Monocular Visual-Inertial Odometry with Planar Regularities

Chuchu Chen*, Patrick Geneva*, Yuxiang Peng, Woosik Lee, and Guoquan Huang

***Abstract*— State-of-the-art monocular visual-inertial odometry (VIO) approaches rely on sparse point features in part due to their efficiency, robustness, and prevalence, while ignoring high-level structural regularities such as planes that are common to man-made environments and can be exploited to further constrain motion. Generally, planes can be observed by a camera for significant periods of time due to their large spatial presence and thus, are amenable for long-term navigation. Therefore, in this paper, we design a novel real-time monocular VIO system that is fully regularized by planar features within a lightweight multi-state constraint Kalman filter (MSCKF). At the core of our method is an efficient robust *monocular*-based plane detection algorithm, which does *not* require additional sensing modalities such as a stereo or depth camera as commonly seen in the literature, while enabling real-time regularization of point features to environmental planes. Specifically, in the proposed MSCKF, long-lived planes are maintained in the state vector, while shorter ones are marginalized after use for efficiency. Planar regularities are applied to both in-state SLAM features and out-of-state MSCKF features, thus fully exploiting the environmental plane information to improve VIO performance. The proposed approach is evaluated with extensive Monte-Carlo simulations and different real-world experiments including an author-collected AR scenario, and shown to outperform the point-based VIO in structured environments.**

Video Demonstration
https://youtu.be/bec7LbYaOS8
AR Table Dataset
https://github.com/rpng/ar_table_dataset

I. Introduction and Related Work

Visual-inertial odometry (VIO) that fuses IMU and camera measurements to provide efficient 3D motion tracking, has emerged as a foundational technology for AR/VR applications [1]–[3], primarily thanks to its low-energy, small-size, low-cost, and complementary sensing characteristics. Substantial research efforts both in industry and academia have recently been devoted to VIO algorithms [4], which can be categorized broadly into optimization-based and filter-based methods. The former formulates a nonlinear least-squares (NLS) problem with all available measurements and iteratively finds an accurate solution at a higher computational cost due to relinearization [5]–[8]. In contrast, filter-based estimators such as the multi-state constraint Kalman filter (MSCKF) [9] remain popular in resource-constrained platforms due to their efficiency [10]–[15]. In particular, the MSCKF processes available visual bearing measurements without keeping the point features in their state vector, which is achieved by projecting the observations onto the nullspace of the corresponding feature measurement Jacobian and inferring feature-independent measurement residuals for EKF update (i.e., linear marginalization [16]). Albeit, it can selectively include SLAM point features in the state – which are marginalized once lost – in order to exploit local temporal map (or loop closure) information while still bounding computational cost [17]–[20].

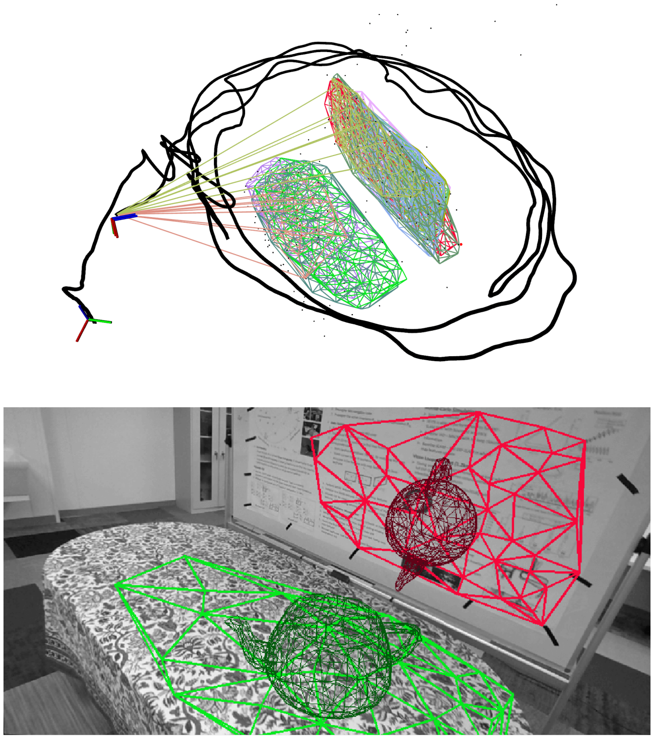

Fig. 1: Environmental reconstruction example on the author-collected AR table_06 (top) along with an AR display render of two teapots on estimated planes (bottom).

While the state-of-the-art MSCKF-based monocular VIO approaches have efficient and robust point feature detection and tracking mechanisms and achieve high-accuracy performance [20], they are unable to utilize structural regularity information (if any) between features, which could prevail in man-made environments. Ideally, one could leverage pixel-level dense depths to enforce structural constraints, but this it is computationally expensive due to the large number of optimization variables (depth map) [21]–[25]. Given stringent resources, it is challenging to extract high-level geometric features such as planes from monocular images and reason about inter-state structural regularities onto low-level point

*These authors contributed equally to the work.

This work was partially supported by the University of Delaware (UD) College of Engineering, the NSF (IIS-1924897, SCH-2014264, CNS-2018905), and Google ARCore. Geneva is also partially supported by the NASA DE Space Grant Graduate Fellowship.

The authors are with the Robot Perception and Navigation Group (RPNG), University of Delaware, Newark, DE 19716, USA. Email: {ccchu,pgeneva,yxpeng,woosik,ghuang}@udel.edu

features. As such, existing literature has primarily focused on explicitly detecting line and plane features with stereo or depth sensors [26]–[29].

In particular, many methods have leveraged line features in conjunction with Manhattan [30] or Atlanta world [31] regularities, improving accuracy due to structured lines (e.g., aligned with building cardinal directions) that directly provide global attitude information [32]–[37]. As environmental planes cannot be directly detected with a single monocular camera since the depth is unavailable, generic depth sensors that can directly measure environmental planes – such as RGB-D [25], [26], [38], [39], 1D LRF [40], [41], or 3D LiDAR [42]–[44] – have been fused with great success. Similarly to line features, planar Manhattan frames have been leveraged with success [45], [46]. Additionally, some works have enforced cross-plane orthogonality, parallelism [36], [38], or point-on-plane regularities [39], but require an additional sensor which increases cost, computation, calibration complexity, and data association challenges. Recently, deep-learning-based methods have become of interest due to their ability to perform single-shot detection of planar surfaces and normals [47]–[50]. For example, RP-VIO [51] leverages a plane segmentation network [52] to separate planar surfaces which are assumed to be static within a dynamic environment and enforce point-on-plane camera homography constraints. While this direction is promising, it typically requires additional computational resources and its generalizability is unclear.

Closest to our work, which leverages planar structural regularities, is that by Rosinol et al. [53]–[55]. They proposed a *stereo* VIO system that incrementally builds and estimates 3D meshes (planes) in-which point-on-plane structural regularities are enforced during optimization. They have shown that the inclusion of planar regularities improves both state estimation and environmental mesh accuracy. This plane detection method was extended to include lines within the monocular VINS-Mono [7] framework in PLP-VIO [56], which additionally enforced point-to-line and line-to-plane regularities. Both *only* enforce structural regularities for vertical and horizontal planes (with respect to gravity), require the inclusion of planes in the state (increasing computation), and may experience significant computational spikes when the number of constraints grows.

This paper presents a new real-time *monocular* MSCKF-based VIO system that efficiently extracts and enforces structural regularities from environmental planes without requiring an additional depth sensor or neural network.

The main contributions of our work include:

- We design an efficient monocular VIO estimator that detects and enforces planar regularities through point-on-plane geometric constraints, which is able to either estimate long-lived SLAM plane features or directly marginalize short-lived ones for efficiency. We also investigate the planar regularization for both in-state SLAM and out-of-state MSCKF point features.
- We develop a novel and robust plane detection and tracking algorithm, which exploits pairwise comparisons of sparse VIO point feature norms and enables real-time estimation without costly dense depth maps or neural network computations.
- We validate the proposed system extensively in both Monte-Carlo simulations and real-world experiments and also release the author-collected datasets for the benefit of the community.

II. Overview of the Proposed System

In this section, we overview the proposed real-time monocular MSCKF-VIO system termed `ov_plane`, which fuses IMU readings and sparse environmental 3D point bearings. Algorithm 1 outlines the main steps of the proposed approach, whose key idea is to take advantage of the (typically) large spatial nature of planar structures to lengthen feature tracks and thus further improve estimation accuracy. To the best of our knowledge, this is the first time that a monocular-VIO estimator is able to rigorously enforce planar regularities within the MSCKF framework.

A. State Vector

At time t_k, the system state $\mathbf{x}_k$ consists of the current navigation states $\mathbf{x}_{I_k}$, historical IMU pose clones $\mathbf{x}_C$, and a subset of 3D environmental (SLAM) point features, $\mathbf{x}_f$, and (SLAM) plane features, $\mathbf{x}_\pi$:

$$\mathbf{x}_k = \left[\mathbf{x}_{I_k}^\top \ \mathbf{x}_C^\top \ \mathbf{x}_f^\top \ \mathbf{x}_\pi^\top\right]^\top, \ \ \mathbf{x}_C = \left[\mathbf{x}_{T_k}^\top \dots \mathbf{x}_{T_{k-c}}^\top\right]^\top \quad (1)$$

$$\mathbf{x}_{I_k} = \left[{}^{I_k}_G\bar{q}^\top \quad {}^G\mathbf{p}_{I_k}^\top \quad {}^G\mathbf{v}_{I_k}^\top \quad \mathbf{b}_g^\top \quad \mathbf{b}_a^\top\right]^\top \quad (2)$$

$$\mathbf{x}_f = \left[{}^G\mathbf{p}_{f_1}^\top \dots {}^G\mathbf{p}_{f_g}^\top\right]^\top, \ \ \mathbf{x}_\pi = \left[{}^G\mathbf{\Pi}_1^\top \dots {}^G\mathbf{\Pi}_h^\top\right]^\top \quad (3)$$

where ${}^I_G\bar{q}$ is the unit quaternion (${}^I_G\mathbf{R}$ in rotation matrix form) that represents the rotation from the global $\{G\}$ to the IMU frame $\{I\}$; ${}^G\mathbf{p}_I$, ${}^G\mathbf{v}_I$, and ${}^G\mathbf{p}_{f_i}$ are the IMU position, velocity, and i'th point feature position in $\{G\}$; $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases; $\mathbf{x}_{T_i} = [{}^{I_i}_G\bar{q}^\top \ {}^G\mathbf{p}_{I_i}^\top]^\top$. We represent each plane, ${}^G\mathbf{\Pi}$, in the global frame with the minimal error state *closest point* (CP) representation, which can be defined using the plane's normal vector ${}^G\mathbf{n}$ and distance scalar ${}^Gd$ as ${}^G\mathbf{\Pi} = {}^G\mathbf{n}{}^Gd$ [57].

B. Propagation with IMU Kinematics

The inertial kinematics are used to evolve the state from time t_k to t_{k+1} [58], [59]:

$$\mathbf{x}_{I_{k+1}} = \mathbf{f}(\mathbf{x}_{I_k}, \mathbf{a}_{m_k}, \boldsymbol{\omega}_{m_k}) \quad (4)$$

where the linear acceleration $\mathbf{a}_{m_k}$ and the angular velocity $\boldsymbol{\omega}_{m_k}$ measurements are contaminated by zero-mean white Gaussian noises. The MSCKF linearizes this nonlinear model and propagates forward the state estimate and covariance [9].

C. Non-Planar Point Feature Update

The camera provides bearing observations of environmental 3D points. These observations can be used to update our state using the following measurement function (note that we here assume the global 3D feature model [20]):

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k =: \mathbf{\Lambda}({}^{C_k}\mathbf{p}_f) + \mathbf{n}_k \quad (5)$$

$${}^{C_k}\mathbf{p}_f = [x \ y \ z]^\top = {}^C_I\mathbf{R}{}^{I_k}_G\mathbf{R}\left({}^G\mathbf{p}_f - {}^G\mathbf{p}_{I_k}\right) + {}^C\mathbf{p}_I \quad (6)$$

$$\mathbf{\Lambda}([x \ y \ z]^\top) =: [x/z \ y/z]^\top \quad (7)$$

where $\mathbf{n}_k$ is the white Gaussian bearing measurement noise and $\{{}^C_I\mathbf{R}, {}^C\mathbf{p}_I\}$ is the camera-IMU rigid transformation. Linearizing Eq. (5) yields the following system:

$$\tilde{\mathbf{z}}_k \simeq \mathbf{H}_k\tilde{\mathbf{x}}_k + \mathbf{n}_k = \mathbf{H}_{T_k}\tilde{\mathbf{x}}_{T_k} + \mathbf{H}_{f_k}{}^G\tilde{\mathbf{p}}_f + \mathbf{n}_k \tag{8}$$

where $\mathbf{H}_{T_k}$ and $\mathbf{H}_{f_k}$ are the measurement Jacobians in respect to the observing pose $\hat{\mathbf{x}}_{T_k}$ and 3D point feature ${}^G\hat{\mathbf{p}}_f$.[1] We can then "stack" the measurements from different timesteps to get:

$$\tilde{\mathbf{z}}_c = \mathbf{H}^c_T\tilde{\mathbf{x}}_C + \mathbf{H}^c_f{}^G\tilde{\mathbf{p}}_f + \mathbf{n}_c \tag{9}$$

where $\tilde{\mathbf{z}}_c$ is the stacked measurement residual; $\mathbf{H}^c_T$ and $\mathbf{H}^c_f$ are the stacked Jacobians; $\mathbf{n}_c \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_c)$ is the stacked measurement noise (normally 1 pixel). We then perform EKF update with two different types of non-planar point features:

- *SLAM Point:* The state $\mathbf{x}_f$ contains ${}^G\mathbf{p}_f$, thus Eq. (9) can directly update the state using the standard EKF equations.
- *MSCKF Point:* For features that are not in the state we project Eq. (9) onto the left nullspace of $\mathbf{H}^c_f$ (i.e., $\mathbf{N}_f^\top\mathbf{H}^c_f = \mathbf{0}$ [9]). Specifically, we can construct the following system which is independent of ${}^G\tilde{\mathbf{p}}_f$:

$$\mathbf{N}_f^\top\tilde{\mathbf{z}}_c = \mathbf{N}_f^\top\mathbf{H}^c_T\tilde{\mathbf{x}}_C + \mathbf{N}_f^\top\mathbf{H}^c_f{}^G\tilde{\mathbf{p}}_f + \mathbf{N}_f^\top\mathbf{n}_c \tag{10}$$

$$\Rightarrow \tilde{\mathbf{z}}'_c = \mathbf{H}'_T\tilde{\mathbf{x}}_C + \mathbf{n}' \tag{11}$$

  This reduces the filter's computational complexity since the feature does not need to be inserted into the state.

In the next section, we will present in detail how the proposed `ov_plane` performs update with planar regularities.

III. Planar Regularities

At the core of the proposed `ov_plane` system are the planar regularities. In the following, we explain how to perform MSCKF update with planar regularities, while addressing practical challenges, which include efficient point-feature updates constrained by (in-state and out-of-state) planes, and robust initialization of plane features that are augmented into the state. For clarity, we refer the reader to Sec. V-A for the proposed real-time extraction and robust matching of planes from sparse visual points.

A. Regularization-Constrained Measurement

The proposed VIO system enforces planar regularities through point-on-plane constraints. Consider a point feature ${}^G\mathbf{p}_f$ that lies on the plane ${}^G\mathbf{\Pi}$, we have:

$$z_d = \left({}^G\mathbf{p}_f^\top{}^G\mathbf{n} - {}^Gd\right) + \sigma_d \tag{12}$$

where σ_d is the noise that softens the constraint and should be zero in the ideal case [60]. We linearized Eq. (12) to get:

$$\tilde{z}_d = \mathbf{H}^d_f{}^G\tilde{\mathbf{p}}_f + \mathbf{H}^d_\pi{}^G\tilde{\mathbf{\Pi}} + \sigma_d \tag{13}$$

We then stack the point feature bearing observation model, Eq. (8), and point-on-plane constraint, Eq. (13):

$$\begin{bmatrix}\tilde{\mathbf{z}}_c \\ \tilde{z}_d\end{bmatrix} = \begin{bmatrix}\mathbf{H}^c_T \\ \mathbf{0}\end{bmatrix}\tilde{\mathbf{x}}_C + \begin{bmatrix}\mathbf{H}^c_f \\ \mathbf{H}^d_f\end{bmatrix}{}^G\tilde{\mathbf{p}}_f + \begin{bmatrix}\mathbf{0} \\ \mathbf{H}^d_\pi\end{bmatrix}{}^G\tilde{\mathbf{\Pi}} + \begin{bmatrix}\mathbf{n}_c \\ \sigma_d\end{bmatrix} \tag{14}$$

$$\Rightarrow \tilde{\mathbf{z}} = \mathbf{H}_T\tilde{\mathbf{x}}_C + \mathbf{H}_f{}^G\tilde{\mathbf{p}}_f + \mathbf{H}_\pi{}^G\tilde{\mathbf{\Pi}} + \mathbf{n} \tag{15}$$

$$= \mathbf{H}_x\tilde{\mathbf{x}}_k + \mathbf{H}_\pi{}^G\tilde{\mathbf{\Pi}} + \mathbf{n} \tag{16}$$

where $\mathbf{H}_T$, $\mathbf{H}_f$, and $\mathbf{H}_\pi$ are the Jacobians for the IMU poses, point feature, and plane feature, respectively; $\mathbf{H}_x = [\mathbf{H}_T \;\; \mathbf{H}_f]$ and $\tilde{\mathbf{x}}_k = [\tilde{\mathbf{x}}_C^\top \;\; \tilde{\mathbf{x}}_f^\top]^\top$; and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ denotes the measurement noise after *whitening*.

Algorithm 1 `ov_plane`

```
Propagation:
  • Propagate the state vector and covariance with inertial
    readings [see Sec. II-B]
Feature Tracking:
  • Extract visual features from the image, then perform sparse
    KLT tracking and outlier rejection.
  • Formulate a 2D Delaunay triangle mesh, detect, and match
    planes [Sec. V-A]
State Management:
  • Initialize SLAM point and plane features into the state if
    sufficient observations / features [Sec. III-C]
  • Merge planes if needed [Sec. III-D]
  • Marginalize SLAM point and plane features from the state
    when tracking is lost
Update:
  • Update non-plane points [Eq. (9), (11)]
  • Update MSCKF plane (out-of-state)
    – Recover points and plane, then jointly refine their
      estimates [Sec. III-B]
    – Nullspace project $\mathbf{H}_\pi$ pre-update [Eq. (22), (24)]
  • Update SLAM plane (in-state)
    – SLAM points directly update planes [Eq. (16)]
    – MSCKF points are projected onto their $\mathbf{H}_f$ nullspace
      before update [Eq. (19)]
```

B. Plane Recovery and Non-Linear Refinement

To enforce point-on-plane constraint, Eq. (13), we first robustly recover the initial guess for the plane by performing RANSAC [61] on a set of co-planar point features (details on how we extract co-planar sets are in Algorithm 2). A plane estimate can be solved from at least three points with the following linear system:

$$\begin{bmatrix}\cdots & {}^G\mathbf{p}_{f,i} & \cdots\end{bmatrix}^\top \boldsymbol{\pi} = \begin{bmatrix}\cdots & 1 & \cdots\end{bmatrix}^\top \tag{17}$$

After obtaining $\boldsymbol{\pi}$, the plane can be recovered by ${}^G\mathbf{n} = \boldsymbol{\pi}/||\boldsymbol{\pi}||$ and ${}^Gd = 1/||\boldsymbol{\pi}||$. The RANSAC inlier set is selected based on the point-to-plane distance threshold, see Eq. (12), with the best-recovered plane having the most inliers and smallest average point-to-plane distance.

If a sufficient number of inliers are found, we perform a *joint* refinement of the point features and plane with fixed camera poses. SLAM points that lie on the plane are fixed during optimization but are included to further improve the plane estimate through their point-on-plane constraints. The non-linear optimization problem is formulated using Eq. (5) and (12) and is solved using ceres-solver [62] that takes 0.5-1.5 milliseconds (ms).

C. Plane Feature Initialization

We wish to initialize long-tracked planes into states, which offer dependable regularization information and constrain a large number of co-planar feature points.

For an MSCKF planar point feature, in analogy to MSCKF feature marginalization, we project Eq. (15) onto the the left

[1] Throughout the paper $\hat{\mathbf{x}}$ is used to denote the *current* estimate of a random variable $\mathbf{x}$ with $\tilde{\mathbf{x}} = \mathbf{x}\boxminus\hat{\mathbf{x}}$ denotes the error state. For the quaternion error state, we employ JPL multiplicative error [59] i.e., $\delta\bar{q} = \bar{q}\otimes\hat{\bar{q}}^{-1} \simeq [\frac{1}{2}\delta\boldsymbol{\theta}^\top \;\; 1]^\top$.

TABLE I: Simulation parameters and prior standard deviations that perturbations of measurements were drawn from.

Parameter	Value	Parameter	Value
Gyro. White Noise	1.6968e-04	Gyro. Rand. Walk	1.9393e-05
Accel. White Noise	2.0000e-3	Accel. Rand. Walk	3.0000e-3
Cam Freq. (hz)	10	IMU Freq. (hz)	400
Num. Clones	11	Total Planes	6
Avg. Feats on Plane	150	Max SLAM Pts	15

nullspace of $\mathbf{H}_f$ (i.e., $\mathbf{N}^\top \mathbf{H}_f = \mathbf{0}$) to get a residual function for plane ${}^G\mathbf{\Pi}$ that is independent to the point feature:

$$\mathbf{N}^\top \tilde{\mathbf{z}} = \mathbf{N}^\top \mathbf{H}_T \tilde{\mathbf{x}}_C + \mathbf{N}^\top \mathbf{H}_\pi {}^G\tilde{\mathbf{\Pi}} + \mathbf{N}^\top \mathbf{n} \tag{18}$$

$$\Rightarrow \quad \tilde{\mathbf{z}}^* = \mathbf{H}_x^* \tilde{\mathbf{x}}_k + \mathbf{H}_\pi^* {}^G\tilde{\mathbf{\Pi}} + \mathbf{n}^* \tag{19}$$

For a SLAM point feature, Eq. (16) can be directly used.

After collecting co-planar MSCKF [Eq. (19)] and SLAM [Eq. (16)] point feature measurements, we stack them into the following linear system:

$$\begin{matrix} \text{MSCKF}: \\ \text{SLAM}: \end{matrix} \begin{bmatrix} \tilde{\mathbf{z}}^* \\ \tilde{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_x^* \\ \mathbf{H}_x \end{bmatrix} \tilde{\mathbf{x}}_k + \begin{bmatrix} \mathbf{H}_\pi^* \\ \mathbf{H}_\pi \end{bmatrix} {}^G\tilde{\mathbf{\Pi}} + \begin{bmatrix} \mathbf{n}^* \\ \mathbf{n} \end{bmatrix} \tag{20}$$

where $\tilde{\mathbf{z}}^*$ and $\tilde{\mathbf{z}}$ represent the MSCKF and SLAM point feature measurement residuals, respectively. We then leverage the method in [20], [63] to initialize ${}^G\mathbf{\Pi}$ into the state.

D. Plane Merging

Planes are initialized into the state as soon as there are sufficient observations, and they are continually refined with new measurements. Over time, multiple planes may merge into a single common plane to reduce redundancy in the state. If more than one planes reside in the state, a pair-wise state constraint update is performed to enforce equality, then followed by the marginalization of all but one plane. For example, plane ${}^G\mathbf{\Pi}_1$ and ${}^G\mathbf{\Pi}_2$ have:

$$\mathbf{z}_p = \left({}^G\mathbf{\Pi}_2 - {}^G\mathbf{\Pi}_1\right) + \mathbf{n}_p \tag{21}$$

where $\mathbf{n}_p$ is the small noise that softens the constraint [60].

E. Planar Point Feature Update

Given the planar point feature linearized measurement function, Eq. (14), we will explain in detail how to process measurements with in-state or out-of-state features. (see Sec. II for non-planar point features). We then consider the following update methods:

- *SLAM Plane + SLAM Point:* Standard EKF update
- *SLAM Plane + MSCKF Point:* Remove the point feature dependency through nullspace projection [see Eq. (18)].
- *MSCKF Plane + SLAM Point:* Remove the plane feature dependency by:
  $$\mathbf{N}_\pi^\top \tilde{\mathbf{z}}_A = \mathbf{N}_\pi^\top \mathbf{H}_T \tilde{\mathbf{x}}_T + \mathbf{N}_\pi^\top \mathbf{H}_f {}^G\tilde{\mathbf{p}}_f + \mathbf{N}_\pi^\top \mathbf{n} \tag{22}$$
  where $\mathbf{N}_\pi$ is the left nullspace of the stacked $\mathbf{H}_\pi$. This requires more than 3 planar point features.
- *MSCKF Plane + MSCKF Point:* Remove the plane and point feature dependency by:
  $$\tilde{\mathbf{z}} = \mathbf{H}_T \tilde{\mathbf{x}}_T + \begin{bmatrix} \mathbf{H}_f & \mathbf{H}_\pi \end{bmatrix} \begin{bmatrix} {}^G\tilde{\mathbf{p}}_f \\ {}^G\tilde{\mathbf{\Pi}} \end{bmatrix} + \mathbf{n} \tag{23}$$
  $$\Rightarrow \mathbf{N}_{f\pi}^\top \tilde{\mathbf{z}} = \mathbf{N}_{f\pi}^\top \mathbf{H}_T \tilde{\mathbf{x}}_T + \mathbf{N}_{f\pi}^\top \mathbf{n} \tag{24}$$
  where $\mathbf{N}_{f\pi}$ is the left nullspace of $[\mathbf{H}_f \quad \mathbf{H}_\pi]$. Observing a feature more than three times is necessary.

TABLE II: Average 20 run RPE and NEES for different algorithm configurations. Units are in degrees / cm. A constraint noise of $\sigma_d = 0.001$ was used. M corresponds to MSCKF features (out-of-state), S for SLAM features (in-state), PT for point features, and PL represents plane features.

Algorithm	60m	80m	100m	120m	NEES(3)
M-PT	0.37 / 4.3	0.44 / 5.0	0.50 / 5.6	0.55 / 6.2	3.39 / 1.75
M-PT & M-PL	0.37 / 4.3	0.43 / 4.9	0.48 / 5.5	0.53 / 6.1	3.34 / 1.72
M-PT & MS-PL	**0.36** / **3.6**	**0.42** / **4.1**	**0.48** / **4.6**	**0.53** / **5.1**	3.99 / 1.44
MS-PT	0.30 / 3.6	0.35 / 4.1	0.40 / 4.6	0.43 / 5.1	3.45 / 1.63
MS-PT & M-PL	0.29 / 3.5	**0.33** / 4.0	**0.37** / 4.5	**0.41** / 4.9	3.09 / 1.44
MS-PT & MS-PL	**0.29** / **2.9**	0.35 / **3.3**	0.39 / **3.7**	0.42 / **4.1**	3.38 / 1.20

IV. Monte-Carlo Simulations

The proposed `ov_plane` is built as an extension to OpenVINS [20]. We generate a room surrounding the simulation trajectory and visual points lying on the planes, see Fig. 2. Data associations between points and planes are assumed to be known. Table I contains the key sensor frequencies, sensor properties, and noise parameters used in the simulation. Errors are reported using the Normalized Estimation Error Squared (NEES), Relative Pose Error (RPE), and Absolute Trajectory Error (ATE) metrics throughout the different experiments (see [64] and [65]).

Results for a 20 run Monte-Carlo are shown in Table II with different estimator configurations. All algorithms remain consistent as their NEES values are close to three. The M-PT & M-PL, which adds MSCKF planes, has little improvement over the baseline M-PT system. We attribute this to the MSCKF plane track length only being that of the sliding window size and the regularity does not improve MSCKF point linearizations by much. But, if the planes with sufficient observations are inserted into the state vector, M-PT & MS-PL, then a clear performance gain can be seen for all trajectory lengths. Within the simulation, the ceiling and floor can be tracked over significant portions of the trajectory allowing for improved feature triangulation and leveraging of the structural regularity information.

Next, we investigate the impact of co-estimating SLAM points and planes. The baseline MS-PT is more accurate than the M-PT as expected, but it is interesting to see that the M-PT & MS-PL is able to perform near the level of accuracy with only estimating, at maximum, six environmental planes alongside MSCKF point features. Again the MSCKF plane, MS-PT & M-PL, has little impact on accuracy over the point-only MS-PT, while the addition of SLAM plane estimation in MS-PT & MS-PL has the overall best accuracy. These simulation results demonstrate the improved VIO performance with planar regularities for both in-state SLAM and out-of-state MSCKF point features.

V. Real-World Experiments

We evaluate the proposed system on the Vicon room scenarios from the EuRoC MAV dataset [66] which provides 20Hz stereo images, 200Hz ADIS16448 MEMS IMU measurements, and optimized groundtruth trajectories. We do not evaluate on the machine hall scenarios due to their cluttered environment and lack of planar structures. An additional AR table dataset was collected as an example scenario in which a user walks around a central table.[2] An Intel Realsense D455[3] with 30Hz RGB-D (depth was not used) and 400Hz BMI055 IMU along with 100Hz OptiTrack poses were recorded in 1-2 minute segments. The groundtruth was recovered using the `vicon2gt` utility [67]. We extract 200 sparse point features and keep a maximum of 15 SLAM point features in MS-PL.[4] Two additional state-of-the-art visual-inertial systems, VINS-Fusion [68] and OKVIS [6], are evaluated in addition to OpenVINS [20], MS-PT, and the proposed `ov_plane` extensions.[5] All methods are run without loop-closure, with a monocular camera and IMU as input, and with spatial-temporal calibration if supported.

Algorithm 2 Plane Detection and Tracking

Sparse Point Features:
- FAST [69] detection with KLT optical flow [70]
- Robustified with 8-point RANSAC
- Provides frame-to-frame plane tracking

Point Feature Preprocessing:
- Point features are incrementally triangulated into 3D if sufficient observations
- Delaunay triangulation of valid features to determine spatial relationships [71]–[73]
- Each triangle's normal is computed using its three points: ${}^G\mathbf{v}_i = \text{normalize}({}^G\mathbf{p}_i - {}^G\mathbf{p}_0)$ ${}^G\mathbf{n}_j = \text{normalize}(\lfloor {}^G\mathbf{v}_2 \times \rfloor {}^G\mathbf{v}_1)$

Vertex Normals:
- Vertex normals of connected triangles are collected
- Compute angle variance and max angle difference between normals $\theta = \text{acosd}({}^G\mathbf{n}_i^\top {}^G\mathbf{n}_j)$
- If either is above a threshold, reject this vertex as being on the "edge" of two planes
- Else average normal is computed and vertex is valid

Vertex Matching Heuristics:
- For each valid vertex, i, compare its neighbors
- Normal difference: $\text{acosd}({}^G\mathbf{n}_i^\top {}^G\mathbf{n}_j) < \Delta\theta$
- Point-to-plane distance: ${}^G\mathbf{n}_i^\top {}^G\mathbf{p}_j - {}^G d_i < \Delta d_z$
- Avg. distance d_i of point ${}^G\mathbf{p}_i$ to N closest points [74] passes plane Z-test: $(d_i - \bar{d})/\sigma_d < z$

Plane Merging / ID Management:
- For all vertexes matched, select the smallest (oldest) plane id and assign it to all
- If no feature has a plane id (from previous frame or match), then assign a new id

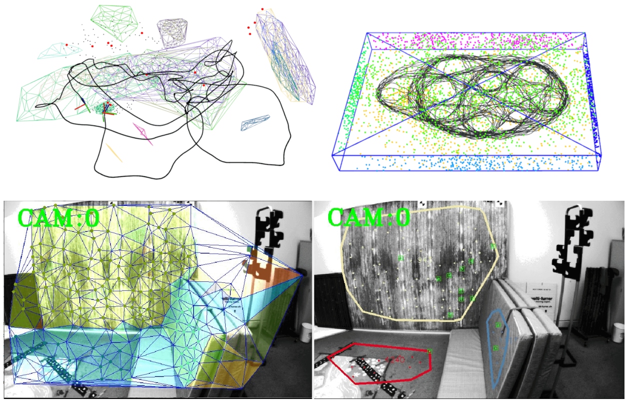

Fig. 2: EuRoC MAV [66] with estimated planes shown as meshes (not all in state vector, top left). Simulation environment (top, right) has a 1.2km trajectory in a 15.2 × 9.5 × 1.7m room (points are colored by plane). Bottom row shows V1_01 sparse tracking mesh with normals (left), and extracted planes (right).

A. Plane Detection and Tracking

Details of the plane extraction are summarized in Algorithm 2, and an example extraction with recovered normals can be seen in bottom of Fig. 2. From a high level, we first perform sparse temporal point feature tracking which provides frame-to-frame matching knowledge. The 3D position of point features are recovered efficiently in the global frame by incrementing their information at each timestep. We then recover a sparse 3D geometric mesh of the environment which is used to recover per-feature normals. A pairwise comparison with a series of heuristics is used to finally cluster points into common planes.

We ran the proposed plane tracker on a series of datasets and summarized its statistics in Table V. In datasets with high dynamic motions, it can be challenging to extract planes because of poor sparse point feature tracking and dynamic movement preventing sufficient observations for feature triangulation. In particular, the V1_03, V2_02, and V2_03 datasets have very dynamic motions which are not amendable for uniform point feature extraction and a large number of sufficiently observed planar features. It can also be noticed that these datasets have a very low number of features per plane, limiting the number of possible SLAM planes (and on some datasets no planes are used in an update). The additional computational cost for plane detection and matching is around 2-4ms, which is similar to sparse point feature tracking (around 3-4ms).

On the self-collected AR table datasets, we observed that due to the larger planar surfaces and longer-view time, planes can be sufficiently tracked for long periods of time with a high number of features per plane. This is amendable for leveraging structural regularities. In addition, it is not possible to take advantage of environmental white walls since no visual point features are extracted to facilitate plane detection. Thus extraction of planes remains limited to regions with sufficient texture.

B. EuRoC MAV Indoor Dataset

Table III shows the average ATE over each dataset for different configurations. Looking first at M-PT, on the V1_01 dataset there is a clear advantage to including SLAM plane features in the state (see Fig. 2 for extracted planes). The use of MSCKF planes seems to show the same performance without planes, mirroring the simulation results. For most datasets with limited plane extraction, see Table V planes per frame, there is very little improvement over point-based VIO. In general, the OpenVINS-based systems demonstrate

[2] https://github.com/rpng/ar_table_dataset

[3] https://www.intelrealsense.com/depth-camera-d455/

[4] All computational results were performed in a single thread on an Intel(R) Xeon(R) E3-1505Mv6 @ 3.00GHz.

[5] Note that we have tried to reproduce the results of [53]–[55] for a fair comparison, but were unable to achieve sufficient accuracy on their 2019 v4.0 code release. The latest main branch no longer supports the use of structural regularities.

TABLE III: EuRoC MAV ATE (degree / cm) along with average timing for the V1_01_easy dataset. $\sigma_d = 0.01$ was used.

Algorithm	V1_01	V1_02	V1_03	V2_01	V2_02	V2_03	Time (ms)
M-PT	0.83 / 8.6	1.57 / 9.1	2.50 / 15.5	1.73 / 12.1	1.34 / 9.4	1.61 / 15.6	8.3 ± 1.7
M-PT & M-PL	0.82 / 8.6	1.58 / 9.2	**2.45** / **15.3**	1.73 / 12.1	**1.22** / 9.7	1.61 / 15.6	12.2 ± 2.7
M-PT & MS-PL	**0.75** / **7.6**	**1.55** / **9.0**	2.50 / 15.5	1.73 / 12.1	1.28 / **8.8**	1.61 / 15.6	12.4 ± 2.7
MS-PT	1.32 / 8.4	1.58 / 7.0	**2.20** / **12.2**	**0.80** / 11.3	1.96 / 8.3	1.77 / 16.9	9.0 ± 2.0
MS-PT & M-PL	**0.61** / **5.3**	1.58 / 7.5	2.32 / 12.5	0.89 / 12.5	1.93 / 7.4	1.77 / 16.9	13.9 ± 3.8
MS-PT & MS-PL	0.75 / 6.9	**1.55** / **6.9**	2.41 / 12.5	0.82 / **10.8**	**1.40** / **6.8**	1.77 / 16.9	13.8 ± 3.4
VINS-Fusion [68]	1.24 / 5.8	2.61 / 11.5	3.61 / 20.5	1.99 / 8.0	3.13 / 8.7	3.54 / 19.7	31.9 ± 12.3*
OKVIS [6]	0.72 / 8.3	2.01 / 14.5	10.47 / 107.4	0.94 / 13.4	1.17 / 19.1	2.37 / 23.3	59.9 ± 31.6*

[*] Timing for VINS-Fusion [68] and OKVIS [6] *only* reports their optimization time (no feature tracking).

TABLE IV: Self-collected AR table ATE (degree / cm) and average timing for the table_01 dataset. $\sigma_d = 0.01$ was used.

Algorithm	table_01	table_02	table_03	table_04	table_05	table_06	table_07	table_08	Time (ms)
M-PT	**0.45** / 6.8	0.85 / 2.4	1.37 / 5.6	0.83 / 7.5	0.78 / 5.0	**0.66** / 4.9	0.94 / 4.8	2.00 / 12.5	8.7 ± 1.7
M-PT & M-PL	0.52 / 6.5	0.91 / 2.5	1.44 / 5.9	0.87 / 7.1	0.76 / 4.9	0.67 / 5.9	**0.85** / 4.7	2.02 / 12.8	13.3 ± 3.2
M-PT & MS-PL	0.67 / **4.6**	**0.72** / **2.0**	**0.96** / **3.0**	**0.75** / **3.2**	**0.62** / **4.0**	0.75 / **4.4**	0.92 / **4.2**	**1.88** / **9.2**	13.9 ± 2.9
MS-PT	**1.15** / 5.7	1.79 / 4.1	2.41 / 6.9	1.28 / 5.7	**0.56** / 2.7	**0.78** / **3.6**	**1.00** / 4.8	**0.68** / 11.2	9.4 ± 2.0
MS-PT & M-PL	1.32 / 5.5	0.89 / 2.5	**1.03** / **4.5**	1.10 / **4.7**	1.01 / 4.4	1.81 / 6.0	1.06 / **4.6**	1.29 / 11.2	15.0 ± 3.9
MS-PT & MS-PL	1.25 / **5.1**	**0.65** / **2.3**	1.05 / 4.6	**0.79** / 5.0	0.70 / **2.6**	1.29 / 4.5	1.12 / 5.1	0.82 / **6.8**	14.7 ± 3.2
VINS-Fusion [68]	1.62 / 5.8	1.32 / 3.0	1.47 / 7.6	1.75 / 5.6	1.12 / 3.4	0.98 / 5.3	1.67 / 9.3	5.03 / 23.3	35.6 ± 17.0*
OKVIS [6]	2.48 / 9.0	2.01 / 7.7	3.94 / 15.3	2.05 / 16.2	0.77 / 24.5	0.74 / 10.2	2.07 / 13.8	1.54 / 19.8	85.5 ± 32.6*

[*] Timing for VINS-Fusion [68] and OKVIS [6] *only* reports their optimization time (no feature tracking).

TABLE V: Tracking statistics and time to perform plane tracking (i.e., it does not include sparse point tracking). Statistics include: features per plane, average plane per frame, average plane tracking length, and active planes in the state per frame.

Dataset	Feat. / PL	PL / Frame	Track Len.	PL Active	Time (ms)
V1_01	19.6 ± 13.3	2.9 ± 1.3	53.4 ± 74.0	0.9 ± 0.7	3.3 ± 0.7
V1_02	13.7 ± 10.9	1.7 ± 1.3	20.0 ± 26.8	0.3 ± 0.5	2.5 ± 0.8
V1_03	10.1 ± 9.4	0.7 ± 1.0	24.9 ± 26.0	0.0 ± 0.2	2.0 ± 0.7
V2_01	8.0 ± 5.0	1.4 ± 1.3	39.9 ± 43.1	0.1 ± 0.3	2.5 ± 0.6
V2_02	9.5 ± 8.1	1.0 ± 1.1	23.3 ± 22.8	0.0 ± 0.1	2.1 ± 0.6
V2_03	6.3 ± 1.8	0.2 ± 0.4	14.4 ± 15.0	0.0 ± 0.0	1.4 ± 0.6
table_01	27.3 ± 13.1	2.7 ± 1.1	61.1 ± 227.6	1.1 ± 0.5	3.5 ± 0.7
table_02	82.0 ± 58.7	2.2 ± 1.3	49.1 ± 249.2	1.2 ± 0.6	4.1 ± 0.9
table_03	33.9 ± 21.3	3.0 ± 1.2	88.5 ± 337.4	1.5 ± 0.6	4.0 ± 0.7
table_04	35.3 ± 23.1	2.1 ± 0.9	68.6 ± 428.0	0.9 ± 0.4	4.2 ± 1.3
table_05	38.6 ± 27.6	2.5 ± 1.0	119.2 ± 327.2	1.2 ± 0.7	3.5 ± 0.6
table_06	43.5 ± 30.5	2.0 ± 0.9	69.3 ± 131.6	1.1 ± 0.8	3.2 ± 0.8
table_07	16.6 ± 8.2	2.8 ± 0.9	106.8 ± 163.8	0.3 ± 0.5	3.0 ± 0.6
table_08	20.7 ± 13.5	1.8 ± 1.0	54.1 ± 260.1	0.6 ± 0.5	2.7 ± 0.6

superior computational efficiency and outperform other state-of-the-art methods.

When SLAM point features are included, MS-PT, the performance gains between point-based and plane-aided become smaller. There can even be cases where the use of planes can hurt performance, which we equate to SLAM point features being more sensitive to incorrect data associations due to their length of time in the state. The system is able to perform well above the real-time threshold of 50ms, with the increase in computation mainly coming from plane detection and matching.

C. AR Table Dataset

The ATE for the self-collected AR table dataset is shown in Table IV. Looking at M-PT, it is clear that there is a significant improvement of 1-3cm of accuracy when planar regularities are used. The table or floor planes are typically tracked over large segments of the trajectory, see Table V average track length, and thus provide a long-term loop-closure for all points. The planes' large spatial volume also allows for more accurate feature triangulation, possibly reducing linearization errors. When SLAM point features are added, there is still a gain of accuracy on most datasets, but there are a few where planar constraints can hurt performance. We plan to investigate this in the future. We additionally see that the use of MSCKF plane features has little impact both in real-world experiments and simulations thus we do not recommend their use as a regularization source.

VI. Conclusion and Future Work

In this work, we developed a novel, lightweight MSCKF-based VIO system that can incorporate planar regularities without requiring an additional sensor or neural network. To the best of the authors' knowledge, we are the first to incorporate planar regularities as higher-level structural information within an efficient *monocular* MSCKF-VIO. To achieve real-time and accurate performance, the proposed VIO architecture is carefully designed with in-state SLAM and out-of-state MSCKF point and plane features. A novel and robust plane detection and tracking algorithm was evaluated and shown to recover co-planar point features efficiently. Extensive simulation and real-world experimental results demonstrate that the proposed system is able to outperform traditional point-based VIO in man-made environments. We have publicly released the AR table dataset for the research community.

In the future, we plan to investigate if planes can reduce linearization errors, crucial to the use of first-estimates Jacobians (FEJ) [15], [75], of point features as they are jointly refined and thus could improve poor point feature triangulation results. We are also interested in applying regularization to lines and including cross-plane constraints. We will additionally investigate efficiently building large-scale plane maps [76] with reduced feature representations.

References

[1] Google, "ARCore," https://developers.google.com/ar.

[2] Apple, "ARKit," https://developer.apple.com/augmented-reality/.

[3] Meta, "Oculus," https://store.facebook.com/quest/.

[4] G. Huang, "Visual-inertial navigation: A concise review," in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019.

[5] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[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] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[8] 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.

[9] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3565–3572.

[10] P. Geneva, N. Merrill, Y. Yang, C. Chen, W. Lee, and G. Huang, "Versatile 3d multi-sensor fusion for lightweight 2d localization," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2020.

[11] P. Geneva and G. Huang, "Map-based visual-inertial localization: A numerical study," in *Proc. International Conference on Robotics and Automation*, Philadelphia, USA, May 2022.

[12] Y. Yang, C. Chen, W. Lee, and G. Huang, "Decoupled right invariant error states for consistent visual-inertial navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1627–1634, 2022.

[13] P. Zhu, Y. Yang, W. Ren, and G. Huang, "Cooperative visual-inertial odometry," in *Proc. of the IEEE International Conference on Robotics and Automation*, Xi'an, China, 2021.

[14] C. Chen, Y. Yang, P. Geneva, W. Lee, and G. Huang, "Visual-inertial-aided online mav system identification," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[15] C. Chen, Y. Yang, P. Geneva, and G. Huang, "FEJ2: a consistent visual-inertial state estimator design," in *Proc. International Conference on Robotics and Automation*, Philadelphia, USA, May 2022.

[16] Y. Yang, J. Maley, and G. Huang, "Null-space-based marginalization: Analysis and algorithm," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, Canada, Sept. 2017, pp. 6749–6755.

[17] M. Li and A. I. Mourikis, "Vision-aided inertial navigation for resource-constrained systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1057–1063.

[18] ——, "Optimization-based estimator design for vision-aided inertial navigation," in *Robotics: Science and Systems*. Berlin Germany, 2013, pp. 241–248.

[19] ——, "High-precision, consistent ekf-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[20] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: a research platform for visual-inertial estimation," in *Proc. of the IEEE International Conference on Robotics and Automation*, Paris, France, 2020. [Online]. Available: https://github.com/rpng/open_vins

[21] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[22] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*. IEEE, 2011, pp. 127–136.

[23] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[24] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "ElasticFusion: Dense slam without a pose graph." Robotics: Science and Systems, 2015.

[25] T. Laidlow, M. Bloesch, W. Li, and S. Leutenegger, "Dense rgb-d-inertial slam with map deformations," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6741–6748.

[26] C. X. Guo and S. I. Roumeliotis, "IMU-RGBD camera navigation using point and plane features," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3164–3171.

[27] Y. Yang and G. Huang, "Observability analysis of aided ins with heterogeneous features of points, lines and planes," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 399–1418, Dec. 2019.

[28] S. Yang and S. Scherer, "Monocular object and plane slam in structured environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3145–3152, 2019.

[29] L. Zhou, G. Huang, Y. Mao, J. Yu, S. Wang, and M. Kaess, "Plc-lislam: Lidar slam with planes, lines and cylinders," *IEEE Robotics and Automation Letters*, 2022.

[30] J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *IEEE International Conference on Computer Vision*, vol. 2. IEEE, 1999, pp. 941–947.

[31] G. Schindler and F. Dellaert, "Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2004, pp. I–I.

[32] D. G. Kottas and S. I. Roumeliotis, "Exploiting urban scenes for vision-aided inertial navigation." in *Robotics: Science and Systems*, 2013.

[33] C. X. Guo, K. Sartipi, R. C. DuToit, G. A. Georgiou, R. Li, J. O'Leary, E. D. Nerurkar, J. A. Hesch, and S. I. Roumeliotis, "Large-scale cooperative 3d visual-inertial mapping in a manhattan world," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1071–1078.

[34] C. X. Guo, K. Sartipi, R. C. DuToit, G. A. Georgiou, R. Li, J. O'Leary, E. D. Nerurkar, J. A. Hesch, and S. I. Roumeliotis, "Resource-aware large-scale cooperative three-dimensional mapping using multiple mobile devices," *IEEE Transactions on Robotics*, vol. 34, no. 5, pp. 1349–1369, 2018.

[35] D. Zou, Y. Wu, L. Pei, H. Ling, and W. Yu, "Structvio: visual-inertial odometry with structural regularity of man-made environments," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 999–1013, 2019.

[36] Y. Li, R. Yunus, N. Brasch, N. Navab, and F. Tombari, "Rgb-d slam with structural regularities," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 581–11 587.

[37] H. Wei, F. Tang, Z. Xu, and Y. Wu, "Structural regularity aided visual-inertial odometry with novel coordinate alignment and line triangulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 613–10 620, 2022.

[38] M. Hsiao, E. Westman, and M. Kaess, "Dense planar-inertial slam with structural constraints," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6521–6528.

[39] Y. Yang, P. Geneva, X. Zuo, K. Eckenhoff, Y. Liu, and G. Huang, "Tightly-coupled aided inertial navigation with point and plane features," in *Proc. International Conference on Robotics and Automation*, Montreal, Canada, May 2019.

[40] M. Scheiber, J. Delaune, S. Weiss, and R. Brockers, "Mid-air range-visual-inertial estimator initialization for micro air vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7613–7619.

[41] J. Hu, J. Hu, Y. Shen, X. Lang, B. Zang, G. Huang, and Y. Mao, "1d-lrf aided visual-inertial odometry for high-altitude mav flight," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 5858–5864.

[42] X. Zuo, Y. Yang, P. Geneva, J. Lv, Y. Liu, G. Huang, and M. Pollefeys, "Lic-fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2020.

[43] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: Lidar-inertial-camera odometry," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Macau, China, Nov. 2019.

[44] W. Lee, Y. Yang, and G. Huang, "Efficient multi-sensor aided inertial navigation with online calibration," in *Proc. of the IEEE International Conference on Robotics and Automation*, Xi'an, China, 2021.

[45] 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.

[46] R. Yunus, Y. Li, and F. Tombari, "Manhattanslam: Robust planar tracking and mapping leveraging mixture of manhattan frames," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6687–6693.

[47] 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.

[48] R. Wang, D. Geraghty, K. Matzen, R. Szeliski, and J.-M. Frahm, "VPLNet: Deep single view normal estimation with vanishing points and lines," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 689–698.

[49] T. Do, K. Vuong, S. I. Roumeliotis, and H. S. Park, "Surface normal estimation of tilted images via spatial rectifier," in *European Conference on Computer Vision*. Springer, 2020, pp. 265–280.

[50] W. Yin, Y. Liu, and C. Shen, "Virtual normal: Enforcing geometric constraints for accurate and robust depth prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[51] 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.

[52] F. Yang and Z. Zhou, "Recovering 3d planes from a single image via convolutional neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 85–100.

[53] A. Rosinol, T. Sattler, M. Pollefeys, and L. Carlone, "Incremental visual-inertial 3d mesh generation with structural regularities," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8220–8226.

[54] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 1689–1696.

[55] A. Rosinol, "Densifying sparse vio: a mesh-based approach using structural regularities," Master's thesis, ETH Zurich; Massachusetts Institute of Technology (MIT), 2018.

[56] X. Li, Y. He, J. Lin, and X. Liu, "Leveraging planar regularities for point line visual-inertial odometry," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5120–5127.

[57] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "LIPS: Lidar-inertial 3d plane slam," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, Spain, Oct. 2018.

[58] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 1997.

[59] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., Mar. 2005.

[60] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *IET Control Theory & Applications*, vol. 4, no. 8, pp. 1303–1318, 2010.

[61] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[62] S. Agarwal, K. Mierle, and Others, "Ceres solver," https://github.com/ceres-solver/ceres-solver, 2012.

[63] M. Li, "Visual-inertial odometry on resource-constrained systems," Ph.D. dissertation, UC Riverside, 2014.

[64] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7244–7251.

[65] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[66] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[67] P. Geneva and G. Huang, "vicon2gt: Derivations and analysis," University of Delaware, Tech. Rep. RPNG-2020-VICON2GT, 2020, available: http://udel.edu/~ghuang/papers/tr_vicon2gt.pdf.

[68] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," *arXiv preprint arXiv:1901.03638*, 2019.

[69] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, 2006, pp. 430–443.

[70] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, Vancouver, BC, Aug. 1981, pp. 674–679.

[71] B. Žalik and I. Kolingerová, "An incremental construction algorithm for delaunay triangulation using the nearest-point paradigm," *International Journal of Geographical Information Science*, vol. 17, no. 2, pp. 119–138, 2003.

[72] M. V. Anglada, "An improved incremental algorithm for constructing restricted delaunay triangulations," *Computers & Graphics*, vol. 21, no. 2, pp. 215–223, 1997.

[73] A. Amirkhanov, K. Äkerblom, and Others, "Constrained delaunay triangulation," https://github.com/artem-ogre/CDT, 2019.

[74] Y. Cai, W. Xu, and F. Zhang, "ikd-tree: An incremental kd tree for robotic applications," *arXiv preprint arXiv:2102.10808*, 2021.

[75] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "A quadratic-complexity observability-constrained unscented Kalman filter for SLAM," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1226–1243, Oct. 2013.

[76] P. Geneva, J. Maley, and G. Huang, "An efficient schmidt-ekf for 3D visual-inertial SLAM," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, June 2019.