LC: add: “Probabilistic Structure from Motion with Objects (PSfMO)”

Semantic 3D Mesh VIO

Antoni Rosinol[1], Siyi Hu[1], Luca Carlone[1]

***Abstract*—Classical implementations of Visual-Inertial Odometry (VIO) algorithms ignore semantic information of the scene, as they rely solely on sparse landmarks. Nevertheless, recent work has shown the advantage of using richer representations of the scene, such as 3D meshes, to extract higher-level information such as structural regularities. In this work, we show that a 3D mesh of the scene can be further utilized to accommodate semantic information, which enhances the mapping side of a classical VIO beyond a sparse and uninformative point-cloud. Towards this end, we use recent work on semidefinite programming and conditional random fields to generate semantic information in real-time on a single-core CPU.**

***Index Terms*—Vision-Based Navigation, Semantic Segmentation.**

Supplementary Material

Videos of the experiments: **TODO:** Add video url

I. Introduction

RECENT advances in VIO are enabling a wide range of applications, ranging from virtual and augmented reality to agile drone navigation [1].

Contributions. In this paper, we propose to *incrementally build a 3D mesh restricted to the receding horizon of the VIO optimization.* In this way, we can map larger areas than a per-frame approach, while memory footprint and computational complexity associated to the mesh remain bounded.

Paper Structure. Section III presents the mathematical formulation of our approach, and discusses the implementation of our VIO front-end and back-end. Section IV reports and discusses the experimental results and comparison against related work. Section V concludes the paper.

II. Related Work

A. TODO

B. Semantic Segmentation

III. Approach

TODO: Apprach We consider a stereo visual-inertial system and adopt a *keyframe*-based approach [2]. This section describes our VIO front-end and back-end. Our front-end proceeds by building a 2D Delaunay triangulation over the 2D keypoints at each keyframe. Then, the VIO back-end estimates the 3D position of each 2D keypoint, which we use to project the 2D triangulation into a 3D mesh. While we incrementally build the 3D mesh, we restrict the mesh to the time-horizon of the VIO optimization, which we formulate in a fixed-lag smoothing framework [3], [4]. The 3D mesh is further used to extract structural regularities in the scene that are then encoded as constraints in the optimization problem.

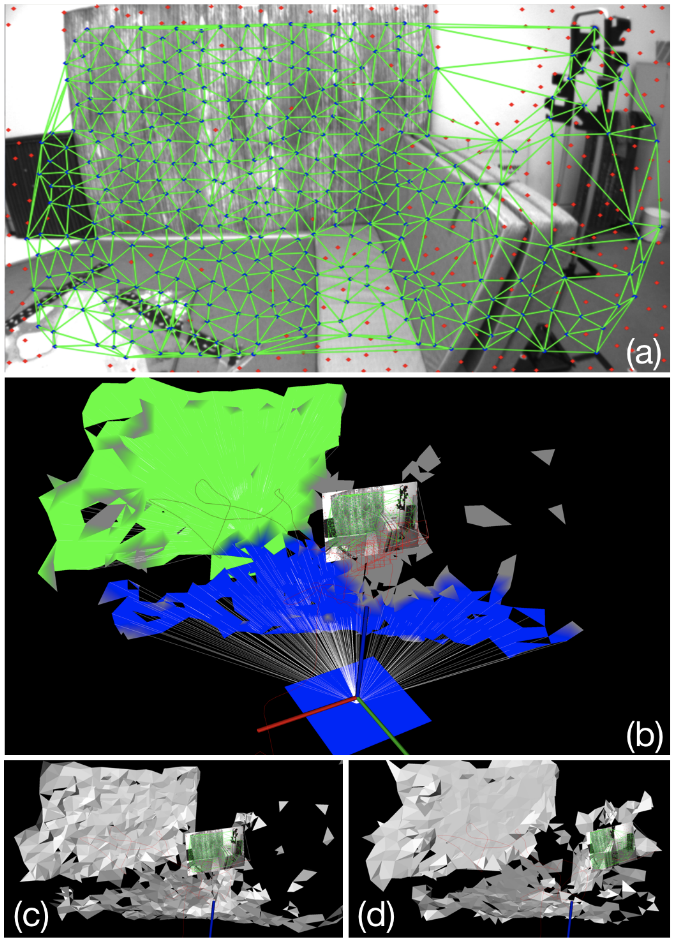

Fig. 1: We propose a VIO pipeline that incrementally builds a 3D mesh of the environment starting from (a) a 2D Delaunay triangulation of keypoints. We also detect and enforce *structural regularities*, c.f. (b) planar walls (green) and floor (blue). The bottom row in the figure compares the mesh constructed (c) without and (d) with structural regularities.

A. Front-end

Our front-end has the same components as a keyframe-based indirect visual-inertial odometry pipeline [2], [5], but it also incorporates a module to generate a 3D mesh, and a module to detect structural regularities from the 3D mesh. We refer the reader to [6, Sec. 4.2.1] for details on the standard modules used, and we focus here instead on the 3D mesh generation and regularity detection.

1) 3D Mesh Generation: Building a consistent 3D Mesh of the environment using a sparse point cloud from VIO is difficult because (i) the 3D positions of the landmarks are noisy, and some are outliers; (ii) the density of the point cloud is highly irregular; (iii) the point cloud is constantly morphing: points are being removed (marginalized) and added, while the

[1] A. Rosinol, S. Hu and L. Carlone are with the Laboratory for Information & Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA, USA, {arosinol,siyihu,lcarlone}@mit.edu

This work was partially funded by ARL DCIST CRA W911NF-17-2-0181.

landmarks' positions are being updated at each optimization step. Therefore, we avoid performing a 3D tetrahedralisation directly from the sparse 3D landmarks, which would require expensive algorithms, such as space carving [7].

2) 3D Mesh Propagation: While some algorithms update the mesh for a single frame [8], [9], we attempt to maintain a mesh over the receding horizon of the fixed-lag smoothing optimization problem (section III-B), which contains multiple frames. The motivation is three-fold: (i) A mesh spanning multiple frames covers a larger area of the scene, which provides more information than just the immediate field of view of the camera. (ii) We want to capture the structural regularities affecting any landmark in the optimization problem. (iii) Anchoring the 3D mesh to the time-horizon of the optimization problem also bounds the memory usage, as well as the computational complexity of updating the mesh.

a) Temporal propagation: deals with the problem of updating the 3D mesh when new keypoints appear and/or old ones disappear in the new frame.

b) Spatial propagation: deals with the problem of updating the global 3D mesh when a new local 3D mesh is available, and when old landmarks are marginalized from the optimization's time-horizon. We solve the first problem by merging the new local 3D mesh to the previous (global) mesh, by ensuring no duplicated 3D faces are present.

3) Regularity Detection:

4) Data Association:

B. Back-end

1) State Space: If we denote the set of all keyframes up to time t by $\mathcal{K}_t$, the state of the system $\mathbf{x}_i$ at keyframe $i \in \mathcal{K}_t$, is described by the IMU orientation $\mathbf{R}_i$, position $\mathbf{p}_i$, velocity $\mathbf{v}_i$ and biases $\mathbf{b}_i$:

$$\mathbf{x}_i \doteq [\mathbf{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i], \tag{1}$$

where the pose $(\mathbf{R}_i, \mathbf{p}_i) \in \mathrm{SE}(3)$, $\mathbf{v}_i \in \mathbb{R}^3$, and $\mathbf{b}_i = [\mathbf{b}_i^g \quad \mathbf{b}_i^a] \in \mathbb{R}^6$, and $\mathbf{b}_i^g, \mathbf{b}_i^a \in \mathbb{R}^3$ are the gyroscope and accelerometer biases, respectively.

We will only estimate the 3D positions $\boldsymbol{\rho}_l$ for a subset Λ_t of all landmarks $\mathcal{L}_t$ visible up to time t: $\{\boldsymbol{\rho}_l\}_{l\in\Lambda_t}$, where $\Lambda_t \subseteq \mathcal{L}_t$. We will avoid optimizing over the rest of the landmarks $\mathcal{L}_t \setminus \Lambda_t$ by using a structureless approach, as defined in [10, Sec. VII], which circumvents the need to add the landmarks' positions as variables in the state. This allows to trade accuracy for speed; as the optimizations complexity increases with the number of variables to be estimated.

The set Λ_t corresponds to the landmarks which we consider to satisfy a structural regularity. In particular, we are interested in co-planarity regularities, which we introduce in section III-B5. Since we need the explicit landmark variables to formulate constraints on them, we avoid using a structureless approach for these landmarks.

Finally, the co-planarity constraints between the landmarks Λ_t require the modelling of the planes Π_t in the scene. Therefore, the variables to be estimated comprise the state of the system $\{\mathbf{x}_i\}_{i\in\mathcal{K}_t}$, the landmarks which we consider to satisfy structural regularities $\{\boldsymbol{\rho}_l\}_{l\in\Lambda_t}$, and the planes $\{\boldsymbol{\pi}_\pi\}_{\pi\in\Pi_t}$. The variables to be estimated at time t are:

$$\mathcal{X}_t \doteq \{\mathbf{x}_i, \boldsymbol{\rho}_l, \boldsymbol{\pi}_\pi\}_{i\in\mathcal{K}_t, l\in\Lambda_t, \pi\in\Pi_t} \tag{2}$$

Since we are taking a fixed-lag smoothing approach for the optimization, we limit the estimation problem to the sets of variables that depend on the keyframes in a time-horizon of size Δ_t. To avoid cluttering the notation, we skip the dependence of the sets $\mathcal{K}_t$, Λ_t and Π_t on the parameter Δ_t.

By reducing the number of state variables to a given window of time Δ_t, we will make the optimization problem tractable and solvable in real-time.

2) Measurements: The input for our system consists on measurements from the camera and the IMU. We define the image measurements at keyframe i as $\mathcal{C}_i$. The camera can observe multiple landmarks l, hence $\mathcal{C}_i$ contains multiple image measurements $\mathbf{z}_i^l$, where we distinguish the landmarks that we will use for further structural regularities $\mathbf{z}_i^{l_c}$ (where the index c in l_c stands for constrained landmark), and the landmarks that will remain as structureless $\mathbf{z}_i^{l_s}$ (where the s in the index of l_s stands for structureless). We represent the set of IMU measurements acquired between two consecutive keyframes i and j as $\mathcal{I}_{ij}$. Therefore, we define the set of measurements collected up to time t by $\mathcal{Z}_t$:

$$\mathcal{Z}_t \doteq \{\mathcal{C}_i, \mathcal{I}_{ij}\}_{(i,j)\in\mathcal{K}_t}. \tag{3}$$

3) Factor Graph Formulation: We want to estimate the posterior probability $p(\mathcal{X}_t|\mathcal{Z}_t)$ of our state $\mathcal{X}_t$, as defined in eq. (2), using the set of measurements $\mathcal{Z}_t$, defined in eq. (3). Using standard independence assumptions between measurements and states, we arrive to the formulation in eq. (4), where we grouped the different terms in factors ϕ:

$$\begin{aligned} p(\mathcal{X}_t|\mathcal{Z}_t) &\overset{(a)}{\propto} p(\mathcal{Z}_t|\mathcal{X}_t)p(\mathcal{X}_t) \\ &= \phi_0(\mathbf{x}_0) \prod_{l_c\in\Lambda_t}\prod_{\pi\in\Pi_t} \phi_{\mathcal{R}}(\boldsymbol{\rho}_{l_c}, \boldsymbol{\pi}_\pi)^{\delta(l_c,\pi)} && \text{(4a)} \\ &\prod_{(i,j)\in\mathcal{K}_t} \phi_{\text{IMU}}(\mathbf{x}_i, \mathbf{x}_j) && \text{(4b)} \\ \prod_{i\in\mathcal{K}_t}\prod_{l_c\in\mathcal{C}_i^c} &\phi_{l_c}(\mathbf{x}_i, \boldsymbol{\rho}_{l_c}) \prod_{i\in\mathcal{K}_t}\prod_{l_s\in\mathcal{C}_i^s} \phi_{l_s}(\mathbf{x}_i) && \text{(4c)} \end{aligned}$$

where we apply the Bayes rule in (a), and ignore the normalization factor over the measurements since it will not influence the result (section III-B4).

Equation (4a) corresponds to the prior information we have over the state $p(\mathcal{X}_t)$. In this term, we encode regularity constraints between landmarks $\boldsymbol{\rho}_{l_c}$ and planes π, which we denote by $\phi_{\mathcal{R}}$. We also introduce the data association term $\delta(l_c, \pi)$, which returns a value of 1 if the landmark l_c is associated to the plane π, 0 otherwise. We explain in section III-A4 how the data association is done. The factor ϕ_0 represents a prior on the first pose of the optimization's time-horizon.

In eq. (4b), we have the factor corresponding to the IMU measurements, which depends only on the consecutive keyframes $(i, j) \in \mathcal{K}_t$.

Finally, eq. (4c) encodes the factors corresponding to the camera measurements. Since we want to distinguish between landmarks that are involved in structural regularities (l_c) and landmarks that are not (l_s), we split the product over C_i; where we write $l_s \in \mathcal{C}_i^s$ or $l_c \in \mathcal{C}_i^c$ when a landmark l_s or l_c, respectively, is seen at keyframe i by the camera. Note that $\mathcal{C}_i = \mathcal{C}_i^c \cup \mathcal{C}_i^s$ and $\mathcal{C}_i^c \cap \mathcal{C}_i^s = \emptyset$.

In fig. 2, we use the expresiveness of factor graphs [11] to show the actual dependencies between the variables in eq. (4)[1].

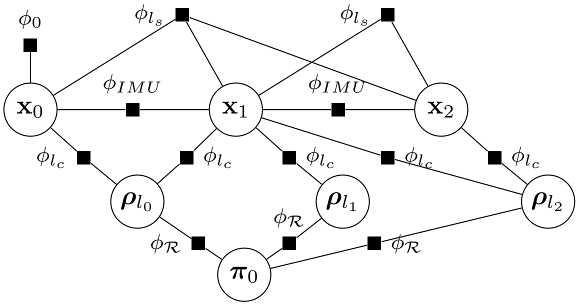

Fig. 2: VIO factor graph combining Structureless (ϕ_{l_s}), Projection (ϕ_{l_c}) and Regularity ($\phi_\mathcal{R}$) factors (SPR). The factor $\phi_\mathcal{R}$ encodes relative constraints between a landmark l_i and a plane π_0.

4) MAP Estimation: Since we are only interested in the most likely state $\mathcal{X}_t$ given the measurements $\mathcal{Z}_t$, we calculate the *maximum a posteriori* (MAP) estimator $\mathcal{X}_t^{\text{MAP}}$. Maximizing $\mathcal{X}_t^{\text{MAP}}$ is nevertheless not as convenient as minimizing the negative logarithm of the posterior probability, which, using eq. (4), simplifies to eq. (5) for zero-mean Gaussian noise:

$$\begin{aligned}\mathcal{X}_t^{\text{MAP}} = \arg\min_{\mathcal{X}_t} \|\mathbf{r}_0\|^2_{\Sigma_0} + \sum_{l_c\in\Lambda_t}\sum_{\pi\in\Pi_t} \delta(l_c,\pi)\,\|\mathbf{r}_\mathcal{R}\|^2_{\Sigma_\mathcal{R}} \\ + \sum_{(i,j)\in\mathcal{K}_t}\|\mathbf{r}_{\mathcal{I}_{ij}}\|^2_{\Sigma_{ij}} + \sum_{i\in\mathcal{K}_t}\left\{\sum_{l_c\in\mathcal{C}_i}\left\|\mathbf{r}_{\mathcal{C}_i^{l_c}}\right\|^2_{\Sigma_\mathcal{C}} + \sum_{l_s\in\mathcal{C}_i}\left\|\mathbf{r}_{\mathcal{C}_i^{l_s}}\right\|^2_{\Sigma_\mathcal{C}}\right\}\end{aligned} \tag{5}$$

where $\mathbf{r}$ represents the residual errors, and $\boldsymbol{\Sigma}$ the covariance matrices. We refer the reader to [10, Sec. VI, VII] for the actual formulation of the preintegrated IMU factors ϕ_{IMU} and structureless factors ϕ_{l_s}, as well as the underlying residual functions $\mathbf{r}_{\text{IMU}}$, $\mathbf{r}_{C_i^{l_s}}$. For the projection factors ϕ_{l_c}, we use a standard monocular and stereo reprojection error formulation as in [4].

5) Regularity Constraints: For the regularity factors $\phi_\mathcal{R}$, we use a co-planarity constraint between a landmark $\boldsymbol{\rho}_{l_c} \in \mathbb{R}^3$ and a plane $\pi = \{\boldsymbol{n}, d\}$, where $\boldsymbol{n}$ is the normal of the plane, which lives in the $\mathrm{S}^2 \doteq \{\mathbf{n} = (n_x, n_y, n_z)^T \big| \|\mathbf{n}\| = 1\}$ manifold, and $d \in \mathbb{R}$ the distance to the origin: $\mathbf{r}_\mathcal{R} = \boldsymbol{n} \cdot \boldsymbol{\rho}_{l_c} - d$. Representing a plane by its normal and distance to the origin is an over-parametrization that will lead to an information matrix that is singular. This is not amenable for Gauss-Newton optimization, since it leads to singularities in the normal equations [13]. To avoid the over-parametrization problem, we optimize in the tangent space $T_{\boldsymbol{n}}S^2 \sim \mathbb{R}^2$ of S^2 and define a suitable retraction $\mathcal{R}_{\boldsymbol{n}}(\boldsymbol{v}) : T_{\boldsymbol{n}}S^2 \in \mathbb{R}^2 \to \mathrm{S}^2$ to map changes in the tangent space to changes to the normals in S^2 [10]. In other words, we rewrite the residuals as:

$$\mathbf{r}_\mathcal{R}(\boldsymbol{v}, d) = \mathcal{R}_{\boldsymbol{n}}(\boldsymbol{v})^\intercal \cdot \boldsymbol{\rho} - d \tag{6}$$

and optimize with respect to the minimal parametrization $\boldsymbol{v}$. This is similar to the proposal of Kaess [13], but we work on the manifold S^2, while Kaess adopts a quaternion parametrization. Note that, a single co-planarity constraint, as defined in eq. (6), is not sufficient to constrain a plane variable, and a minimum of three are needed instead. Nevertheless, degenerate configurations exist, e.g. three landmarks on a line would not fully constrain a plane. Therefore, we ensure that a plane candidate has a minimum number of constraints before adding it to the optimization problem. **TODO:** Do we need to mention robust cost functions at all? I think so, reviews were picky about outliers!

IV. Experimental Results

We benchmark the proposed approach against the state of the art on real datasets, and evaluate trajectory and map estimation accuracy, as well as runtime. We use the EuRoC dataset [14], which contains visual and inertial data recorded from an micro aerial vehicle flying indoors. The EuRoC dataset includes eleven datasets in total, recorded in two different scenarios. The *Machine Hall* scenario (MH) is the interior of an industrial facility. It contains very little (planar) regularities. The *Vicon Room* (V) is similar to an office room where walls, floor and ceiling are close together, and other planar surfaces are visible (boxes, stacked mattresses). Datasets V1 and V2 differ only by the position of the objects in the scene. Each dataset provides the ground truth trajectory of the drone, allowing us to evaluate the accuracy of our estimation. For the V datasets, we are also provided with a ground truth point cloud of the scene, which we use to evaluate the accuracy of our mesh.

Compared techniques. To assess the advantages of our proposed approach, we compare three formulations that build one on top of the other. First, we denote as **S**, the approach that would neither use regularity factors, nor projection factors, but only use Structureless factors (ϕ_{l_s}, in eq. (4c)). Second, we denote as **SP**, the approach which would use Structureless factors, combined with Projection factors for those landmarks that have co-planarity constraints (ϕ_{l_c}, in eq. (4c)), but without using regularity factors. Finally, we denote as **SPR**, our proposed formulation using Structureless, Projection and Regularity factors ($\phi_\mathcal{R}$, in eq. (4a)). The IMU factors (ϕ_{IMU}, in eq. (4b)) are implicitly used in all three formulations. We also compare our results with other state-of-the-art implementations in table II. In particular, we compare the Root Mean Squared Error (RMSE) of our pipeline against OKVIS [15], MSCKF [16], ROVIO [5], VINS-MONO [3], and SVO-GTSAM [10], using the reported values in [17]. We refer the reader to [17] for details on the particular implementations and set of parameters used for each algorithm. Note that these algorithms use a monocular camera, while we use a stereo camera. Therefore, while [17] aligns the trajectories using $\mathrm{Sim}(3)$, we use instead $\mathrm{SE}(3)$. Nevertheless, the scale is observable for all algorithms since they use an IMU. We only report the values for VINS-MONO when its loop-closure module is disabled.

[1] We will use the notation proposed in [12] to represent the factor graph.

A. Localization Performance

The state of our optimization problem comprises the poses of the IMU, the velocities, the IMU biases, the planes' parameters, and the landmarks' positions. In this section we start by benchmarking the quality of the trajectory estimates, which are of paramount importance for control and AR/VR applications. The plane and landmark estimates will be assessed in the next section IV-B, where we evaluate the quality of the mesh. We will assess the quality of the plane and landmark estimates in Section IV-B.

Performance Metrics: Absolute Translation Error (ATE). ATE looks at the translational part of the relative pose between the ground truth pose and the corresponding estimated pose at a given timestamp. We first align our estimated trajectory with the ground truth trajectory both temporally and spatially (in SE(3)), as explained in [6, Sec. 4.2.1]. We refrain from using the rotational part since the trajectory alignment ignores the orientation of the pose estimates. Table I shows the ATE for our pipeline when using the pipelines S, SP, and our proposed approach SPR on the EuRoC dataset.

First, if we look at the performance of the different algorithmic variants for the datasets MH_03, MH_04 and MH_05 in table I, we observe that all methods perform equally. This is because in these datasets no structural regularities were detected. Hence, the proposed pipeline SPR gracefully downgrades to a standard structureless VIO pipeline (S). Second, looking at the results for dataset V2_03, we observe that both the SP and the SPR pipeline achieve the exact same performance. In this case, structural regularities are detected, resulting in Projection factors being used. Nevertheless, since the number of regularities detected is not sufficient to spawn a new plane estimate, no structural regularities are actually enforced in the factor graph. Finally, table I shows that the SPR pipeline consistently achieves better results over the rest of datasets where structural regularities are detected and enforced. In particular, the performance of SPR increases up to 28% on the median ATE compared to the SP pipeline for datasets with multiple planes (e.g. V1 and V2).

Table II shows that our approach, using structural regularities (SPR), achieves the best results when compared with the state-of-the-art, on datasets with structural regularities, such as in datasets V1_01_easy and V1_02_medium, where multiple planes are present (walls, floor). We observe a 19% improvement compared to the next best performing algorithm (SVO-GTSAM) in dataset V1_01_easy, and a 26% improvement in dataset V1_02_medium compared to ROVIO and VINS-MONO, which achieve the next best results. We also see that the performance of our pipeline is on-par with other state-of-the-art approaches when no structural regularities are present, such as in datasets MH_04_difficult and MH_05_difficult.

TABLE I: ATE for pipelines S, SP, and SPR. Our proposed approach SPR achieves the best results for all datasets where structural regularities are detected and enforced.

	ATE [cm]					
	S		SP		SPR (**Proposed**)	
EuRoC Sequence	Median	RMSE	Median	RMSE	Median	RMSE
MH_02_easy	12.9	13.1	17.6	16.7	**12.6**	**13.0**
MH_03_medium	**21.0**	**21.2**	**21.0**	**21.2**	**21.0**	**21.2**
MH_04_difficult	**17.3**	**21.7**	**17.3**	**21.7**	**17.3**	**21.7**
MH_05_difficult	**21.6**	**22.6**	**21.6**	**22.6**	**21.6**	**22.6**
V1_01_easy	5.6	6.4	6.2	7.7	**5.3**	**5.7**
V1_02_medium	7.7	8.9	8.7	9.4	**6.3**	**7.4**
V1_03_difficult	17.7	23.1	13.6	17.6	**13.5**	**16.7**
V2_01_easy	8.0	8.9	6.6	8.2	**6.3**	**8.1**
V2_02_medium	8.8	12.7	9.1	13.5	**7.1**	**10.3**
V2_03_difficult	37.9	41.5	**26.0**	**27.2**	**26.0**	**27.2**

TABLE II: ATE's RMSE of the state-of-the-art techniques (reported values from [17]) compared to our proposed SPR pipeline, on the EuRoC dataset. A cross (×) states that the pipeline failed. In **bold** the best result, in blue the second best.

	RMSE ATE [cm]					
Sequence	OKVIS	MSCKF	ROVIO	VINS-MONO	SVO-GTSAM	**SPR**
MH_01_e	16	42	21	27	**5**	**14**
MH_02_e	22	45	25	**12**	**3**	13
MH_03_m	24	23	25	**13**	**12**	21
MH_04_d	34	37	49	23	**13**	**22**
MH_05_d	47	48	52	35	**16**	**23**
V1_01_e	9	34	10	**7**	**7**	**6**
V1_02_m	20	20	**10**	**10**	11	**7**
V1_03_d	24	67	**14**	**13**	×	17
V2_01_e	13	10	12	**8**	**7**	**8**
V2_02_m	16	16	14	**8**	×	**10**
V2_03_d	29	113	**14**	**21**	×	27

Performance Metrics: Relative Pose Error (RPE). While the ATE provides information on the global consistency of the trajectory estimate, it does not provide insights on the moment in time when the erroneous estimates happen. Instead, the Relative Pose Error (RPE) is a metric for investigating the local consistency of a SLAM trajectory. RPE aligns the estimated and ground truth pose for a given frame i, and then computes the error of the estimated pose for a frame $j > i$ at a fixed distance farther along the trajectory. We calculate the RPE from frame i to j in translation and rotation (absolute angular error) [6, Sec. 4.2.3]. As [18], we evaluate the RPE over all possible trajectories of a given length, and for different lengths. Nevertheless, instead of calculating the mean of all RPE for a given trajectory length, we report the maximum, the minimum, the first and third quartile, as well as the median.

RPE results. In Figure 3, we show the results for dataset V2_02, where we observe that using our proposed pipeline SPR, with respect to the SP pipeline, leads to: (i) an accuracy improvement of up to 50% in translation and 30% in rotation (based on the maximum improvement on the median of the errors), and, (ii) an average improvement over all trajectory lenghts of 20% in translation and 15% in rotation (for the median errors).

B. Mapping quality

We use the ground truth point cloud for V1 dataset to assess the quality of the mesh by calculating its *accuracy* as defined in [19].

./results/V2_02_medium/traj_relative_errors_boxplots-eps-converted-to.pdf

Fig. 3: Detailed comparison of the state estimation accuracy while using pipeline S, SP, and our proposed approach SPR on different EuRoC datasets.

Performance Metric: Map Accuracy. Comparing a mesh with a dense point cloud can be achieved by generating a point cloud from the mesh itself, and then comparing both point clouds. In our case, we compute a point cloud by sampling the mesh with a uniform density of 10^3 points per square meter. We also register the resulting point cloud to the ground truth point cloud using an iterative closest point algorithm. With the newly registered point cloud, we can compute a cloud to cloud distance to assess the accuracy of the mesh relative to the ground truth point cloud. More specifically, for each point r of the estimated cloud from the mesh $\mathcal{R}$, we search the nearest point in the ground truth cloud $\mathcal{G}$, and compute their Euclidean distance $d_{r\rightarrow\mathcal{G}}$:

$$d_{r\rightarrow\mathcal{G}} = \min_{g\in\mathcal{G}} \|r - g\|_2 \quad \text{for} \quad r \in \mathcal{R}. \tag{7}$$

Table III shows that both the mean and the standard deviation of the distance from the mesh to the ground truth point cloud (eq. (7)) decreases when enforcing structural regularities, as done in the SPR pipeline. On average, each point sampled on the mesh generated by the SPR pipeline is 0.5 cm closer to the ground truth point cloud than the points sampled on the mesh generated by the SP pipeline. Therefore, enforcing structural regularities makes the estimated mesh closer to the real scene.

We also report the accuracy $\mathcal{A}(\tau)$, defined as the fraction of estimated points which are within a distance threshold τ of the ground truth point cloud [19], [20]:

$$\mathcal{A}(\tau) = \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \Big[d_{r\rightarrow\mathcal{G}} < \tau\Big]_I \times 100, \tag{8}$$

where $[P]_I$ is the Iverson bracket, defined as:

$$[P]_I = \begin{cases} 1 & \text{if } P \text{ is true;} \\ 0 & \text{otherwise,} \end{cases}$$

TABLE III: Statistics for the cloud to cloud absolute distance from the mesh to the ground truth point cloud $d_{r\rightarrow\mathcal{G}}$ (eq. (7)) for dataset `V1_01_easy`.

	VIO Type	
$d_{r\rightarrow\mathcal{G}}$ Statistics	SP	SPR (**Proposed**)
Mean $\bar{d}_{\mathcal{R}}$ [cm]	4.9	**4.4**
Standard Deviation $\sigma_{\mathcal{R}}$ [cm]	5.0	**4.6**

?? shows the actual error distributions for $d_{r\rightarrow\mathcal{G}}$, and the mesh accuracy $\mathcal{A}(\tau)$ for different distance thresholds τ, for both the SP and the SPR pipelines respectively. In terms of accuracy values $\mathcal{A}(\tau)$, we can see in **??** that the SPR pipeline consistently achieves more accurate mesh estimates (between 3%-7% better) for distance thresholds $\tau < 10cm$.

As a reminder, the SP pipeline still uses the mesh to detect regularities, but, contrary to the SPR pipeline, it does not enforce the structural regularities on the landmarks.

In **??**, we color-encode each point on the estimated point cloud with the error distances $d_{r\rightarrow\mathcal{G}}$. We can observe that, when we do not enforce structural regularities, significant errors are actually present on the planar surfaces, especially on the walls (**??** top). Instead, when regularities are enforced, the errors on the walls and the floor are reduced (**??** bottom). **TODO:** Remove everything except this and the results in table 3 A closer view on the wall itself, bottom figures (c) and (d) of fig. 1, shows that it is visually clear that adding co-planarity constraints results in smoother walls.

C. Timing

The pipelines S, SP, and SPR differ in that they try to solve an increasingly complicated problem. While the S pipeline does not include neither the 3D landmarks nor the planes as variables in the optimization problem, the SP pipeline includes 3D landmarks, and the pipeline using regularities (SPR) further includes planes as variables. Moreover, the SP has significantly less constraints between the variables than the SPR pipeline. Hence, we can expect that the optimization times for the different pipelines will be each bounded by the other as $t^{opt}_S < t^{opt}_{SP} < t^{opt}_{SPR}$, where t^{opt}_X is the time taken to solve the optimization problem of pipeline X.

Figure 4 shows the time taken to solve the optimization problem for each type of pipeline. Experimentally, we observe that the optimization time follows roughly the expected distribution $t^{opt}_S < t^{opt}_{SP} < t^{opt}_{SPR}$. We also observe that if the number of plane variables is large ($\sim 10^1$), and consequently the number of constraints between landmarks and planes also gets large ($\sim 10^2$), the optimization problem cannot be solved in real-time. For example, for the keyframe index 250 in Figure 4, we can see that a spike is present caused by the detection of multiple planes and landmarks with regularities.

V. Conclusion

In this paper, we have shown a VIO algorithm capable of building a 3D mesh of the scene without relying on extra regularization steps, but instead enforcing structural constraints

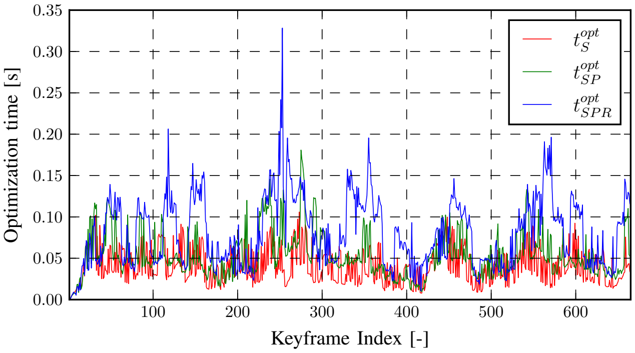

Fig. 4: Comparison of the time to solve the optimization problem for pipeline S, SP, and SPR for dataset `V1_01_easy`.

extracted from the mesh. We have also presented a way to incrementally build a 3D mesh while restricting it to the time-horizon of the optimization problem. Therefore, the mesh spans multiple viewpoints, thereby covering an extended area; yet the size of the mesh remains bounded, allowing for real-time operation.

After evaluation of our approach, we have seen that enforcing co-planarity constraints between landmarks provides more accurate state **TODO:** and mesh estimates than simply ignoring these structural regularities. In particular, the state estimation improves its global consistency by 26% (Absolute Translation Error), while its local consistency improves by up to 50% (Relative Position Error in translation), in scenes with structural regularities. Moreover, our proposed VIO algorithm surpasses in localization accuracy the state-of-the-art in scenes exhibiting structural regularities. We also show that structural constraints improve the accuracy of the mesh by up to 7%.

Finally, while the results are promising, we are not yet enforcing higher level regularities (such as parallelism or orthogonality) between planes. Therefore, these improvements could be even larger, potentially rivaling pipelines enforcing loop-closures.

References

[1] T. Sayre-McCord, W. Guerra, A. Antonini, J. Arneberg, A. Brown, G. Cavalheiro, Y. Fang, A. Gorodetsky, D. McCoy, S. Quilter, F. Riether, E. Tal, Y. Terzioglu, L. Carlone, and S. Karaman, "Visual-inertial navigation algorithm development using photorealistic camera simulation in the loop," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018.

[2] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial SLAM using nonlinear optimization," *Int. J. Robot. Research*, 2015.

[3] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *arXiv preprint arXiv:1708.03852*, 2017.

[4] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 3886–3893, extended arxiv preprint: 1610.03344 (pdf).

[5] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*. IEEE, 2015.

[6] A. Rosinol, "Densifying Sparse VIO: a Mesh-based approach using Structural Regularities." Master's thesis, ETH Zurich, 2018-09-14.

[7] Q. Pan, G. Reitmayr, and T. Drummond, "Proforma: Probabilistic feature-based on-line rapid model acquisition." in *BMVC*, vol. 2. Citeseer, 2009, p. 6.

[8] W. N. Greene and N. Roy, "Flame: Fast lightweight mesh estimation using variational smoothing on delaunay graphs," in *Int. Conf. Comput. Vis. (ICCV)*, 2017.

[9] L. Teixeira and M. Chli, "Real-Time Mesh-based Scene Estimation for Aerial Inspection," in *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems(IROS)*, 2016.

[10] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, 2017.

[11] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, Feb. 2001.

[12] L. Dietz, "Directed factor graph notation for generative models," *Max Planck Institute for Informatics, Tech. Rep*, 2010.

[13] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4605–4611.

[14] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Research*, vol. 35, pp. 1157–1163, 2015.

[15] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial SLAM using nonlinear optimization," in *Robotics: Science and Systems (RSS)*, 2013.

[16] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, Apr. 2007, pp. 3565–3572.

[17] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," *Memory*, vol. 10, p. 20, 2018.

[18] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2012.

[19] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.