1745 radians (10 degrees), showing that the plane normals extracted by PlaneRCNN are comparable to those from RANSAC model within reason. In addition, Fig. 5 qualitatively shows plane masks drawn in 3D, the 3D mask centroid and the plane normals for both PlaneRCNN and RANSAC. It can be seen that the plane normals extracted using these 2 methods are comparable.

We also validate the plane uncertainty parameters from the probabilistic PlaneRCNN algorithm. The plane uncertainty

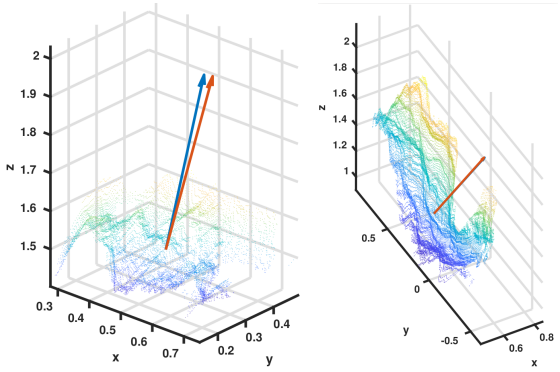

Fig. 5: Planes masks in 3D, the 3D mask centroid and plane normals for PlaneRCNN in blue and RANSAC in orange.

are used to calculate 2 sigma-bound values for the plane normal predictions. We then compare to see whether the plane normal parameters from RANSAC model fall within the 2 sigma-bounds of the plane normal parameters we obtain from PlaneRCNN. The 3^{rd} column of Table I shows the percentage of plane parameters from RANSAC can that fall within 2 sigma-bounds of the plane parameter values from our modified PlaneRCNN version. For all the 3 datasets considered, the value is greater than (72%) which shows that the plane uncertainty parameters from the probabilistic PlaneRCNN are acceptable.

C. Probabilistic normals vs deterministic normals

In this section we validate the use of probabilistic planes from our modified version of PlaneRCNN vs using deterministic planes extracted by RANSAC initialized with a fixed covariance value in the estimation framework. For the two methods considered, we fix the trajectories and mapping environment and change the uncertainty parameters used to initialize $\mathbf{P_m}$ in the estimation framework. 10 Monte Carlo run is conducted and the average trace of $\mathbf{P_m}$ is tabulated in Table II. The results of this experiment demonstrate the superiority of using probabilistic estimates of PlaneRCNN for mapping in the IEFK framework, as shown by lower average mapping uncertainty values.

Type of planes used	Average mapping uncertainty
PlaneRCNN probabilistic planes	**8.69**
RANSAC deterministic planes	23.18

TABLE II: Comparison of probabilistic plane normals vs deterministic plane normals on overall mapping uncertainty.

D. Hardware experiments

In this section we discuss the hardware experiment conducted. A 6-DOF UR5 arm is used for executing the informative trajectories from our planner and the camera used is

the Realsense D455 with its internal Bosch BMI055 IMU. The camera provides global shutter RGB images at 30Hz and IMU measurements at 200Hz.

We constrain the trajectories to be executed on the robot arm, to those trajectories for which a valid inverse kinematics solution exists for each pose. We also require that the joint limits of the robot are not violated and the robot does not collide with itself or the environment. We leverage the HAP [28], to enable fast and direct sampling of trajectories in $\mathbb{SE}(3)$ which, given a robot, task-space and environment model, computes a subspace in $\mathbb{SE}(3)$ to sample from such that the resulting executed robot trajectory satisfies our desired constraints. This subspace is represented using a discrete roadmap of poses, and the roadmap is provided to the RRT* variant planner to bias its sampling towards.

The sampled trajectories are post-processed and ensured to be within the provided subspace by checking for time-continuous safety and any large changes in arm configuration between two consecutive poses. If either of these conditions are violated, the trajectory is discarded. In practice it was found that a majority of trajectories were within the subspace and not discarded owing to the robustness of the planner.

To test the improvement in terms of robustness, for the estimation algorithm, we run the experiment in an environment where limited point features can be extracted but is rich in plane features. The setup of the area used is shown in Fig. 1a and 1b. The state, feature map and the associated covariance matrix are estimated as described in Section V after execution of our trajectories on the arm. We compare the mapping uncertainty for the approach that uses point features and the approach with point and plane features. Both experiments have the same number of features, so the size of the map covariance is the same. Fig. 6 shows a plot comparing the average mapping uncertainty when the two approaches are run over 3 Monte Carlo runs each. In this plot, it can be seen that the mapping uncertainty using points is greater than the one using both points and planes for estimation. The spikes in the red plot observed at around 11 trajectory segments, also seen around 17 and 23 trajectory segments were observed during those instances when very few or none point features were seen by the filter.

In texture-less environments, the estimation algorithm suffers substantially as it does not observe sufficient enough point features to reduce the feature covariance during update events. However, the algorithm that uses both points and planes features observes plane features and these help reduce mapping uncertainty as shown in Fig. 6. For the plots, we only considered those experiments where both scenarios had the same number of trajectory segments without diverging.

The video shows the informative trajectory our planner generates for active mapping objective in texture-less environments. From times 4 to 11 seconds in the video, the camera is facing the floor with no point features extracted except the ground plane. This one plane only prevents diverging. In the setup where only points features are considered, the estimation diverges when features are not observed for periods greater than 5 seconds.

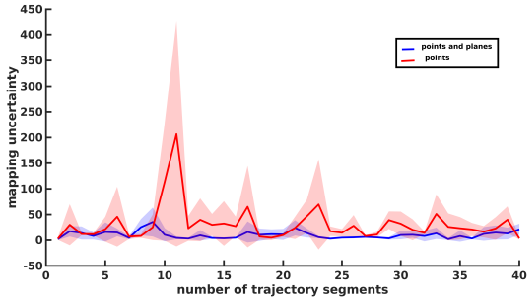

Fig. 6: Average mapping uncertainty, with error bounds of the estimation with points only and with points and planes.

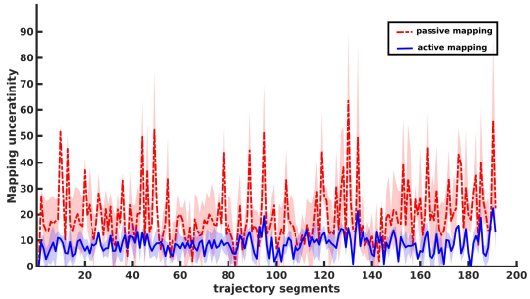

Fig. 7: Comparison of mapping uncertainty for an active vs passive mapping approach.

1) Comparison of active mapping vs passive mapping: This experiment evaluates the impact that active planning has on the overall mapping uncertainty versus a passive planning approach. The RRT* algorithm is used to explore the environment to find a trajectory that reduces the mapping uncertainty the most by using the cost function in Eqn. 17. This trajectory is compared against preset trajectories. For both methods, the same number of trajectory segments are added to the trajectory and a 10 Monte Carlo run is conducted. The results of this experiment are shown in Fig 7, where the blue solid line represents the runs' average for active mapping while the red dashed line represents the passive mapping average. The shaded area represents the respective 2σ bounds. The plots show that active mapping outperforms the passive one. At the end of the 200 segment trajectories, the average trace of $\mathbf{P_m}$ is 10.17 in the active mapping while the passive approach has a trace of 27.36. This is consistent with what is expected because the RRT* algorithm guarantees asymptotic optimality using the rewiring step.

VIII. Conclusion

We introduced a novel probabilistic plane extraction and modelling framework that allows the extraction of plane parameters and their associated uncertainties based on Plane-RCNN. The extracted probabilistic plane features are fused with point features in order to increase the robustness of a visual-inertial active mapping system in texture-less environments, where estimation algorithms based on points alone struggle. A customized IEKF is proposed that enables the estimation to handle the delay introduced by the neural network inference. Extensive experiments carried out in low-texture scenes show improved robustness of our proposed framework using both point and plane features in estimation, as opposed to using only point features.

REFERENCES

[1] C. Liu, K. Kim, J. Gu, Y. Furukawa, and J. Kautz, "Planercnn: 3d plane detection and reconstruction from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4450–4459.

[2] C. Stachniss, *Robotic mapping and exploration*. Springer, 2009, vol. 55.

[3] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *arXiv preprint arXiv:2207.00254*, 2022.

[4] M. Usayiwevu, F. Sukkar, C. Yoo, R. Fitch, and T. Vidal-Calleja, "Continuous planning for inertial-aided systems," *arXiv preprint arXiv:2209.05285*, 2022.

[5] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 298–304.

[6] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[7] L. Jinyu, Y. Bangbang, C. Danpeng, W. Nan, Z. Guofeng, and B. Hujun, "Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 4, pp. 386–410, 2019.

[8] P.-H. Le and J. Košecka, "Dense piecewise planar rgb-d slam for indoor environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4944–4949.

[9] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 2100–2106.

[10] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[11] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, 2021.

[12] J. Li, X. Zhou, B. Yang, G. Zhang, X. Wang, and H. Bao, "Rlp-vio: Robust and lightweight plane-based visual-inertial odometry for augmented reality," *Computer Animation and Virtual Worlds*, p. e2046, 2022.

[13] M. Hsiao, E. Westman, G. Zhang, and M. Kaess, "Keyframe-based dense planar SLAM," in *Proc. IEEE Intl. Conf. on Robotics and Automation, ICRA*, Singapore, May 2017, pp. 5110–5117.

[14] P. Kim, B. Coltin, and H. J. Kim, "Linear rgb-d slam for planar environments," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 333–348.

[15] X. Li, Y. Li, E. P. Örnek, J. Lin, and F. Tombari, "Co-planar parametrization for stereo-slam and visual-inertial odometry," *IEEE Robotics and Automation Letters*, 2020.

[16] K. Ram, C. Kharyal, S. S. Harithas, and K. M. Krishna, "Rp-vio: Robust plane-based visual-inertial odometry for dynamic environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 9198–9205.

[17] K. Lindgren, S. Leung, W. D. Nothwang, and E. J. Shamwell, "Boom-vio: Bootstrapped monocular visual-inertial odometry with absolute trajectory estimation through unsupervised deep learning," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 516–522.

[18] N. Fairfield and D. Wettergreen, "Active slam and loop prediction with the segmented map using simplified models," in *Field and service robotics*. Springer, 2010, pp. 173–182.

[19] B. Mu, M. Giamou, L. Paull, A.-a. Agha-mohammadi, J. Leonard, and J. How, "Information-based active slam via topological feature graphs," in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 5583–5590.

[20] H. Qin, Z. Meng, W. Meng, X. Chen, H. Sun, F. Lin, and M. H. Ang, "Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1339–1350, 2019.

[21] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 1. IEEE, 2002, pp. 540–545.

[22] A. Loquercio, M. Segu, and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3153–3160, 2020.

[23] X. Boyen and D. Koller, "Tractable inference for complex stochastic processes," *arXiv preprint arXiv:1301.7362*, 2013.

[24] J. Gast and S. Roth, "Lightweight probabilistic deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3369–3378.

[25] T. D. Larsen, N. A. Andersen, O. Ravn, and N. K. Poulsen, "Incorporation of time delayed measurements in a discrete-time kalman filter," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, vol. 4. IEEE, 1998, pp. 3972–3977.

[26] T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.

[27] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, no. 2, 2010.

[28] F. Sukkar, J. Wakulicz, K. M. B. Lee, and R. Fitch, "Motion planning in task space with gromov-hausdorff approximations," *arXiv preprint arXiv:2209.04800*, 2022.