根据三角剖分加入结构化特征

Incremental Visual-Inertial 3D Mesh Generation with Structural Regularities

Antoni Rosinol[1], Torsten Sattler[2], Marc Pollefeys[3], Luca Carlone[1]

***Abstract*— Visual-Inertial Odometry (VIO) algorithms typically rely on a point cloud representation of the scene that does not model the topology of the environment. A 3D mesh instead offers a richer, yet lightweight, model. Nevertheless, building a 3D mesh out of the sparse and noisy 3D landmarks triangulated by a VIO algorithm often results in a mesh that does not fit the real scene. In order to regularize the mesh, previous approaches decouple state estimation from the 3D mesh regularization step, and either limit the 3D mesh to the current frame [1], [2] or let the mesh grow indefinitely [3], [4]. We propose instead to tightly couple mesh regularization and state estimation by detecting and enforcing *structural regularities* in a novel factor-graph formulation. We also propose to incrementally build the mesh by restricting its extent to the time-horizon of the VIO optimization; the resulting 3D mesh covers a larger portion of the scene than a per-frame approach while its memory usage and computational complexity remain bounded. We show that our approach successfully regularizes the mesh, while improving localization accuracy, when structural regularities are present, and remains operational in scenes without regularities.**

***Index Terms*— SLAM, Vision-Based Navigation, Sensor Fusion.**

Supplementary Material

https://www.mit.edu/~arosinol/research/struct3dmesh.html

I. Introduction

Recent advances in VIO are enabling a wide range of applications, ranging from virtual and augmented reality to agile drone navigation [5]. While VIO methods can deliver accurate state estimates in real-time, they typically provide a sparse map of the scene. In particular, feature-based methods [6]–[9] produce a point cloud that is not directly usable for path planning or obstacle avoidance. In those cases, a denser map is built subsequently, e.g., by using (multi-view) stereo algorithms [10], [11]. Alternatively, direct every-pixel methods estimate denser point clouds online [12]–[14]. Nevertheless, these algorithms rely on GPUs which consume relatively high amounts of power, making them impractical for computationally-constrained systems such as micro aerial vehicles or smartphones. Furthermore, these models typically decouple trajectory estimation and mapping, resulting in a loss of accuracy [15], and produce representations that are expensive to store and manipulate. Ideally one would like to use a map representation that (i) is lightweight to compute and store, (ii) describes the topology of the environment, and (iii) couples state estimation and mapping, allowing one to improve the other and vice versa. A 3D mesh representation is lightweight, while it provides information about the topology of the scene. Moreover, a 3D mesh allows for extracting the structure of the scene, which can potentially be used to improve simultaneously the accuracy of the pose estimates and the mesh itself, thereby coupling state estimation and mapping.

Recent approaches have tried to avoid the caveats of every-

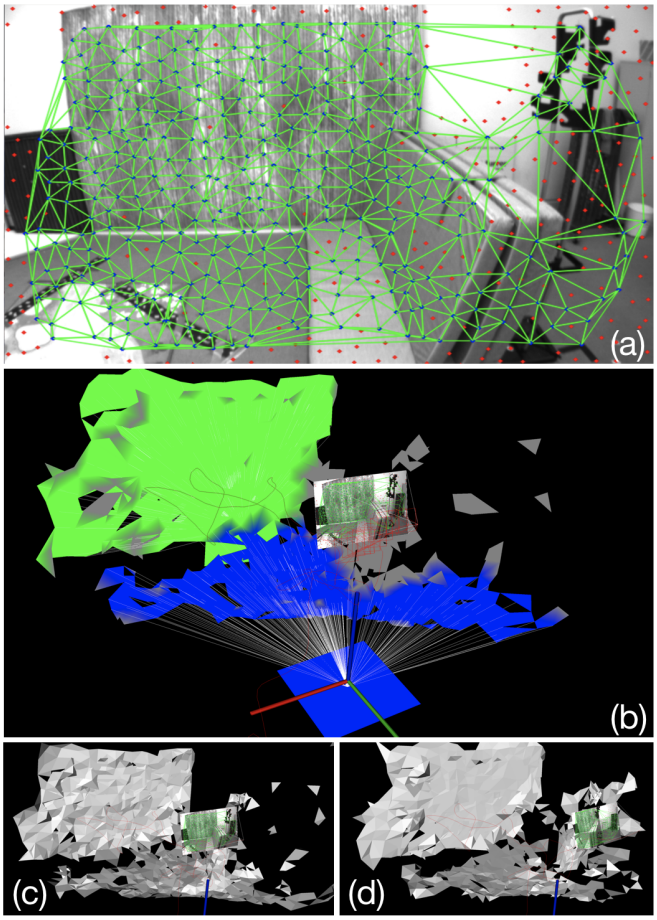

Fig. 1: We propose a VIO pipeline that incrementally builds a 3D mesh of the environment starting from a 2D Delaunay triangulation of keypoints (a). We also detect and enforce *structural regularities*, c.f. (b) planar walls (green) and floor (blue). The bottom row compares the mesh constructed (c) without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

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

[2]T. Sattler is with the Department of Electrical Engineering, Chalmers University of Technology, Sweden. This work was done while Torsten was at ETH Zürich, torsat@chalmers.se

[3]M. Pollefeys is with the Department of Computer Science, ETH Zürich, and with Microsoft, Switzerland, marc.pollefeys@inf.ethz.ch

This work was partially funded by ARL DCIST CRA W911NF-17-2-0181, Lincoln Laboratory, and the Zeno Karl Schindler foundation.

pixel methods by using a 3D mesh over the set of sparse 3D landmarks triangulated by a VIO pipeline. Nevertheless, these approaches perform regularization of the mesh as a post-processing step – decoupling state estimation and mesh generation – and work on a per-frame basis [1], [2]. Our approach instead tightly couples the 3D mesh generation and the state estimation by enforcing structural constraints in a factor-graph formulation, which allows for joint mesh regularization and pose estimation. We also maintain the 3D mesh over the receding horizon of the VIO's fixed-lag optimization problem, thereby spanning multiple frames and covering a larger area than the camera's immediate field-of-view.

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. We also propose to use the 3D mesh to *detect and enforce structural regularities* in the optimization problem, thereby improving the accuracy of both the state estimation and the mesh at each iteration, while circumventing the need for an extra regularization step for the mesh. In particular, we extract co-planarity constraints between landmarks (Fig. 1), and show that we can detect these structural priors in a non-iterative way, contrary to RANSAC-based approaches [16]. Overall, our approach runs in real-time by using a single CPU core. Moreover, we do not rely on sensors such as LIDAR or RGB-D cameras, instead we use a (stereo) monochrome camera.

Finally, we provide an extensive experimental evaluation on the EuRoC dataset [17], where we compare the proposed VIO approach against state-of-the-art methods. Our evaluation shows that (i) the proposed approach produces a lightweight representation of the environment that captures the geometry of the scene, (ii) leveraging structural regularities improves the state and map estimation, surpassing the state-of-the-art when structural regularities are present, while (iii) performing on-par with standard VIO methods in absence of regularities.

II. Approach

We consider a stereo visual-inertial system and adopt a *keyframe*-based approach [7]. 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 [18], [19]. The 3D mesh is further used to extract structural regularities in the scene that are then encoded as constraints in the back-end optimization.

A. Front-end

Our front-end has the same components as a keyframe-based indirect visual-inertial odometry pipeline [7], [20], 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 [21, 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: using a sparse point cloud from VIO to create a 3D mesh 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 landmarks' positions are being updated at each optimization step. Therefore, we avoid performing a 3D tetrahedralisation from the landmarks, which would require expensive algorithms, such as space carving [22]. Instead, we perform a 2D Delaunay triangulation only over the tracked keypoints in the latest frame, as shown in Fig. 1 (a); and project the 2D triangulation in 3D using the fact that each tracked keypoint has a 3D landmark associated (Fig. 1 (b)). For the first frame, no keypoint is yet tracked, hence no 3D mesh is generated.

The Delaunay triangulation maximizes the minimum angle of all the angles of the triangles in the triangulation; thereby avoiding triangles with extremely acute angles. Since we want to promote triangles that represent planar surfaces, this is a desirable property, as it will promote near isotropic triangles that cover a good extent of a potentially planar surface. Nevertheless, having an isotropic triangle in 2D does not guarantee that the corresponding triangle in 3D will be isotropic, as one of the vertices could be projected far from the other two. Furthermore, a triangle in the 2D image will result in a 3D triangle independently of whether it represents an actual surface or not. We deal with some of these misrepresentative faces of the mesh by using simple geometric filters that we detail in [21, Sec. 3.2.1].

2) 3D Mesh Propagation: While some algorithms update the mesh for a single frame [1], [2], we attempt to maintain a mesh over the receding horizon of the fixed-lag smoothing optimization problem (Section II-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. The 3D mesh propagation can be decomposed in two parts.

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. Unfortunately, most of the keypoints' positions on the 2D image change each time the camera moves. Hence, we re-compute a 2D Delaunay triangulation from scratch over the keypoints of the current frame. We can then project all the 2D triangles to 3D mesh faces, since we are keeping track of the landmark associated to each keypoint.

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. At the same time, when a landmark is marginalized from the optimization, we remove any face in the 3D mesh that has the landmark as a vertex. This operation is not without caveats. For example, the removed landmark might be at the center of a wall, thereby leaving a hole when surrounding faces of the mesh are deleted. While we did not attempt to solve this issue, the problem usually appears on the portion of the mesh that is not currently visible by the camera. Also, we do not explicitly deal with the problem of occlusions.

3) Regularity Detection: By reasoning in terms of the triangular faces of the mesh, we can extract the geometry in the scene in a non-iterative way (unlike RANSAC approaches). In particular, we are interested in co-planarity regularities between landmarks, for which we need to first find planar surfaces in the scene. In our approach, we only detect planes that are either vertical (i.e. walls) or horizontal (i.e. floor, tables), which are structures commonly found in man-made environments. Fig. 1 (b) shows the faces of the mesh associated to a vertical wall in green, while the blue faces correspond to the floor. To detect horizontal planes, we cluster the faces of the mesh with vertical normals, and then build a 1D histogram of the height of the vertices. After smoothing the histogram with a Gaussian filter, the resulting local maximums of the histogram correspond to predominant horizontal planes. Among these planes, we take the candidates with the most inliers (above a minimum threshold of 20 faces). To detect vertical planes, we cluster the faces of the mesh which have a horizontal normal. Then, we build a 2D histogram; where one axis corresponds to the shortest distance from the plane of the 3D face to the world origin[1], and the other axis corresponds to the azimuth of the normal with respect to the vertical direction[2]. Candidate selection is done the same way as in the horizontal case.

4) Data Association: With the newly detected planes, we still need to associate which landmarks are on each plane. For this, we use the set of landmarks of the 3D faces that voted for the given plane in the original histogram. Once we have a new set of planes detected, we still need to check if these planes are already present in the optimization problem to avoid duplicated plane variables. For this, we simply compare the normals and distances to the origin of the plane to see if they are close to each other.

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

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

We will only encode in the optimization 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 encoding the rest of the landmarks $\mathcal{S}_t = \mathcal{L}_t \setminus \Lambda_t$ by using a structureless approach, as defined in [8, Sec. VII], which circumvents the need to add the landmarks' positions as variables in the optimization. This allows trading-off accuracy for speed, since the optimization's 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 II-B.5. 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 in a time-horizon of length 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 variables to a given window of time Δ_t, we will make the optimization problem more tractable and solvable in real-time.

2) Measurements: The input for our system consists of 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 l_c (where the index c stands for 'constrained' landmark), and the landmarks that will remain as structureless l_s (where the index 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 variables $\mathcal{X}_t$ (Eq. (2)) using the set of measurements $\mathcal{Z}_t$ (Eq. (3)). Using standard independence assumptions between measurements, we arrive to the following formulation where we grouped the different

[1]The world origin corresponds to the first estimated pose of the IMU.

[2]Since gravity is observable via the IMU, we have a good estimate of what the vertical direction is.

terms in factors ϕ:

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

where we apply the Bayes rule in (a), and ignore the normalization factor since it will not influence the result (Section II-B.4). Eq. (4a) corresponds to the prior information we have about $\mathcal{X}_t$. The factor ϕ_0 represents a prior on the first state of the optimization's time-horizon. The following terms in Eq. (4a) encode regularity factors ϕ_R between constrained landmarks l_c and planes π. 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 (Section II-A.4). In Eq. (4b), we have the factor corresponding to the IMU measurements which depends only on the consecutive keyframes $(i, j) \in \mathcal{K}_t$. Eq. (4c) encodes the factors corresponding to the camera measurements. We add a projection factor ϕ_{l_c} for each observation of a constrained landmark l_c, where we denote by $\Lambda_t(i) \subseteq \Lambda_t$ the set of constrained landmarks seen by keyframe i. Finally, we add structureless factors ϕ_{l_s} for each of the landmarks $l_s \in \mathcal{S}_t$; note that these factors depend on the subset of keyframes that observe l_s, which we denote by $\mathcal{K}_t(l_s) \subseteq \mathcal{K}_t$. In Fig. 2, we use the expressiveness of factor graphs [23], [24] to show the dependencies between the variables in Eq. (4).

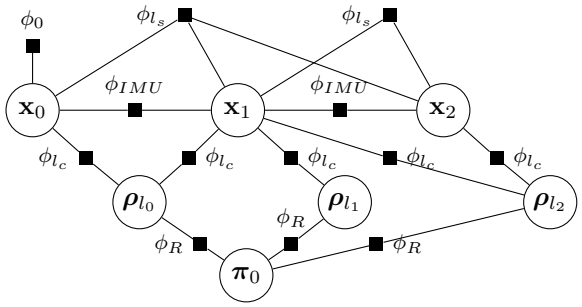

Fig. 2: VIO factor graph combining Structureless (ϕ_{l_s}), Projection (ϕ_{l_c}) and Regularity (ϕ_R) factors (SPR). The factor ϕ_R encodes relative constraints between a landmark l_i and a plane π_0.

4) MAP Estimation: Since we are only interested in the most likely $\mathcal{X}_t$ given the measurements $\mathcal{Z}_t$, we calculate the *maximum a posteriori* (MAP) estimator $\mathcal{X}_t^{\text{MAP}}$. Minimizing the negative logarithm of the posterior probability in Eq. (4) (under the assumption of zero-mean Gaussian noise) leads to a nonlinear least-squares problem:

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

where $\mathbf{r}$ represents the residual errors, and $\boldsymbol{\Sigma}$ the covariance matrices. We refer the reader to [8, 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 $\mathbf{r}_{\text{IMU}}$, $\mathbf{r}_{l_s}$. For the projection factors ϕ_{l_c}, we use a standard monocular and stereo reprojection error as in [19].

5) Regularity Constraints: For the regularity residuals $\boldsymbol{r}_R$, we use a co-planarity constraint between a landmark $\boldsymbol{\rho}_{l_c} \in \mathbb{R}^3$ and a plane $\boldsymbol{\pi} = \{\boldsymbol{n}, d\}$, 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 $d \in \mathbb{R}$ is the distance to the world origin: $\mathbf{r}_R = \boldsymbol{n} \cdot \boldsymbol{\rho}_{l_c} - d$. This plane representation is nevertheless an over-parametrization that will lead to a singular information matrix. This is not amenable for Gauss-Newton optimization, since it leads to singularities in the normal equations [25]. To avoid this 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 of the normals in S^2 [8]. In other words, we rewrite the residuals as:

$$\mathbf{r}_R(\boldsymbol{v}, d) = \mathcal{R}_{\boldsymbol{n}}(\boldsymbol{v})^\mathsf{T} \cdot \boldsymbol{\rho} - d \tag{5}$$

and optimize with respect to the minimal parametrization $[\boldsymbol{v}, d] \in \mathbb{R}^3$. This is similar to [25], but we work on the manifold S^2 instead of adopting a quaternion parametrization. Note that a single co-planarity constraint, as defined in Eq. (5), 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.

III. 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 [17], 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 little (planar) regularities. The *Vicon Room* (V) is similar to an office room where walls, floor, and ceiling are visible, as well as other planar surfaces (boxes, stacked mattresses).

Compared techniques. To assess the advantages of our proposed approach, we compare three formulations that build one on top of another. First, we denote as **S** the approach that uses only Structureless factors (ϕ_{l_s}, in Eq. (4c)). Second, we denote as **SP** the approach that uses 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 (ϕ_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 [26], MSCKF [6], ROVIO [20], VINS-MONO [18], and SVO-GTSAM [8], using the reported values in [27]. Note that these algorithms use a monocular camera, while we use a stereo camera. Therefore, while [27] 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. No algorithm uses loop-closure.

A. Localization Performance

Absolute Translation Error (ATE). The 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 [21, 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 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 pipelines SP and SPR gracefully downgrade 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 pipelines 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. 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, SPR decreases the median APE by 27.6% compared to the SP approach for dataset `V1_02`, which has multiple planes.

Table II shows that the SPR approach achieves the best results when compared with the state-of-the-art on datasets with structural regularities, such as in datasets `V1_01` and `V1_02`, 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`, and a 26% improvement in dataset `V1_02` 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` and `MH_05`.

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_01_easy	13.7	15.0	12.4	15.0	**10.7**	**14.5**
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: RMSE of the state-of-the-art techniques (reported values from [27]) compared to our proposed SPR pipeline, on the EuRoC dataset. A cross ($\times$) 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	16	42	21	27	**5**	**14**
MH_02	22	45	25	**12**	**3**	13
MH_03	24	23	25	**13**	**12**	21
MH_04	34	37	49	23	**13**	**22**
MH_05	47	48	52	35	**16**	**23**
V1_01	9	34	10	**7**	**7**	**6**
V1_02	20	20	**10**	**10**	11	**7**
V1_03	24	67	**14**	**13**	$\times$	17
V2_01	13	10	12	**8**	**7**	**8**
V2_02	16	16	14	**8**	$\times$	**10**
V2_03	29	113	**14**	**21**	$\times$	27

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, RPE is a metric for investigating the local consistency of a 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) [21, Sec. 4.2.3]. As [28], we evaluate the RPE over all possible trajectories of a given length, and for different lengths.

Fig. 3 shows 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 average improvement of the median of the RPE over all trajectory lengths of 20% in translation and 15% in rotation, and (ii) a maximum accuracy improvement of 50% in translation and 30% in rotation of the median of the RPE.

B. Mapping quality

We use the ground truth point cloud for dataset `V1` to assess the quality of the 3D mesh by calculating its *accuracy*,

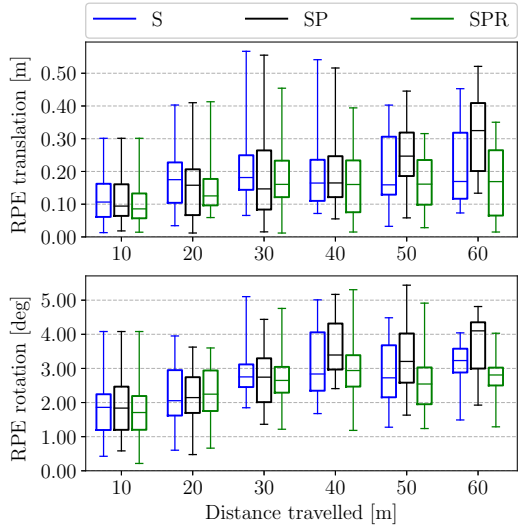

Fig. 3: Boxplots of the RPE on dataset `V2_02` for the approaches S, SP, and SPR (proposed).

as defined in [10]. To compare the mesh with the ground truth point cloud, we compute a point cloud by sampling the mesh with a uniform density of 10^3 points$/m^2$. We also register the resulting point cloud to the ground truth point cloud. In Fig. 4, we color-encode each point r on the estimated point cloud with its distance to the closest point in the ground-truth point cloud $\mathcal{G}$ ($d_{r\to\mathcal{G}}$). We can observe that, when we do not enforce structural regularities, significant errors are actually present on the planar surfaces, especially on the walls (Fig. 4 top). Instead, when regularities are enforced, the errors on the walls and the floor are reduced (Fig. 4 bottom). A closer view on the wall itself (Fig. 1(c)-(d)) provides an illustrative example of how 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 optimization 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, SP has significantly less factors between 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 $t_S^{opt} < t_{SP}^{opt} < t_{SPR}^{opt}$, where t_X^{opt} is the time taken to solve the optimization problem of pipeline X.

Fig. 5 shows the time taken to solve the optimization problem for each type of pipeline. We observe that the optimization time follows roughly the expected distribution. We also notice 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 (see keyframe index 250 in Fig. 5). This can be avoided by restricting the number of planes in the optimization. Finally,

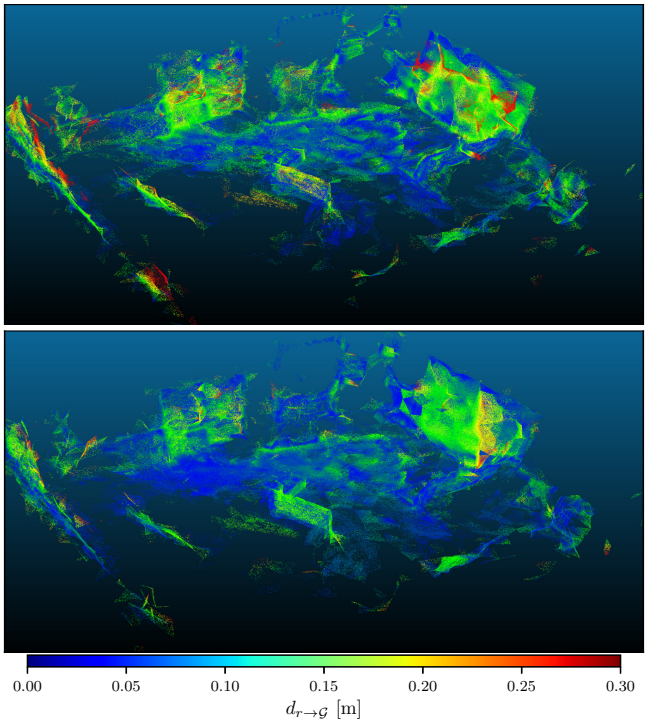

Fig. 4: Point cloud sampled from the estimated 3D mesh color-encoded with the distance to the ground truth point cloud (`V1_01`), for SP approach (top) and SPR (bottom).

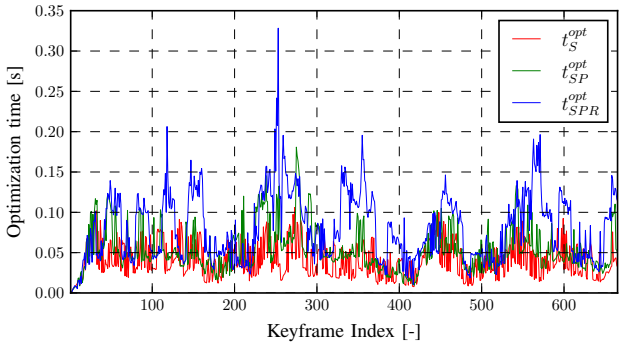

Fig. 5: Comparison of the time to solve the optimization problem for pipeline S, SP, and SPR for dataset `V1_01`.

the SPR pipeline has the overhead of generating the mesh. Nevertheless, it takes just 8ms per frame.

IV. Conclusion

We present a VIO algorithm capable of incrementally building a 3D mesh of the scene restricted to a receding time-horizon. Moreover, we show that we can improve the state estimation and mesh by enforcing structural regularities present in the scene. Hence, we provide a tightly coupled approach to regularize the mesh and improve the state estimates simultaneously.

Finally, while the results presented 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.

提出了一种VIO算法，它能够增量地构建场景的三维网格，限制在一个滚动的时间内。
通过加强场景中的结构规则来改进状态估计和网格。因此，这是一种紧耦合的方法来规范化网格并同时改进状态估计。
但我们还没有在平面之间强制执行更高级别的规则（如平行性或正交性）。未来有可能与强制loop closure的pipeline相匹敌。

REFERENCES

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

[2] 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.

[3] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, *et al.*, "Detailed real-time urban 3d reconstruction from video," *International Journal of Computer Vision*, vol. 78, no. 2-3, pp. 143–167, 2008.

[4] V. Litvinov and M. Lhuillier, "Incremental solid modeling from sparse and omnidirectional structure-from-motion data," in *British Machine Vision Conference*, 2013.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] S. Pillai, S. Ramalingam, and J. Leonard, "High-performance and tunable stereo reconstruction," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016.

[12] R. A. Newcombe and A. J. Davison, "Live dense reconstruction with a single moving camera," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, June 2010, pp. 1498–1505.

[13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2320–2327.

[14] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. Davison, and A. Fitzgibbon, "KinectFusion: Real-time dynamic 3D surface reconstruction and interaction," in *SIGGRAPH*, Aug. 2011, p. 23.

[15] L. Platinsky, A. J. Davison, and S. Leutenegger, "Monocular visual odometry: Sparse joint optimisation or dense alternation?" in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 5126–5133.

[16] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, "An improved RANSAC for 3D point cloud plane segmentation based on normal distribution transformation cells," *Remote Sensing*, vol. 9, no. 5, p. 433, 2017.

[17] 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.

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

[19] 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).

[20] 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.

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

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

[23] 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.

[24] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017. [Online]. Available: http://dx.doi.org/10.1561/2300000043

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

[26] 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.

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

[28] 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.