Towards Robust Monocular Visual Odometry for Flying Robots on Planetary Missions

M. Wudenka[1,2], M. G. Müller[1,3], N. Demmel[2], A. Wedler[1], R. Triebel[1,2], D. Cremers[2], W. Stürzl[1]

Abstract— In the future, extraterrestrial expeditions will not only be conducted by rovers but also by flying robots. The technical demonstration drone Ingenuity, that just landed on Mars, will mark the beginning of a new era of exploration unhindered by terrain traversability. Robust self-localization is crucial for that. Cameras that are lightweight, cheap and information-rich sensors are already used to estimate the ego-motion of vehicles. However, methods proven to work in man-made environments cannot simply be deployed on other planets. The highly repetitive textures present in the wastelands of Mars pose a huge challenge to descriptor matching based approaches.

In this paper, we present an advanced robust monocular odometry algorithm that uses efficient optical flow tracking to obtain feature correspondences between images and a refined keyframe selection criterion. In contrast to most other approaches, our framework can also handle rotation-only motions that are particularly challenging for monocular odometry systems. Furthermore, we present a novel approach to estimate the current risk of scale drift based on a principal component analysis of the relative translation information matrix. This way we obtain an implicit measure of uncertainty. We evaluate the validity of our approach on all sequences of a challenging real-world dataset captured in a Mars-like environment and show that it outperforms state-of-the-art approaches. The source code is publicly available at:
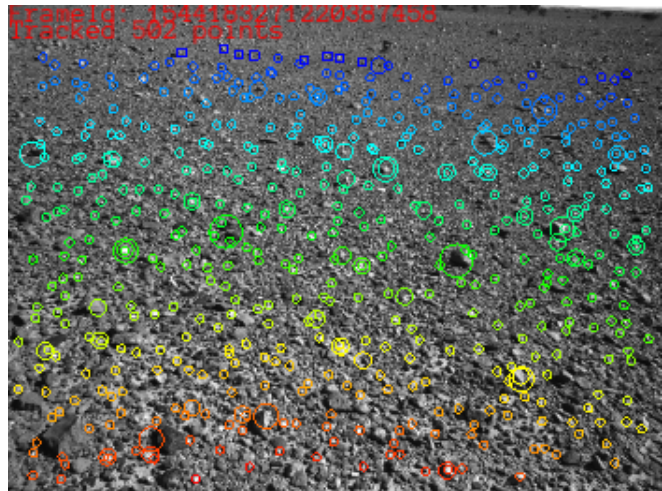
https://github.com/DLR-RM/granite

I. INTRODUCTION

Since 1997, selected regions of Mars have successfully been explored by robotic vehicles, performing more and more tasks autonomously. Essential for their autonomy is the ability of self-localization aided by visual odometry (VO), i.e. dead reckoning ego-motion from an image stream. Robust and accurate state estimation becomes even more important for flying robots, since they have to work fully autonomously. This is the case for future mission concepts [2] and also for the current NASA Mars 2020 mission and its flying robot Ingenuity. The VO of Ingenuity is designed for limited computational resources and flight durations of about $90\,\mathrm{s}$ on flat terrain [3]. Being able to robustly navigate a flying robot in any environment on Mars would enable fast exploration, independent of the traversability of the terrain. Flying robots can act as scouts for rovers and also discover and explore areas where rovers cannot reach. Furthermore, flying swarms that increase the overall fault tolerance through redundancy are conceivable, although increasing the need for cheap and lightweight drones.
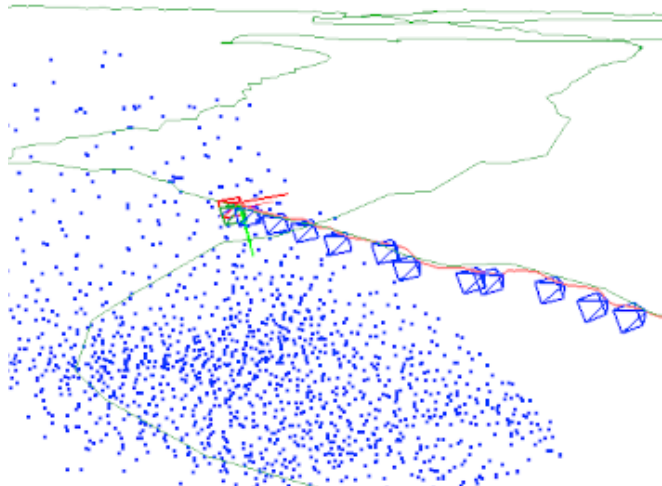

(a) Landmarks projected onto the image with their estimated inverse distance color coded. Circles illustrate the feature patch size at highest resolution, resulting in larger circles for features initialized at lower resolutions (see Sec. IV-A).

(b) Estimated trajectory (red) and the ground truth trajectory based on GPS data (green) are shown, as well as current camera pose (red) and the keyframe camera poses (blue).

Fig. 1: Our robust monocular odometry at work in a Mars-analogue environment (example from sequence D-3 of the MADMAX dataset [1]).

In the last decades, academic research has brought up various different solution strategies to tackle the visual odometry problem. However, when evaluating recent state-of-the-art systems on datasets which resemble Mars-like environ-

[1] Institute of Robotics and Mechatronics, German Aerospace Center (DLR). `{firstname.lastname}@dlr.de`
[2] Computer Vision Group, Department of Informatics, Technical University of Munich, Germany
[3] Autonomous Systems Lab, ETH Zurich, Switzerland

ments such as the one recorded in the Morocco-Acquired Dataset of Mars-Analogue eXploration (MADMAX) [1], we especially found that methods relying on descriptor-based feature matching lack robustness. While direct methods seem promising, they still require significantly more computational resources and are overall not as mature. They heavily rely on a motion model as initialization for the optimization algorithm. In our experiments we found that indirect methods based on Kanade-Lucas-Tomasi (KLT) optical flow tracking [4] for data association balance robustness, accuracy and computational demands well.

For visual odometry, a monocular setup is especially challenging as the scene scale cannot be observed and thus, the gauge freedom has seven degrees; one more than that of a stereo setup. Hence, there is one more dimension where drift accumulates. Moreover, during pure rotation, no feature positions can be measured by triangulation. On the other hand, a single camera as main sensor is appealing as being lightweight and cheap. Furthermore, stereo setups effectively degrade to monocular when flying at high altitudes, as the ratio of stereo baseline and feature distance becomes small. In this paper, we therefore describe how we generalize the VO described in [5] to monocular setups. Moreover, we show that an enhanced feature-point and keyframe selection can improve the robustness on repetitively textured environments, as present in the planetary-like MADMAX dataset.

Crucial to the robustness of the whole robotic system is knowledge of the uncertainties caused by each component. Consequently, we explore probabilistically motivated methods to estimate the current tracking quality and to detect the risk of scale drift. Such information is relevant for decision making and path planning.

To summarize, our contributions are:

- Generalization of the VO presented in [5] to monocular setups (Sec. IV-C).
- Handling of rotation-only motion (Sec. IV-B).
- Increased robustness in environments with highly repetitive textures as present in planetary exploration as well as to frames drops and large motions between frames by means of improved feature initialization (Sec. IV-A) and keyframe selection (Sec. IV-D).
- A novel probabilistic measure as indicator for scale drift (Sec. V).
- Comparison of our system with other state of the art methods on a dataset relevant to planetary exploration (Sec. VI).

Fig. 1 shows an example of our visual odometry running on a sequence of the MADMAX dataset [1].

II. RELATED WORK

In the last decades, various systems for tracking a camera through VO were developed. They all differ in the chosen data association method, optimization strategy and supported sensor modalities: monocular, stereo and/or the fusion of additional sensors such as Inertial Measurement Units (IMUs). Additionally, some methods aim for global consistency and therefore are called Simultaneous Localization and Mapping

(SLAM) systems. While modern VO systems achieve very high accuracy through bundle adjustment, they do not provide an easy to interpret measure of confidence. Therefore, we split our related work discussion in two parts. First, we present methods relevant to camera tracking and research about uncertainty estimation later.

Monocular Odometry: Similar to our approach, the visual-inertial odometry of the Mars Helicopter [3] uses KLT tracking [4] for data association. Moreover, an Extended Kalman Filter (EKF) is used to process landmarks that are all assumed to lie on a flat ground plane. Thus, it cannot fly over hilly terrain. The scene scale is obtained via an altimeter pointing nadir. In [6] an experimental system is presented that relaxes the flat surface assumption. Since Parallel Tracking and Mapping (PTAM) [7] has proven that windowed geometric bundle adjustment can run in real time on modern hardware, many new smoothing systems were presented reaching new levels of accuracy. One of the most elaborate frameworks is ORB-SLAM3 [8]. It uses descriptor based matching and optimizes a local window of the covisibility graph, as well as a global pose graph in parallel. ORB-SLAM3 can handle monocular, monocular-inertial, stereo and stereo-inertial setups. VINS-Mono [9] is the most similar system to ours. It uses KLT tracking and fixed-lag smoothing in the frontend to fuse visual and inertial measurements. The odometry described in [5], named "Basalt", uses KLT tracking as well. Basalt can handle stereo and stereo-inertial setups, but not monocular ones. Because of its capabilities and features, which we will further summarize in Sec. IV, we base our approach on it. Although Basalt and VINS-Mono have a lot in common, according to Table I in [5], Basalt performs better than VINS-Fusion, the stereo version of VINS-Mono. In the seminal work of [10], a direct but sparse method was presented with noticeable performance in situations where indirect methods show problems, as in the presence of motion blur. However, the unstable map initialization and the reliance on a motion model for initializing the optimization algorithm hinders a deployment in our domain.

Uncertainty Estimation: Methods, that use Gauss-Newton optimization, compute the Fisher information matrix. Even though it encodes the uncertainty in the current state estimate, it is not straight forward using it for decision making. Kuo et al. [11] suggested to use negative entropy, a scalar value describing the tracking quality of the current frame to aid keyframe selection. We take this formulation a step further and investigate the tracking quality of the current optimization window to detect the risk of scale drift.

III. NOTATION

Throughout this paper, we express scalars as light lowercase letters a, vectors as bold lowercase letters $\boldsymbol{b}$ and matrices as bold uppercase letters $\boldsymbol{C}$. Elements of the Special Euclidean Group $SE(3)$ are denoted as $\boldsymbol{T}_i^j$ so that they transform a point in coordinate frame i into j. In that context index 'w' denotes the world reference frame. The matrix
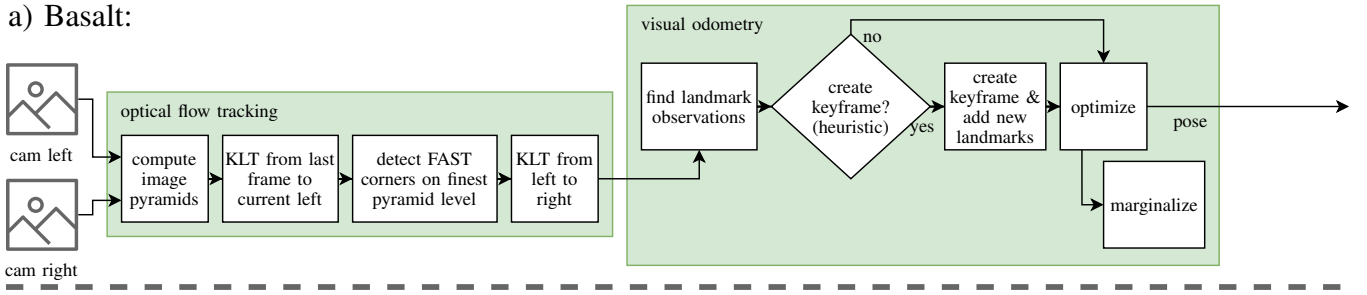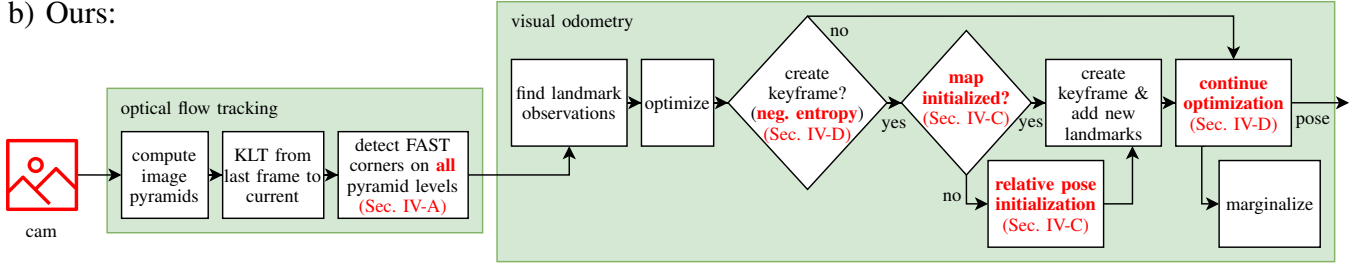
Fig. 2: Flow chart that compares a) the original stereo-based Basalt-VO [5] with b) our monocular approach. Modifications and additions are highlighted in red.

representation of such a rigid-body transformation is

$$T = \begin{bmatrix} R & t \\ \mathbf{0}_{1,3} & 1 \end{bmatrix}. \tag{1}$$

The operators $\mathrm{trans}(T) = t$ and $\mathrm{rot}(T) = R$ extract the translational part $t \in \mathbb{R}^3$ and rotational part $R \in SO(3)$ from the transformation, respectively. The operator $[\cdot]_\times$ converts the tangent vectors $w \in \mathbb{R}^3$ (Rodrigues vectors) to elements of the Lie algebra $\mathfrak{so}(3)$:

$$[\boldsymbol{w}]_\times = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \tag{2}$$

The exponential map that converts elements from Lie algebra to the manifold $SO(3)$ is denoted $\exp : \mathfrak{so}(3) \to SO(3)$.

Similarly to Rodrigues vectors, the elements of the Lie algebra $\mathfrak{se}(3)$ can be expressed as so called twist vectors, i.e. pose increments $\boldsymbol{\xi} \in \mathbb{R}^6$. To convert a twist vector from the tangent space around one transformation to the tangent space of another one the adjoint action is defined as

$$\mathrm{Adj}_{\boldsymbol{T}} = \begin{bmatrix} R & [t]_\times R \\ \mathbf{0}_{3,3} & R \end{bmatrix}. \tag{3}$$

IV. MONOCULAR ODOMETRY

Basalt, as described in [5] is a stereo and stereo-inertial odometry that minimizes the geometric reprojection error. Utilizing sparse KLT optical flow tracking, it estimates the $SE(2)$ transform of FAST corners [12] from the previous left camera image to the current and from the current left camera image to the image of the right camera. To achieve robustness at the occurrence of large optical flow, a pyramidal approach is used, tracking feature patches from the coarsest

to finest level. For outlier detection, every feature is tracked forward and backward. Only features that return to the starting point are considered valid. To estimate the current camera pose a fixed-lag smoothing [13] strategy on a sparse set of keyframes is applied. Two feature points $\in \mathcal{P}$ in two keyframes $\in \mathrm{obs}(i)$ connected by KLT-tracking are considered observations of the same landmark i at pixel coordinates z_{it}. Every landmark is hosted by one keyframe $h(i)$ and expressed in the corresponding coordinate frame. Landmarks are parameterized as bearing vectors and inverse distances d. An initial inverse distance is determined by triangulation. Keyframe poses, landmark bearing vectors and inverse distances are optimized jointly in a sliding window (fixed-lag smoothing). Old keyframes are marginalized into the prior E_m to keep the problem size bounded. Thus, Basalt optimizes the cost function

$$E = E_\mathrm{reproj} + E_\mathrm{m} \tag{4}$$

$$E_\mathrm{reproj} = \sum_{\substack{i \in \mathcal{P} \\ t \in \mathrm{obs}(i)}} \boldsymbol{r}_{it}^\top \Sigma_{it}^{-1} \boldsymbol{r}_{it} \tag{5}$$

$$\boldsymbol{r}_{it} = \boldsymbol{\pi}\left(\boldsymbol{T}_{h(i)}^t \boldsymbol{l}_i\right) - \boldsymbol{z}_{it}. \tag{6}$$

$\boldsymbol{T}_{h(i)}^t$ is the relative pose from host keyframe to observing keyframe, $\boldsymbol{l}_i$ the landmark position in front of the host frame and $\boldsymbol{\pi}$ the camera projection function. Basalt also features a globally consistent mapping layer, which is not considered in this paper.

We chose to base our work on Basalt for the following reasons:

- Robustness: Basalt shows robust tracking results on the MADMAX dataset. The robustness of KLT-tracking in space exploration is also demonstrated by [3].
- Computational efficiency: Sparse optical flow tracking and fixed-lag smoothing well balance performance and

TABLE I: Tracking failure count per MADMAX dataset sequence [1]. †: FAST corners are only searched for in the finest level; ‡: the current frame is made keyframe.

	A-0	A-1	A-2	A-3	A-4	A-5	A-6	B-0	B-1	B-2	B-3	B-4	B-5	B-6	B-7	C-0	C-1	C-2
Ours	1	2	1	1	6	0	8	5	0	5	0	0	0	0	0	0	3	4
Ours †	1	2	2	1	7	1	10	5	0	4	1	0	0	0	0	3	3	3
Ours ‡	13	4	1	3	60	7	99	62	14	7	0	0	1	1	0	12	7	10

	D-0	D-1	D-2	D-3	D-4	E-0	E-1	E-2	F-0	F-1	F-2	F-3	F-4	F-5	G-0	G-1	G-2	H-0
Ours	2	3	2	8	7	4	5	2	0	0	2	1	3	2	1	0	8	0
Ours †	2	3	2	8	10	5	6	2	0	0	2	1	3	2	2	0	8	1
Ours ‡	2	17	4	12	18	11	19	2	2	10	4	6	26	10	10	20	36	1

computational demands.

- General camera models: Basalt uses bearing vectors. So it can be used with arbitrary camera models, allowing us to use wide-angle lenses.
- Code quality: Basalt is well documented and unit tested.

However, Basalt does not meet all of our requirements for lightweight flying robots for space exploration:

- Monocular systems are not supported.
- Rotation-only motion cannot be tracked.
- The keyframe selection heuristics are hard to tune and hinder robust operation.
- No uncertainty measure is provided.

In the following sections, we detail our improvements to Basalt. Firstly, we explain and evaluate how our method increases robustness by improving the feature point detection (Sec. IV-A). Secondly, we introduce tracking of rotation-only motion (Sec. IV-B) into the pipeline. Subsequently, we present our map initialization for monocular configuration (Sec. IV-C) and robustified keyframe selection (Sec. IV-D). The differences between Basalt and our system are highlighted in Fig. 2.

A. Feature Detection at Different Scale Levels

We found that tracking becomes unstable with high image resolutions when features are only detected at the finest level, especially during events causing high optical flow, e.g., rapid motion or frame drops. Using more image pyramid levels makes the system invariant to different image resolutions and scene object scales. Given an image pyramid with n levels, we not only initialize feature patches for FAST corners at level 0 (highest resolution), but up to level $n - 2$. In subsequent images, all patches are then tracked from the coarsest to their initialization level. The detection of corner features at different levels results in a much higher usable feature count. Features found in coarser levels are more distinctive in the coarsest levels, which helps when large optical flow occurs. To account for the different tracking resolutions, we weight the reprojection errors according to the feature initialization level c with

$$w = \frac{1}{2^c}. \tag{7}$$

We evaluate this modification by counting the number of restarts after tracking failures for the different MADMAX [1] sequences, presented in Table I. The numbers in the first row ("Ours") are usually smaller or equal to those in the second ("Ours †"), which indicates that initializing features on all

resolution levels leads to a reduction of tracking failures. Note that the numbers for sequences B-2 and C-2 are not meaningful as tracking failures occur only at the end of the sequence where people walk around, covering a large part of the camera's field of view.

B. Pure Rotation Tracking

We explicitly allow landmarks to have an inverse distance d of 0 ("at infinity"). Such points contribute only to the estimation of rotation but not of translation. Contrary to Basalt, we add a small user-defined prior tying the relative translation between keyframes connected by landmarks with $d = 0$ to **0**, avoiding an undetermined system. A landmark becomes a "point at infinity" in two ways: The first possibility is that during initial triangulation no baseline between two frames which observe the landmark is larger than a threshold; the landmark is then inserted into the map with $d = 0$. The second possibility can occur during bundle adjustment. Optimizing the inverse distance d of a landmark is essentially a constrained optimization problem with $d \geq 0$. During unconstrained Gauss-Newton (GN) optimization, it can happen that $d < 0$. This is solved by a projection into the valid region $d \leftarrow \max(0, d)$ after each GN update step, resulting in $d = 0$.

Allowing landmarks with $d = 0$ enables us to implicitly track rotation-only camera motion without the need for extra panorama maps as in [14]. This feature is especially important for MAVs that want to take panorama overviews from a high altitude. Similarly, we can track pure rotations before a map is initialized. However, if a monocular odometry looses all points with $d > 0$, the following relative pose map initialization (see Sec. IV-C) determines a new scene scale. The resulting two sub-maps can be scale-aligned by a higher level system, as for example [15]. However, this is out of the scope of this paper. A scale-alignment using ground-truth data is visualized in Fig. 3. Every time a new observation of a point with $d = 0$ was made, we perform a new triangulation attempt.

C. Map Initialization

In case there is no landmark with $d > 0$, the relative translation between two frames cannot be measured. This happens, when the map does not contain any landmarks (after system startup or reset due to tracking failure) or when the map only holds landmarks with $d = 0$ (after pure rotational motion around the camera center). Therefore, whenever less than five observations of landmarks with $d > 0$ are
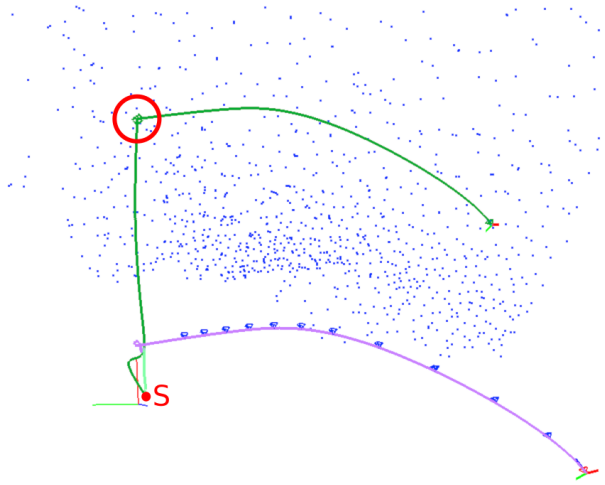
(a) Two sub-maps (turquoise and purple) with different scales separated due to large rotation-only motion (red circle).
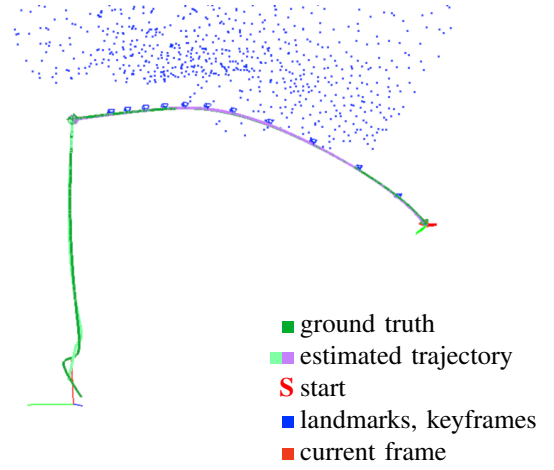
(b) The two sub-maps after separate $\mathrm{Sim}(3)$-alignment to ground truth.

Fig. 3: After rotation-only motion a new sub-map is initialized, leading to a new scene scale. However, the different sections could be linked by a SLAM system. Landmarks, keyframes and current frame are shown only for the last optimization window of the second sub-map.

found in the current image, we attempt a relative pose map initialization. Similarly to [16], for every previous keyframe we generate relative motion hypotheses to the current frame with homography [17] and five-point algorithm [18] in a parallelized RANSAC scheme [19]. Note, that after system startup our keyframe selection strategy (Sec. IV-D) ensures that the first frame is assigned as a keyframe. The best hypothesis in terms of reprojection errors is compared to the rotation-only estimate. If it has more inliers and a smaller sum of reprojection errors we attempt a map initialization: First, we scale the translation such that the average distance of the triangulated points is $\bar{\rho}$, arbitrary but fixed. Then, as proposed in [14], we approximate the parallax α of the scene as

$$\alpha = 2 \arctan\left(\frac{t}{2\bar{\rho}}\right) \qquad (8)$$

where t is the relative distance of the two frames used for initialization. If the parallax is higher than a threshold $\bar{\alpha}$, we consider the initialization attempt successful. Throughout our evaluation we found that $\bar{\alpha} = 5°$ yields robust results. To fix the scene scale we add a new error term E_{scalefix} to the optimization problem as defined in Eq. (4):

$$E_{\text{scalefix}} = w_{\text{scalefix}} \left(|\operatorname{trans}((\boldsymbol{T}_j^{\text{w}})^{-1}\boldsymbol{T}_i^{\text{w}})| - t\right)^2. \qquad (9)$$

Here, w_{scalefix} is a user-defined weight factor. $\boldsymbol{T}_j^{\text{w}}$ and $\boldsymbol{T}_i^{\text{w}}$ are the absolute poses of the frames used for map initialization. This constraint is added to the marginalization prior as soon as either i or j is removed from the current optimization window.

D. Keyframe Selection

The keyframe selection of Basalt is based on heuristics. The user has to define a) a minimum number of frames between two keyframes and b) a threshold percentage for how many features in the current frame should successfully be associated to landmarks. We found that a lot of fine tuning is necessary to adapt those values to new setups and other datasets. In comparison, the approach presented in [11] generalized better to other settings: After a few iterations of non-linear optimization, we evaluate the negative entropy E of the pose of the new frame $\boldsymbol{T}_n$ as

$$E(\boldsymbol{T}_n) = \ln(|\boldsymbol{I}_{\boldsymbol{T}_n}|) \qquad (10)$$

using the Fisher information matrix $\boldsymbol{I}_{\boldsymbol{T}_n} \in \mathbb{R}^{6\times6}$. If $E(\boldsymbol{T}_n)$ drops below a certain threshold percentage of the running average of the negative entropy since the last keyframe insertion, a new keyframe is added. Contrary to [11], we do not make the current but the previous frame a keyframe. Typical cases where significant reduction of negative entropy can be observed are frame drops and rapid motions. Those represent worst case scenarios for optical flow tracking as feature tracks might be torn apart. Therefore, it is crucial to the robustness of a VO that every surviving feature track is harnessed. We achieve this by making the frame before the negative entropy dropped a keyframe and thus, adding – by means of triangulation – new landmarks to the map. The significant improvement due to this design decision can be seen in Table I when comparing the first row ("Ours") with "Ours ‡" (third row). "Ours ‡" uses, as proposed in [11], the frame for which the drop in negative entropy was detected as keyframe, resulting in significantly more tracking failures.

V. ESTIMATING SCALE DRIFT

Similar to the probabilistic keyframe selection criterion of [11], we found that investigating the relative translation information matrix of the current optimization window leads to a good indicator for scale drift. Since visual odometry is essentially a parameter estimation problem, we have
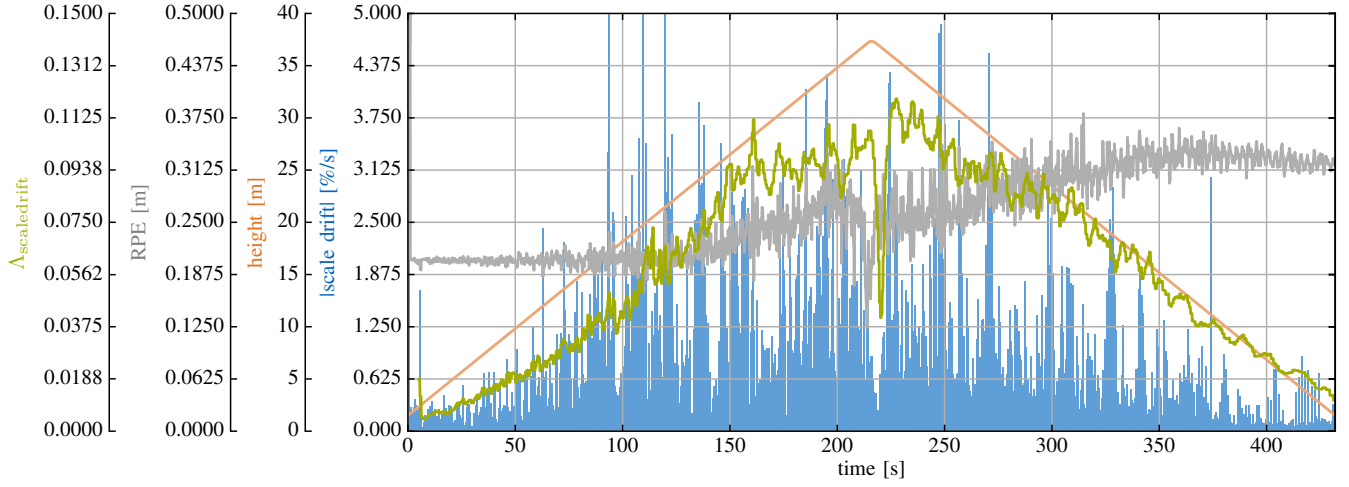
Fig. 4: Evaluation of scale drift and uncertainty estimation of our monocular method on a synthetic sequence. The virtual MAV first ascends and later descends (orange) in a spiral. The Relative Pose Error (RPE) (gray) is evaluated with a $\Delta t = 0.5\,\mathrm{s}$ and by $\mathrm{Sim}(3)$-aligning the first 100 frames. After map initialization (at $\sim 5\,\mathrm{s}$), $\Lambda_{\mathrm{scaledrift}}$ (green) shows to be a good indicator for the actual relative scale drift (blue), computed by comparing the estimated trajectory to the perfect ground truth.
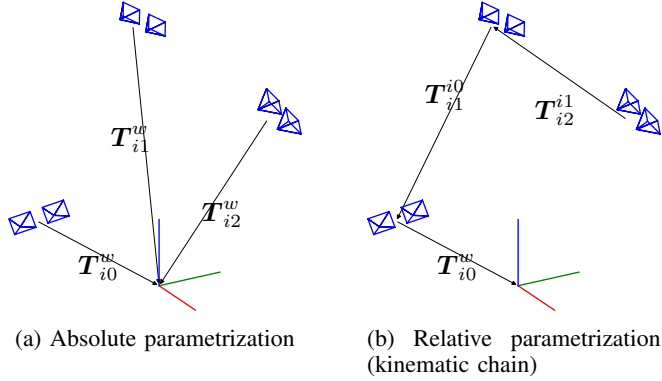

(a) Absolute parametrization

(b) Relative parametrization (kinematic chain)

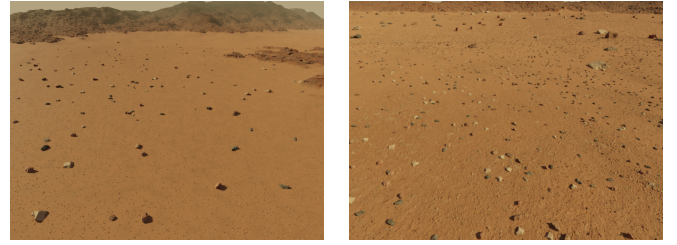Fig. 5: Different ways of keyframe pose parametrization.

Fig. 6: Our photo realistic recreation of the Martian surface using Blender (www.blender.org). It allowed us to synthetically create arbitrary test sequences with perfect ground truth.

measurements z, the true state x and a measurement function $h(\hat{x}) = \hat{z}$ that relates the estimated state $\hat{x}$ to the expected measurement $\hat{z}$. In order to find $\hat{x}$ that is as close to the unknown x as possible, we minimize a non-linear least squares problem with the residuals $r(\hat{x}) = h(\hat{x}) - z$. Assuming that the measurements are disturbed by white noise with covariance Σ_z, the covariance of the current state can be obtained through

$$\Sigma_{\hat{x}} \approx \left(J_h^{\top} \Sigma_z^{-1} J_h \right)^{-1} \tag{11}$$

where J_h is the Jacobian of h evaluated at $\hat{x}$. The inverse of the covariance $\Sigma_{\hat{x}}$ is also known as Fisher information $I_{\hat{x}} = \Sigma_{\hat{x}}^{-1}$. Note that $J_h = J_r$ and therefore the matrix $J_h^{\top} \Sigma_z^{-1} J_h$ is the same as the H matrix computed in a standard Gauss-Newton optimization with update step $\Delta \hat{x}$:

$$\Delta \hat{x} = - \left(J_h^{\top} \Sigma_z^{-1} J_h \right)^{-1} J_h^{\top} r(\hat{x}) \tag{12}$$

$$= -H^{-1} b. \tag{13}$$

To obtain the Fisher information of the relative translations between keyframe states we reparametrize the problem as kinematic chain (Fig. 5). So the state x consists of n relative keyframe poses $\begin{bmatrix} T_0^1 & T_1^2 & ... & T_{n-1}^n \end{bmatrix}$ and m landmark positions $\begin{bmatrix} l_0 & l_1 & ... & l_{m-1} \end{bmatrix}$. The relative pose in Eq. (6) is defined as $T_{h(i)}^t = (T_t^{\mathrm{w}})^{-1} T_{h(i)}^{\mathrm{w}}$. However, for the kinematic chain formulation this becomes $T_{h(i)}^t = T_{b_1}^t T_{b_2}^{b_1} ... T_{h(i)}^{b_n}$ or $T_{h(i)}^t = (T_t^{b_1})^{-1} (T_{b_1}^{b_2})^{-1} ... (T_{b_n}^{h(i)})^{-1}$ (depending on the traversal direction) for all keyframes b_i that are in between the host and target frame. The Jacobian $J_{r_{it}}$ can then be obtained by taking the derivatives of π and $T_{h(i)}^t$ and applying the chain rule. Since we are using the same decoupled left increment as [5], defined as

$$\mathrm{inc}(T, \xi = \begin{bmatrix} v \\ w \end{bmatrix}) = \begin{bmatrix} \exp([w]_{\times}) R & t + v \\ 0_{1,3} & 1 \end{bmatrix} \tag{14}$$

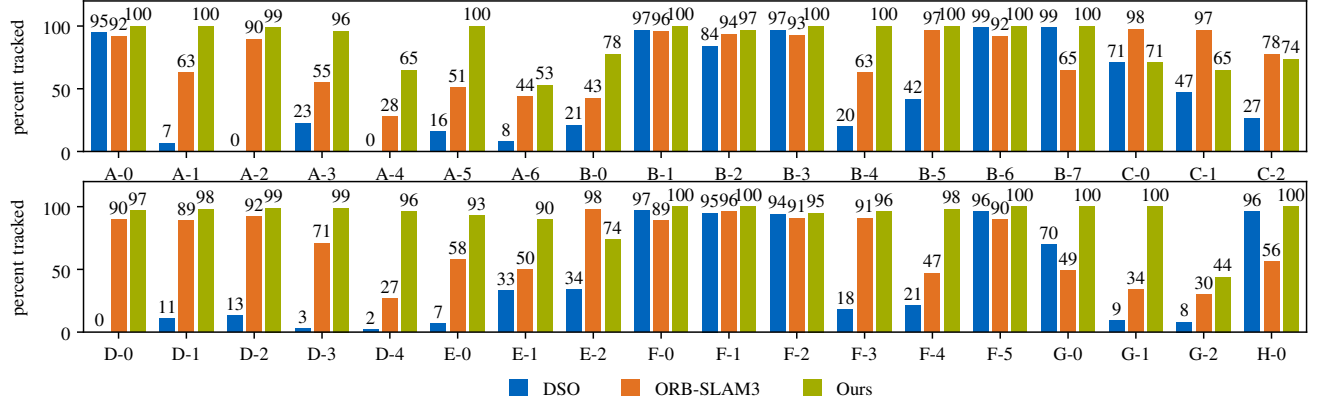the partial derivatives of $T_{h(i)}^t$ w.r.t. the relative pose incre-

Fig. 7: Maximum relative sequence length that could be tracked on the MADMAX dataset [1]. The best value out of three runs is taken. Note that, in contrast to DSO, ORB-SLAM3 and our approach can restart after a failure. For those the longest continuously tracked sequence is taken. Numbers are rounded to nearest integer.

ments $\boldsymbol{\xi}$ are, depending on the direction,

$$\frac{\delta \boldsymbol{T}_{h(i)}^t}{\delta \boldsymbol{\xi}(\boldsymbol{T}_{b_{k+1}}^{b_k})} = \mathrm{Adj}_{\boldsymbol{T}_t^{b_k}} \begin{bmatrix} \boldsymbol{I}_3 & [\mathrm{trans}(\boldsymbol{T}_{b_{k+1}}^{b_k})]_\times \\ \boldsymbol{0}_{3,3} & \boldsymbol{I}_3 \end{bmatrix} \quad (15)$$

$$\frac{\delta \boldsymbol{T}_{h(i)}^t}{\delta \boldsymbol{\xi}(\boldsymbol{T}_{b_k}^{b_{k+1}})} = \mathrm{Adj}_{(\boldsymbol{T}_{b_{k-1}}^t)^{-1}}$$

$$\cdot \begin{bmatrix} -\mathrm{rot}(\boldsymbol{T}_{b_k}^{b_{k+1}})^\top & \boldsymbol{0}_{3,3} \\ \boldsymbol{0}_{3,3} & -\mathrm{rot}(\boldsymbol{T}_{b_k}^{b_{k+1}})^\top \end{bmatrix}. \quad (16)$$

These Jacobians give us the Fisher information of a distribution of the entire state $p(\hat{\boldsymbol{x}}|\boldsymbol{z}_0...\boldsymbol{z}_k)$. In a first step we marginalize out all landmark variables using the Schur complement [20]. This leaves us with a distribution of the relative poses $p(\boldsymbol{T}_0^1, \boldsymbol{T}_1^2, ..., \boldsymbol{T}_{n-1}^n | \boldsymbol{z}_0...\boldsymbol{z}_k)$. After transforming and adding the marginalization prior, we marginalize out all rotation variables, again, using the Schur complement. This gives us a distribution with only the translation variables $\boldsymbol{t}$ left: $p(\mathrm{trans}(\boldsymbol{T}_0^1), \mathrm{trans}(\boldsymbol{T}_1^2), ..., \mathrm{trans}(\boldsymbol{T}_{n-1}^n)|\boldsymbol{z}_0...\boldsymbol{z}_k)$. We call the resulting matrix $\boldsymbol{H_t}$ the relative translation information matrix. Let $\boldsymbol{H_t} = \boldsymbol{Q}\,\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{Q}^{-1}$ be the eigenvalue decomposition of $\boldsymbol{H_t}$. Inspired by Principal Component Analysis, we propose to normalize the inverse square root of the smallest eigenvalue by the average relative translation of the current optimization window:

$$\Lambda_{\mathrm{scaledrift}} = \left(\sqrt{\min(\boldsymbol{\lambda})} \frac{1}{n} \sum_{k=0}^{n-1} |\mathrm{trans}(\boldsymbol{T}_k^{k+1})| \right)^{-1}. \quad (17)$$

This gives us a metric for estimating the expected relative scale drift, which can be used in decision making. A path planner, for example, could decide to fly lower, return to the home point or actively search for an object of known size.

To assess the validity of Eq. (17) perfect ground truth is needed. Thus, we use a photo-realistic simulation (Fig. 6). In this setup the virtual MAV flies up- and downwards in a spiral with constant speed. Fig. 4 shows the MAV height, the resulting scale drift as well as $\Lambda_{\mathrm{scaledrift}}$. To show the effects of accumulated scale drift we include the Relative

Pose Error (RPE) [21]. As can be seen, $\Lambda_{\mathrm{scaledrift}}$ is a good indicator for the actual scale drift.

VI. COMPARISON WITH STATE OF THE ART

We evaluate our method on the MADMAX dataset [1] as it is recorded in an environment similar to the Martian surface. Due to its repetitive textures, unstructured terrain and frame drops it is a challenging benchmark dataset relevant to planetary robotic missions. In Fig. 7, we show the percentage of the overall sequence that was tracked without failing (for comparison with results of VINS-Mono and ORB-SLAM2 using both visual and inertial sensor data see [1]). Note that ORB-SLAM3 and our method can restart after failure and therefore return the longest continuously tracked section whereas DSO does not have such a feature. While ORB-SLAM3 fails at random points in time, our method only looses track when frame drops and fast motion coincide. In Table II, the Root Mean Squared Relative Pose Error (RMS RPE) is shown, indicating that our method is also more precise. Given the ground truth pose $\boldsymbol{Q}_t \in SE(3)$ and the estimated pose $\boldsymbol{T}_t \in SE(3)$ at time t we define the RMS RPE for estimated trajectories with unknown scale analogously to [21] as

$$\delta \boldsymbol{Q}_t = \boldsymbol{Q}_{t-\Delta}^{-1} \boldsymbol{Q}_t \qquad \delta \boldsymbol{T}_t = \boldsymbol{T}_{t-\Delta}^{-1} \boldsymbol{T}_t \quad (18)$$

$$s_t = \frac{|\mathrm{trans}(\delta \boldsymbol{Q}_t)|}{|\mathrm{trans}(\delta \boldsymbol{T}_t)|} \quad (19)$$

$$\mathrm{RMS\ RPE} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} |s_t \mathrm{trans}(\delta \boldsymbol{T}_t) - \mathrm{trans}(\delta \boldsymbol{Q}_t)|^2}. \quad (20)$$

Here, $\mathcal{T}$ is the set of timestamps for which a pose is estimated without those from the first Δ seconds. To get the poses $\boldsymbol{Q}_t$, $\boldsymbol{Q}_{t-\Delta}$ and $\boldsymbol{T}_{t-\Delta}$ we use linear interpolation on the trajectories. By calculating the scale factor s_t we eliminate the scale and only measure the error of the estimated relative direction. Since the ground truth has a temporal resolution of $1\,\mathrm{s}$, we computed the values shown in Table II with

TABLE II: RMS RPE of the longest run on the MADMAX dataset [1]. Calculated with $\Delta = 4\,\mathrm{s}$. Below each entry, the maximum relative sequence length ("tracking percentage", see Fig. 7.) is given in parentheses. For runs with tracking percentage $> 50\%$, the best RMS RPE result is highlighted.

	A-0	A-1	A-2	A-3	A-4	A-5	A-6	B-0	B-1	B-2	B-3	B-4	B-5	B-6	B-7	C-0	C-1	C-2
DSO [10]	0.461	0.446	-	1.484	0.353	0.570	1.063	0.287	0.473	0.545	0.307	0.690	0.412	0.807	0.642	0.260	0.359	0.423
	(95%)	(7%)	(0%)	(23%)	(0%)	(16%)	(8%)	(21%)	(97%)	(84%)	(97%)	(20%)	(42%)	(99%)	(99%)	(71%)	(47%)	(27%)
ORB-SLAM3 [8]	0.306	**0.668**	0.273	0.980	1.368	1.190	1.290	0.447	0.252	0.200	**0.222**	**0.444**	0.798	0.788	0.661	0.341	1.450	**0.551**
	(92%)	(63%)	(90%)	(55%)	(28%)	(51%)	(44%)	(43%)	(96%)	(94%)	(93%)	(63%)	(97%)	(92%)	(65%)	(98%)	(97%)	(78%)
Ours	**0.278**	0.673	**0.272**	**0.813**	**0.949**	**0.831**	**1.181**	**0.659**	**0.234**	**0.194**	0.410	1.936	**0.685**	**0.635**	**0.633**	**0.258**	**0.218**	0.611
	(100%)	(100%)	(99%)	(96%)	(100%)	(65%)	(53%)	(100%)	(97%)	(100%)	(100%)	(100%)	(100%)	(100%)	(100%)	(100%)	(65%)	(74%)
	D-0	D-1	D-2	D-3	D-4	E-0	E-1	E-2	F-0	F-1	F-2	F-3	F-4	F-5	G-0	G-1	G-2	H-0
DSO [10]	0.368	0.395	0.166	0.128	0.226	0.493	0.441	0.261	0.180	0.193	0.178	0.222	0.178	0.518	0.547	0.182	0.203	0.817
	(0%)	(11%)	(13%)	(3%)	(34%)	(7%)	(33%)	(34%)	(95%)	(94%)	(18%)	(21%)	(96%)	(70%)	(9%)	(8%)	(9%)	(96%)
ORB-SLAM3 [8]	0.133	0.140	0.140	**0.202**	0.270	0.475	1.991	0.976	0.167	0.181	**0.165**	0.705	1.690	0.517	0.334	0.276	1.380	0.348
	(90%)	(89%)	(92%)	(71%)	(27%)	(58%)	(50%)	(98%)	(89%)	(96%)	(91%)	(91%)	(47%)	(90%)	(49%)	(34%)	(30%)	(56%)
Ours	**0.105**	**0.124**	**0.131**	0.216	0.924	0.376	1.783	0.638	0.158	0.180	0.173	0.702	1.661	0.494	0.535	0.413	0.313	**0.291**
	(97%)	(98%)	(99%)	(99%)	(96%)	(93%)	(90%)	(74%)	(100%)	(100%)	(95%)	(96%)	(98%)	(100%)	(100%)	(100%)	(44%)	(100%)

$\Delta = 4\,\mathrm{s}$. Not visible in the RMS RPE is that ORB-SLAM3, by performing global bundle adjustment, shows less scale drift. Of course, our approach could also benefit from global optimization of all keyframes.

VII. CONCLUSIONS

We presented a monocular odometry that is especially robust in environments relevant for planetary exploration. We achieved this by using KLT tracking with features detected at different image pyramid levels and by refining the keyframe selection strategy of [11]. We detailed our map initialization strategy since for monocular setups, special care has to be taken at the start of a new trajectory. By allowing landmarks with inverse distance $d = 0$, our system can handle rotation-only motion sequences. Compared to other methods our system shows significantly less tracking failures and high local accuracy on the MADMAX dataset [1]. In Eq. (17), we proposed a quantity that allows for the detection of scale drift which is crucial to overall system robustness. It is straight forward to apply a similar criterion to detect when a stereo setup degrades to a monocular one. This happens during flights at high altitude, as the stereo baseline cannot constrain the scene scale well anymore. We will further explore the practical application of our findings during the ARCHES mission [2].

ACKNOWLEDGMENT

REFERENCES

[1] L. Meyer, M. Smíšek, *et al.*, "The MADMAX dataset for visual-inertial rover navigation on Mars," *Journal of Field Robotics*, In press.

[2] M. J. Schuster, M. G. Müller, *et al.*, "The ARCHES space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 4, pp. 5315–5322, 2020.

[3] D. S. Bayard, D. T. Conway, *et al.*, *Vision-Based Navigation for the NASA Mars Helicopter*. American Institute of Aeronautics and Astronautics, 2019.

[4] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 2, 1981, p. 674–679.

[5] V. Usenko, N. Demmel, *et al.*, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robotics and Automation Letters (RA-L) and International Conference on Intelligent Robotics and Automation (ICRA)*, vol. 5, no. 2, pp. 422–429, 2020.

[6] J. Delaune, D. S. Bayard, and R. Brockers, "Range-visual-inertial odometry: Scale observability without excitation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 2421–2428, 2021.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007.

[8] C. Campos, R. Elvira, *et al.*, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *arXiv abs/2007.11898*, 2020.

[9] 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.

[10] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[11] J. Kuo, M. Muglikar, *et al.*, "Redesigning SLAM for arbitrary multi-camera systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2116–2122.

[12] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[13] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," *Journal of Field Robotics*, vol. 27, no. 5, pp. 587–608, 2010.

[14] C. Pirchheim, D. Schmalstieg, and G. Reitmayr, "Handling pure camera rotation in keyframe-based slam," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013, pp. 229–238.

[15] R. Elvira, J. D. Tardós, and J. M. M. Montiel, "ORBSLAM-Atlas: a robust and accurate multi-map system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6253–6259.

[16] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[17] O. Faugeras and F. Lustman, "Motion and structure from motion in a piecewise planar environment," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 3, 1988.

[18] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[19] 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, p. 381–395, 1981.

[20] M. I. A. Lourakis and A. A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Transactions on Mathematical Software*, vol. 36, no. 1, 2009.

[21] J. Sturm, N. Engelhard, *et al.*, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE/RSJ International Conference on Intelligent Robot Systems (IROS)*, 2012, pp. 573–580.