Instant Visual Odometry Initialization for Mobile AR

Alejo Concha, Michael Burri, Jesus Briales, Christian Forster, and Luc Oth

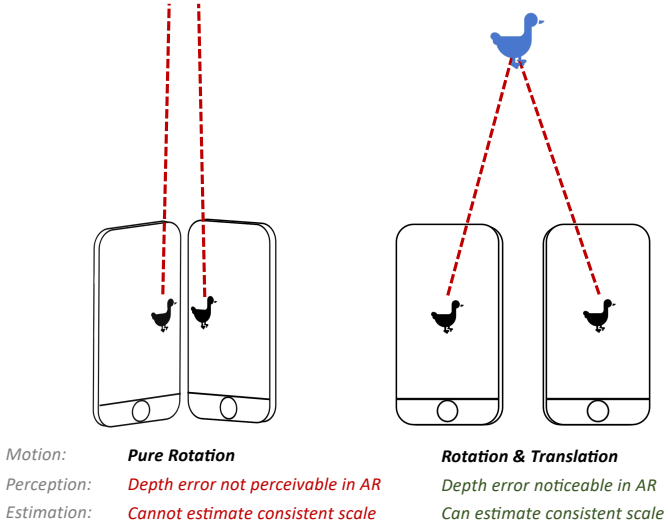

Fig. 1. The left part of the Figure shows that depth is neither perceivable nor recoverable during pure rotational motion. On the other hand, if the device translates as shown on the right, it is possible to estimate consistent depth in a monocular setting up to a constant scale factor. However, an in-consistent scale over multiple frames would be user noticeable. Our initialization method naturally moves from the left to the right by starting to estimate translation magnitude corresponding to a consistent scale as depth errors become perceivable.

Abstract—Mobile AR applications benefit from instant initialization to display world-locked effects promptly. However, standard visual odometry or SLAM algorithms require motion parallax to initialize (see Figure 1) and, therefore, suffer from delayed initialization. In this paper, we present a 6-DoF monocular visual odometry that initializes instantly and without motion parallax. Our main contribution is a pose estimator that decouples estimating the 5-DoF relative rotation and translation direction from the 1-DoF translation magnitude. While scale is not observable in a monocular vision-only setting, it is still paramount to estimate a *consistent* scale over the whole trajectory (even if not physically accurate) to avoid AR effects moving erroneously along depth. In our approach, we leverage the fact that depth errors are not perceivable to the user during rotation-only motion. However, as the user starts translating the device, depth becomes perceivable and so does the capability to estimate consistent scale. Our proposed algorithm naturally transitions between these two modes. Our second contribution is a novel residual in the relative pose problem to further improve the results. The residual combines the Jacobians of the functional and the functional itself and is minimized using a Levenberg–Marquardt optimizer on the 5-DoF manifold. We perform extensive validations of our contributions with both a publicly available dataset and synthetic data. We show that the proposed pose estimator outperforms the classical approaches for 6-DoF pose estimation used in the literature in low-parallax configurations. Likewise, we show our relative pose estimator outperforms state-of-the-art approaches in an odometry pipeline configuration where we can leverage initial guesses. We release a dataset for the relative pose problem using real data to facilitate the comparison with future solutions for the relative pose problem. The proposed odometry is currently used as a pre-SLAM initialization module in world-locked AR effects in Instagram and Facebook.

Index Terms—Monocular initialization, relative pose estimator, Visual Odometry, AR instant placement.

1 Introduction

Knowing the precise 6-DoF motion of a mobile phone allows us to augment the real-world with virtual effects. A popular use-case of this capability is the placement of virtual furniture to make purchasing decisions. The pose of the mobile phone can be estimated with camera-based Simultaneous Localization and Mapping (SLAM) or Visual Odometry (VO). These methods detect and track salient features in the environment and thereby localize the camera. While SLAM builds and optimizes a map of the observed scene, a VO system just maintains a sliding-window of the most recent estimated camera poses and landmarks at the cost of lower global accuracy but higher compute efficiency. Since many world-locked AR experiences are very short in nature, and compute usage is of high importance, a VO system suffices for most applications. Cadena et al. [5] is a complete summary of the present and future of SLAM / VO.

Widely used mobile SDKs such as ARKit or ARCore leverage the inertial measurement unit (IMU) in addition to the camera sensors. The IMU is an ideal complementary sensor to the camera as it (1) renders gravity and scale observable and (2) provides reliable angular velocity and linear acceleration even when the cameras are occluded or the images suffer motion-blur or low-contrast. There exists a large body of devices without accurate and stable spatial-temporal camera-IMU

All authors are with Facebook Zurich, Switzerland. E-mail: aconchabelenguer@gmail.com alejocb@fb.com

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

calibration. Additionally, there exist many low-end smartphones with accelerometer but no gyroscope or phones with faulty sensor data due to manufacturing imprecisions. Ubiquitous mobile AR needs to support low-end devices, which motivated the development of our vision-only VO system. Due to the major advantage of leveraging the IMU, the system is able to optionally include and process inertial measurements in case they are available.

Many world-locked mobile AR experiences are very short in nature (a few seconds) as users rapidly browse through different AR effects that rely on different underlying capabilities such as world-locked, face-locked, and object-locked. One way to increase user satisfaction is to initialize the tracker instantaneously to minimize time-to-fun. However, standard VO and visual-inertial odometry initialization techniques require translational motion to initialize (approx. 10 cm, depending on the scene depth). This is because many closed-form solutions that are used to jointly initialize the initial camera poses and 3D landmark require translation to triangulate the 3D points or may even be degenerate in the rotation-only case. This motivated our work on a VO system that initializes instantaneously and independently of the user motion. The proposed VO is used for instant initialization of world-locked AR effects from smartphone apps used worldwide– i.e., as a pre-SLAM component of any supported SLAM system (in-house, ARKit, ARCore).

While our contribution has been motivated by the mobile AR use-case, there are several other applications in the robotics community that rely on instant initialization of VO or SLAM. An example is the rapid deployment of vision-based micro aerial vehicles as shown in Faessler et al. [11].

2 Related work

Initializing stereo (Mur-Artal et al. [24]) or RGB-D systems (Newcombe et al. [25], Concha and Civera [7]) is straightforward in the vast majority of cases. These systems can estimate the depth of features without moving the device. There are some works for fast initialization using IMU sensors like Bloesch et al. [3] or Li et al. [20]. In the context of VO initialization, Huang et al. [17] proposes a method for initializing from first frame using vanishing points and some indoor cues. Moreover, single-view depth estimation has the potential to initialize a map from first frame (Fácil et al. [10]).

In this paper, and contrary to some of the above-mentioned works, we do not make any assumptions regarding the structure of the environment and we do not use additional sensors to solve this problem either. In this context, VO initialization can be divided in two main research lines:feature-based (indirect) initialization methods and direct initialization methods. Different initialization techniques have been explored in both research lines:

2.1 Feature-based (indirect) initialization methods

Feature-based initialization approaches extract features in an image that are matched in another image of the same scene with a different viewpoint. Feature matches are used to estimate both an initial relative pose between the views and a map using either closed-form solvers or optimization-based techniques:

2.1.1 Minimal solvers based on essential and homography matrices

The relative pose can be recovered by estimating an essential matrix (eight-point algorithm, Zisserman and Hartley [2]) or a homography matrix (Faugeras and Lustman [13], Longet [21]) between the cameras, depending on the camera motion and scene structure. Camera poses are then used to triangulate the feature correspondences and build a map that is used for tracking in the subsequent frames. Even though these solvers have been used successfully for decades, they still present some issues that motivated this paper:

- Non-instantaneous initialization. The Essential matrix might not be correctly estimated in the case of zero translation. A zero matrix becomes a valid solution in such situations and therefore its constraints deteriorate.
- Non-minimal parametrization: The parameters from the essential and the homography matrix are not directly related to the actual motion-related parameters and the parametrization is not minimal, which makes it harder to withdraw conclusions from the actual estimated values during the optimization.
- Structure-dependency. Essential matrix estimation is an ill-defined problem with planar scenes. A homography model (Faugeras et al. [13]) is proposed in the literature for these situations as backup plan (Mur-Artal et al. [23]). A homography model cannot be applied to non-planar scenes, unless points are far away where translation cannot be recovered. Mur-Artal et al. [23] deals with this problem by using a model selection strategy: An homography and an essential matrix are estimated in parallel and the one that produces the lowest re-projection error is taken as a candidate for initialization.
- Solution multiplicity. Homography solvers return multiple solutions (Faugeras et al. [13]) that need to be disambiguated. Some of the solutions can be trivially rejected but for others there is not a better option than triangulate and track feature correspondences for a few frames with a SLAM system to then check their re-projection error in subsequent frames Mur-Artal et al. [23].

2.1.2 Optimization-based initialization methods

Several optimization-based approaches have been explored that address some of the limitations of minimal solvers that estimate an Essential or Homography matrix. Kneip and Lynen [18] have addressed this problem proposing a direct optimization of frame to frame rotation. Lee and Civera [19] have recently extended this work to multiple views, proposing a rotation-only bundle adjustment optimizer. On the downside, these approaches need to cope with non-trivial and non-convex optimization problems for which finding the right optimal solution vs other minima can turn challenging. Briales et al. [4] proposed for the first time a solver for (an equivalent form of) the problem formulation by Kneip and Lynen [18] which comes with global optimality guarantees. The approach however relies on solving a convex SDP relaxation of an equivalent Quadratically Constrained Quadratic Program (QCQP) formulation of the problem (Park and Boyd [26]), which is computationally intensive (with respect to real-time standards) and makes it non-straightforward to leverage initial guesses for the relative pose. García-Salguero et al. [16] extends this work and proposes an optimality certifier for the relative pose problem.

The main advantage of relative pose estimators is their ability to accurately estimate rotations independently of the scene structure and motion of the camera. Translations are also independent of the scene structure and are accurately estimated if there is enough baseline between the cameras. For these reasons, we have used a relative pose estimator in our formulation, see section 3 for more details.

2.2 Direct initialization methods

Direct approaches directly use pixel intensity for tracking and mapping. For initialization, direct approaches either use the above-mentioned solvers from indirect approaches to initialize their systems Engel et al. [9] or create an inaccurate first map. For example, in Engel et al. [8], a random map is initialized and in Concha and Civera [6] the features are placed at a constant distance from the first camera. In these approaches, the initial inaccurate map is continuously refined as the camera moves hoping for an eventual convergence of the map parameters to their true values. Generating an initial inaccurate map can work in some situations but can fail catastrophically if the initial map is very different from the actual one. This is because the map is directly used to estimate all 6-DoF parameters, meaning errors in the map are directly propagated to all estimation parameters and the estimation can eventually diverge.

3 Contributions and Outline

Optimization-based relative pose estimators solve most of the limitations (section 2.1.1) that are present in classical minimal solvers that

estimate the essential matrix between two views. Even though relative pose solvers are good at estimating the two-view problem, they fail to estimate a *consistent* scale in a VO setting as they return a unit-norm translation. While the *true* scale is not observable in a monocular vision-only setting, it is still paramount to maintain a consistent scale by estimating the translation magnitude in the two-view problem. This is required to recover a camera trajectory that is accurate up to a single scale factor that remains constant throughout the session. To solve this issue:

- We propose to combine a relative pose estimator with a translation-magnitude-estimator. The translation-magnitude-estimator minimizes keyframe to frame re-projection errors using depth-estimates of past feature observations as input. With this formulation we use all feature correspondences to estimate the relative pose while only using the correspondences with estimated depth to estimate the magnitude of the translation. In traditional SLAM systems, only those features with an accurate estimated depth can be used to estimate the full 6-DoF pose of the camera. This is why direct approaches that initialize an inaccurate map can fail during initialization when the map is inaccurate (see section 2.2), the error in the map is directly propagated to all motion parameters which can eventually diverge. We also initialize an initial inaccurate map, however, in our case the errors in the map do not affect the estimated relative pose but only the estimated translation magnitude. We exploit the fact that while translation is not observable, the accuracy of the map does not matter. The reverse also holds: at the point where translation becomes significant, feature depth becomes observable. We show in the experimental section 7 that our 5-DoF + 1-DoF optimizer outperforms the classical 6-DoF pose estimator in low-parallax motions.
- One challenge with solving the optimization problem underlying relative pose estimators is that the underlying formulation is not a sum of squares, but rather a sum of algebraic errors. As a result we cannot directly apply classical simple and lightweight Gauss-Newton-like algorithms, but need to resort to more generic Newton-like methods like Riemannian Trust Regions – Absil et al. [1].

  These more generic solvers are not as generally available as classical Least Squares solvers like Gauss-Newton or Levenberg–Marquardt (More [22]), and often lack the maturity of the latter (specially when it comes to availability as efficient C++ implementations within common libraries). Besides, in resource-constrained platforms like mobile devices, where each new library dependency matters, it is desirable that we rely on a classical (already available) Least Squares solver.

  As an alternative to the above-mentioned limitations, Kneip and Lynen [18] proposes to minimize a different functional consisting of the sum of squared Jacobians of the original functional and therefore a classical Gauss-Newton or Levenverg-Marquardt algorithm can be applied. This approach, by definition, is equivalent to finding a local minimum for the original functional (where Jacobians become null). The surrogate residual solved in Kneip and Lynen [18] has multiple equally valid global minimum (as many as local minimum exist in the original functional), making it much harder to converge to the globally optimal solution.

  As a second contribution, we extend the Jacobian minimization trick of Kneip and Lynen [18] and include the objective function itself in the residual. This basically makes the surrogate residual well-defined as there will be a single global optimum, making convergence to the right solution much easier. Another characteristic of this approach is that it allows the usage of initial guesses for the parameters to estimate, which is a big advantage for visual odometries.

The rest of the paper is structured as follows: the problem is defined in Section 4. Sections 5 and 6 explain the proposed approach in detail. We validate our contributions in Section 7. Finally, conclusions and future lines of work are given in Section 8. Additionally, we include an Appendix in Section 9 for the derivations omitted from Section 5.

4 Problem definition

Estimate in real-time and without initialization a consistent 6-DoF trajectory of the motion of a monocular device with calibrated camera parameters. We can solve this by estimating $\boldsymbol{T} \in \mathtt{SE(3)}$, the pose transform between a previous frame (keyframe) and the current frame given a set of feature correspondences between them ($\boldsymbol{f_i}$ in the past keyframe and $\boldsymbol{f_i'}$ in the current frame) and – optionally – the depth of the features in the source view (d_i). Where i the index of the feature. Note if $\boldsymbol{p_i}$ is a 3D point in the reference frame of the first view and $\boldsymbol{p_i'}$ is the same point in the reference frame of the second view, they are related through $\boldsymbol{T}$ as $\boldsymbol{p_i'} = \boldsymbol{T}\boldsymbol{p_i}$.
Where $\boldsymbol{T}$ is defined as follows:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{u}s \\ 0_{1\times 3} & 1 \end{bmatrix}, \text{s.t. } \boldsymbol{R} \in \mathrm{SO}(3), \boldsymbol{u} \in S^2, s \in \mathbb{R} \tag{1}$$

Where SO(3) is the rotation group (Stuelpnagel [27]) and S^n represents points in the n-dimensional unit sphere. $\boldsymbol{u}$ is the translation direction, a 3D unit-norm vector.

Through the paper, we refer to the magnitude of the translation as $s \in \mathbb{R}$. Whenever we refer to 5-DoF in this paper we mean the relative pose between two views $(\boldsymbol{R}, \boldsymbol{u})$, while if we refer to 6-DoF we mean we additionally estimate the magnitude of the translation to produce a full 6-DoF pose $(\boldsymbol{R}, \boldsymbol{u}, s) \rightarrow (\boldsymbol{R}, \boldsymbol{t})$ with consistent scale along the trajectory (but *not* metric scale, as this is a monocular odometry).

5 6-DOF Pose estimation

Our pipeline is composed of two main components, a frame to frame tracker and a pose estimator (see Figure 2). In this section we explain the proposed 6-DoF estimator, which is the main contribution of this paper. We will explain the implementation details of the frame-to-frame tracker in the next section 6 for completeness.

Following the 6-DoF parametrization in eq. 1, we can split the 6-DoF estimation problem into (1) the estimation of the relative pose ($\boldsymbol{R} \in \mathrm{SO}(3)$ with unit-norm translation $\boldsymbol{u} \in S^2$) and (2) the estimation of the magnitude of the translation (s). The relative pose can be estimated using all feature correspondences between two views using a relative pose estimator (see section 5.1 for details). Once the relative pose is estimated, we estimate the magnitude of the translation. To estimate the magnitude of the translation, we also need as input the depth of the *optionally* triangulated features, which might have been estimated using past 6-DoF poses via triangulation (see Section 5.2 for details).

5.1 5-DoF Relative Pose Estimator

The relative pose problem is the problem of determining the relative rotation and direction of the translation between two frames using 2D-2D correspondences.

Given the bearing vectors ($\boldsymbol{f_i}$ and $\boldsymbol{f_i'}$) from a set of correspondences, Kneip and Lynen [18] proposed to solve the relative pose problem by enforcing the co-planarity of the epipolar plane normals $\boldsymbol{m_i} = \boldsymbol{R}\boldsymbol{f_i} \times \boldsymbol{f_i'}$ for all epipolar planes (see Figure 3 for a graphical explanation). This is achieved by minimizing the minimum eigenvalue of the covariance matrix of the epipolar plane normals $\boldsymbol{M}(\boldsymbol{R}) = \sum_{i=1}^{n} \boldsymbol{m_i}\boldsymbol{m_i}^\intercal$, where the rotation matrix $\boldsymbol{R}$ is the variable to optimize:

$$\hat{\boldsymbol{R}} = \arg\min_{\boldsymbol{R}} \lambda_{min}\left(\boldsymbol{M}(\boldsymbol{R})\right) \tag{2}$$

We refer the reader to the original work Kneip and Lynen [18] for more details on the derivation of the functional. Once $\boldsymbol{R}$ is estimated, $\boldsymbol{u}$ can be retrieved as the eigenvector corresponding to the minimum eigenvalue of $\boldsymbol{M}\left(\hat{\boldsymbol{R}}\right)$:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \boldsymbol{u}\boldsymbol{M}\left(\hat{\boldsymbol{R}}\right)\boldsymbol{u}^\intercal \tag{3}$$

Instead, we follow a similar approach proposed by Briales et al. [4], solving translation and rotation simultaneously. The equation

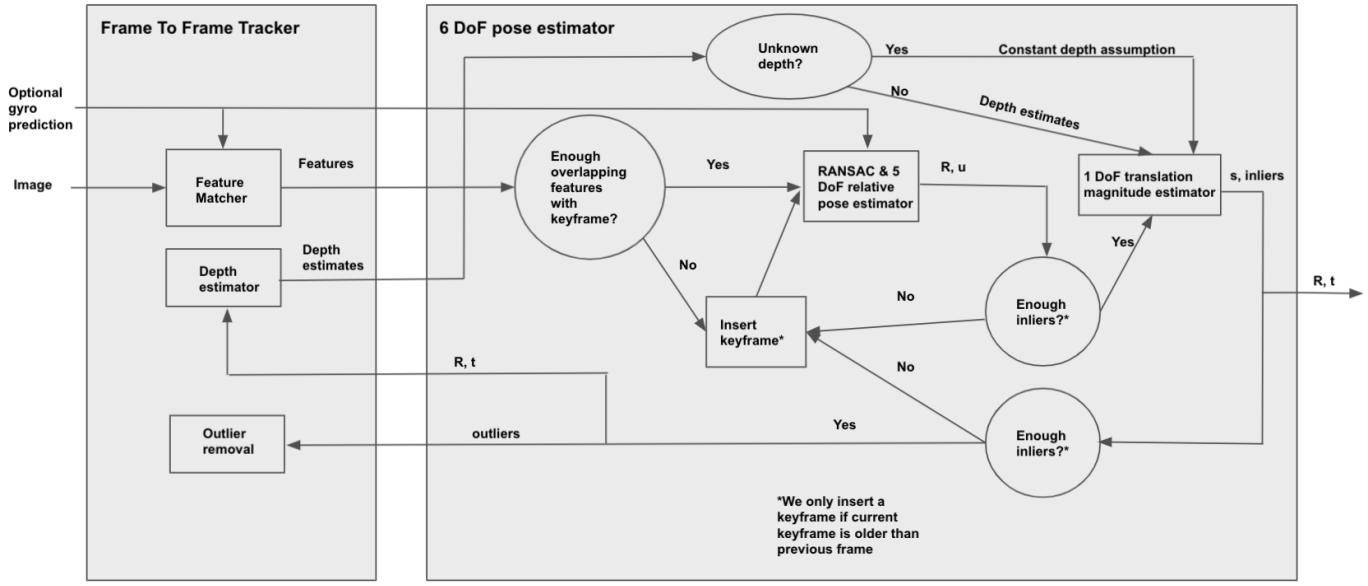

Fig. 2. Pipeline of our approach. Images (and optionally IMU gyro predictions) feed our visual odometry. The frame-to-frame tracker matches features and produces feature correspondences (bearing vectors) that are consumed by the 6-DoF pose estimator. The pose estimator contains a relative pose estimator (5-DoF) that takes the bearing correspondences and estimates a 5-DoF pose. The 5-DoF pose and the estimated depth of the features (depth is estimated by the tracker) are used to estimate the magnitude of the translation, completing the final 6-DoF pose. If depth is not available (during the first frames), we use the constant depth assumption. A new keyframe is inserted in the previous frame if there are not enough overlapping features between current keyframe and last frame or if any of the estimators fail. If current keyframe is already the previous frame, we do not insert a new keyframe. Outliers from pose estimators are sent to the tracker for removal.

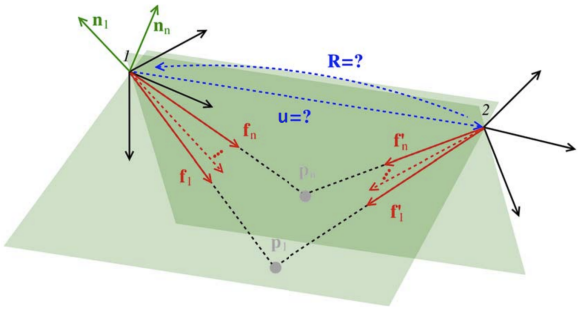

Fig. 3. Graphical definition of the relative pose problem. Bearing vectors (in red) from two different views are the known variables. The rotation $\boldsymbol{R}$ and translation direction $\boldsymbol{u}$ between the views are estimated by minimizing the equation 7. Figure borrowed from Kneip and Lynen [18].

.

3 is minimized directly with respect to both rotation and direction translation:

$$\hat{\boldsymbol{R}}, \hat{\boldsymbol{u}} = \underset{\boldsymbol{u},\boldsymbol{R}}{\arg\min}\, \boldsymbol{u}\boldsymbol{M}(\boldsymbol{R},)\,\boldsymbol{u}^{\mathsf{T}} \tag{4}$$

The computation of the Jacobians of this equation is simplified following the derivations available in Briales et al. [4] to obtain the following functional:

$$\hat{\boldsymbol{R}}, \hat{\boldsymbol{u}} = \underset{\boldsymbol{R}\in \mathrm{SO}(3)\boldsymbol{u}\in S^2}{\arg\min}\, \boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^{\mathsf{T}} \tag{5}$$

Where the data matrix $\boldsymbol{C} \in \mathrm{Sym}_{27}$ gathers all data (derived from the original bearing vectors). $\boldsymbol{x} = vec\,(\boldsymbol{r}\boldsymbol{u}^{\mathsf{T}})$ where vec() is a vectorization function and $\boldsymbol{r}$ is the vectorized form of the rotation matrix $\boldsymbol{R}$.

At this stage, we diverge from the QCQP procedure in Briales et al. [4] to solve the optimization problem. While Briales et al. [4] investigated the properties of the optimization (certifying the solution is globally optimal), we are interested in building a real-time VO system where the relative pose estimator is its main building block. For this reason, we choose a Levenberg–Marquardt optimizer, which is a classical optimizer that is better suited for frame-to-frame optimizations since it is efficient and allows us to leverage predictions from the previous frame as initial guesses. The challenge with the equation 5 is that the residual is scalar and therefore the number of unknowns (5) does not match the size of the residual (1). To solve this problem, we applied the trick proposed in Kneip and Lynen [18] where instead of minimizing the functional, we minimize the Jacobians of the functional. That way, the dimension of the residual equals the number of unknowns. For clarity, we split the 5 parameters to optimize in two different variables. The so(3) rotation parameters are stored in variable $\boldsymbol{\theta}$ while the 2 parameters to optimize from the 3D unit-norm manifold are stored in variable $\boldsymbol{\beta}$. Using this convention, the functional reads as follows:

$$\hat{\boldsymbol{R}}, \hat{\boldsymbol{u}} = \underset{\boldsymbol{R}\in \mathrm{SO}(3),\boldsymbol{u}\in S^2}{\arg\min} \sum_{i=1}^{3} \frac{\partial \boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^{\mathsf{T}}}{\partial \theta_i} + \sum_{i=1}^{2} \frac{\partial \boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^{\mathsf{T}}}{\partial \beta_i} \tag{6}$$

The second contribution of this paper is the inclusion of the functional in the residual to optimize to further improve the results. The original residual has convergence issues when the initial guesses of the parameters to estimate are not good because it only minimizes the Jacobians of the functional. Adding the functional to the residual improves the convergence properties of the problem, as shown in the experimental section. The final residual $f(\boldsymbol{R},\boldsymbol{u})$ reads as follows:

$$\hat{\boldsymbol{R}}, \hat{\boldsymbol{u}} = \underset{\boldsymbol{R}\in \mathrm{SO}(3),\boldsymbol{u}\in S^2}{\arg\min}\, f(\boldsymbol{R},\boldsymbol{u}) \tag{7}$$

$$f(\boldsymbol{R},\boldsymbol{u}) = \sum_{i=1}^{3} \frac{\partial \boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^{\mathsf{T}}}{\partial \theta_i} + \sum_{i=1}^{2} \frac{\partial \boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^{\mathsf{T}}}{\partial \beta_i} + W\boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^{\mathsf{T}} \tag{8}$$

Where the constant W is used to tune the properties of the estimation by differently weighting the Jacobians and 1-d residual. How this constant influences the results is evaluated in section 7.3. The functional is minimized using a standard Levenberg–Marquardt optimizer where the initial guess for the rotation can be obtained by gyro propagation if available or from the previous frame otherwise. The initial guess for the translation is obtained using equation 3 given the set of feature correspondences and the initial guess for the rotation.

The jacobians of the functional ($\frac{\partial \boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^\intercal}{\partial \theta_i}$ and $\frac{\partial \boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^\intercal}{\partial \beta_i}$) are derived in the appendix of the paper.

The main advantage of relative pose estimator methods is that they always estimate an accurate rotation (even in low parallax motions) and do not need the depth of the feature matches to do so. One issue is that the translation direction might not be accurate in low-parallax motion. Fortunately, we can leverage the fact that translations are not perceivable by the user in low-parallax motions. Another issue is that we cannot use robust cost functions to down-weight outlier matches because the re-projection errors are not directly part of the residual. Because of this, we need to heavily rely on the RANSAC (Fischler et al. [14]) approach that is explained in the next section 6.

The limitation of our relative pose estimator is its low accuracy when the initial guess for the rotation is not good. This is expected as we do a non-linear optimization where the initial guess is refined. This is quantitatively demonstrated in experiment 7.2.

5.2 1-DoF translation magnitude estimator

The relative pose estimator estimates unit norm translations where the magnitude (s) of the translation is unknown. This is addressed by this component, which estimates the magnitude of the translation (s) and updates the final 3-DoF translation as $\boldsymbol{t} \in R^3 := s * \boldsymbol{u} \in S^2$.

The translation magnitude is optimized using a Levenberg–Marquardt solver by minimizing the squared re-projection errors (r_{geo}) of the features with estimated depth, which are re-projected from the keyframe to the current frame.

$$\hat{s} = \arg\min_s r_{geo}, \tag{9}$$

$$r_{geo} = \sum_{i=1}^{n} g\left(\frac{\left(\pi\left(\boldsymbol{R}\boldsymbol{f_i}d_i + \boldsymbol{u}s\right) - \pi\left(\boldsymbol{f_i'}\right)\right)^2}{\sigma_i^2}\right) \tag{10}$$

Where $g()$ is a robust cost function and $\pi()$ is the camera projection function that transforms 3D points in the camera frame into 2D camera coordinates. d_i is the estimated depth of the feature in the source frame and σ_i is the uncertainty of the pixel, taken from the pyramid level where the feature was observed.

6 Implementation Details

6.1 Constant depth assumption

Note that for the very first frames, we do not know the depth d_i of the features and we assume constant depth. As soon as we pass a minimal parallax angle (which was experimentally set to 1.0 degrees), we start estimating depth by triangulating the feature correspondences. The reader might wonder what happens during the first frames in low-parallax cases. The pose estimator estimates a relative 5-DoF pose between keyframe and frame where the rotation should always be correct and the unit-norm translation might not be correct in cases of low parallax motion. However, in such cases, the translation magnitude estimator will estimate a magnitude that will be close to zero (features are far away) and therefore the relative translation between keyframe and frame will be close to zero and will not affect negatively to the global estimation (and the user will not perceive it either). The accuracy of the translation direction and the translation magnitude naturally increase as the baseline between keyframe and frame also increases. The higher the baseline, the more perceivable the translation becomes and the more features we can triangulate. We keep using the constant depth assumption until we have a minimum set of triangulated features. The reader is referred to Figure 4 for a graphical explanation.

6.2 Frame-to-frame tracker

We match features from frame-to-frame by projecting 3D points (see previous subsection), corresponding to the features in the keyframe, into the current frame and correlating a warped 8x8 pixel-size patch along the epipolar line in the current view, similar to the depth estimation stage in Forster et al. [15]. To account for pose prediction inaccuracy, the search width around the epipolar line is variable. The output of the matching stage are keyframe-to-frame feature matches ($\boldsymbol{f_i'}$) which are used for relative pose estimation (section 5.1). The frame-to-frame tracker also contains a module to refine the depth (d_i) of the tracked features in the keyframe once the estimated pose $\boldsymbol{T}$ is ready (see Figure 2). The depth of the features is used to estimate the magnitude of the relative translations (see section 5.2) and to update the 3D points corresponding to the features in the keyframe.

If a feature is lost or is not observed by the tracker, we still keep it alive for 150 frames. We do this to keep features alive for as long as possible to "simulate" that a pseudo-map is tracked and our odometry can therefore re-use old information and increase the accuracy and robustness of the tracker –specially with pure rotational motions, where the pseudo-map cannot be augmented.

6.3 Keyframe heuristics

The 6-DoF pose estimator (5-DoF from section 5.1 + 1-DoF from section 5.2) estimates the pose of the current frame with respect to a previous keyframe. Estimating the pose with respect to a keyframe, instead of the previous frame, reduces drift.

New keyframes are added based on the following heuristics: not enough inliers in the relative pose estimator –we classify a point as an inlier if its Sampson distance (Fathy et al. [12]) is small enough–, not enough overlapping features with estimated depth between keyframe and frame and high re-projection error in the translation magnitude estimator. If a keyframe has to be inserted, it is inserted in the previous frame and we estimate the 6-DoF pose between the last 2 frames. Note that at the time we do pose estimation, we don't know the depth of the features in the current frame. However, since we do a keyframe to frame optimization we do know the depth of the features in the keyframe as those were triangulated when the keyframe was processed.

6.4 Outlier removal

As with all optimization problems, the formulation of the relative pose estimator (section 5.1) is sensitive to outlier feature tracks. However, we are not dealing with a per-feature geometric error but an algebraic one, meaning we cannot use robust cost functions to down-weight outliers. Hence, we wrap the minimization in a RANSAC loop (Fischler et al. [14]) where first we initialize the relative rotation from the best estimated rotation (or the rotation prior) and the translation direction is initialized from equation 3 using the subsampled –from the inliers set– feature correspondences. We repeat this operation during 5 iterations, updating the inliers of our problem. After the 5 iteration, we additionally refine the pose (minimizing eq. 7) during 7 more iterations. After every iteration, outliers are always removed with respect to the entire set of correspondences. To determine if a point is an outlier we check if its squared Sampson distance is higher than a predefined threshold. We stop early if we reach convergence, which is obtained if the error is not reduced and the number of inliers does not increase. Otherwise, we update the best relative pose and the inliers.

The translation magnitude estimator uses only the inliers coming from the relative pose estimator. We use a robust cost function to down-weight outliers. After the estimation, we detect additional outliers by checking the re-projection error of the features. A feature is considered an outlier if the re-projection error is higher than 1.5 pixels. Outliers from both the relative pose estimator and the translation magnitude estimator are removed from the tracker in a post-processing step.

6.5 Initial guess for translation magnitude estimator

When we introduced a new keyframe in the previous frame ($n-1$), the translation magnitude is computed with respect to that frame. In this case, the estimated translation magnitude will be very close to zero, and therefore we initialize the initial guess to zero ($s_n = 0$).

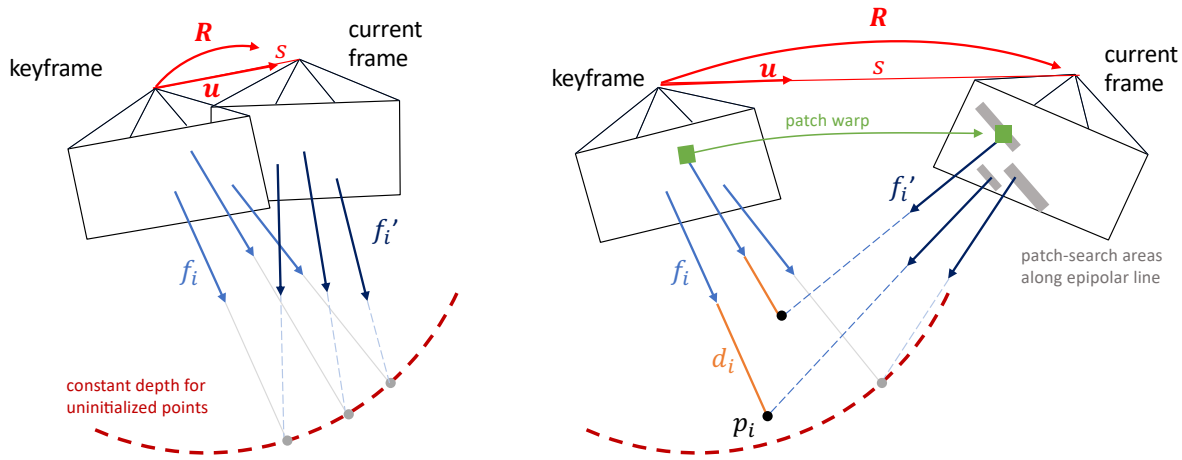

Fig. 4. Left image: For the very first frames, we cannot estimate the depth of the features. All features are used for both estimating the relative pose ($\boldsymbol{R}$, $\boldsymbol{u}$) and the translation magnitude (s). For the translation magnitude estimation, we assume constant depth for all features. Right image: Once the baseline between cameras increases, we can estimate the depth of some of the features. Again, all features are used to estimate the relative pose ($\boldsymbol{R}$, $\boldsymbol{u}$). However, only those features with estimated depth are used for estimating the translation magnitude s. Our method moves from left to right and therefore stops using the constant depth assumption once 10 features with estimated depth are available.

In the other case, when we estimate the translation magnitude with respect to a keyframe that is older than the previous frame, the current magnitude (s_n) is initialized from the last estimated magnitude ($s_n := \hat{s}_{n-1}$). However, we need to take into account that the magnitude sign may flip from frame to frame. This is because the initial guess for the translation is computed feeding the initial guess of the rotation into the closed form solution from equation 3, which can flip the sign of the translation due to its symmetry in the functional of this equation ($\boldsymbol{u}\boldsymbol{M}\left(\hat{\boldsymbol{R}}\right)\boldsymbol{u}^\intercal = (-\boldsymbol{u})\boldsymbol{M}\left(\hat{\boldsymbol{R}}\right)(-\boldsymbol{u}^\intercal)$). To solve this, we check if the sign of the 2-DoF direction has flipped from last frame ($\boldsymbol{u}_n \simeq -\hat{\boldsymbol{u}}_{n-1}$) and if that is the case we also flip the sign of the magnitude prediction ($s_n := -\hat{s}_{n-1}$).

7 Experiments

7.1 Pose estimator validation

In this section, we compare our proposed pose estimator (5-DoF (eq. 7) + 1-DoF (eq. 9)) against the gold standard solution for this problem, which is doing Bundle Adjustment (Triggs et al. [29]) without optimizing the position of the 3D points and therefore only optimize the 6-DoF pose of the camera:

$$\hat{\boldsymbol{R}}, \hat{\boldsymbol{t}} = \underset{\boldsymbol{R}\in SO(3), \boldsymbol{t}\in\mathbb{R}^3}{\arg\min} \; r, \tag{11}$$

$$r = \sum_{i=1}^{n} g\left(\frac{\left(\pi\left(\boldsymbol{R}\boldsymbol{f_i}d_i + \boldsymbol{t}\right) - \pi\left(\boldsymbol{f_i'}\right)\right)^2}{\sigma_i^2}\right) \tag{12}$$

We simulate a map of 200 landmarks and a trajectory of $\sim$1 second (37 frames) with a sigma equal to 0.75 for the pixel uncertainty. Around 170 landmarks are observed per frame and from a distance to first camera between 1 and 6 meters. The camera moves around 25 degrees and 1.0 meters on average. We use a spherical camera model (as in Kneip and Lynen [18]) with an image size of 640 by 480 and a focal length of 200 pixels. As stated by Briales et al. [4], increasing the Field Of View (FOV) makes the relative pose problem easier, as this results in a better constraining of the optimization objective. The reader is referred to Briales et al. [4] for an evaluation of the FOV in the relative pose problem.

We run 50 experiments for each algorithm. In this experiment, the estimation is always computed with respect to the first frame and keyframes do not need to be inserted since most of the landmarks are observed by the 37 frames. This ensures a fair comparison since we focus on evaluating the pure estimation and not the keyframing heuristics. We are interested in two comparisons:

- Comparing both approaches when the depth is unknown. We use the constant depth assumption and therefore fix all depths to the same value. This is to validate our proposal, showing it is particularly accurate during the first frames when depth is not available and 6-DoF approaches suffer.

- Comparing both approaches when the depth of the features is known, which is the normal use case for 6-DoF approaches. This is to confirm that our estimator works well in the normal case, meaning it is not only good for initialization but also for tracking.

The problem of the 6-DoF estimator is that the error in the estimated depth of the features is going to propagate to all the estimated parameters (rotation and translation). However, in our (5-DoF + 1-DoF) estimator, the error in the estimated depth only propagates to the translation magnitude and it does not have any influence in the estimated rotation or in the translation direction. Our results, that can be observed in Figure 7.1, confirm these hypotheses. Note that the rotational error in our approach is independent of the accuracy of the depth, which is really important for our use case as users might rotate their phones without applying any translational motion. On the other hand, the 6-DoF estimator estimates a significantly less accurate rotation if the estimated depth is unknown, confirming our hypothesis.

In Figure 7.1 we can withdraw a similar conclusion for the translation error. In this case, if the depth is unknown our estimator is also more accurate than the 6-DoF estimator. When the depth is known, both estimators behave similarly. Also, as expected, the final translation error is increased in both estimators when depth is unknown. However, not having depth has a way smaller influence in our estimator (median error increases from 3 % to 6 %) than in the 6-DoF estimator (median error increases from 3 % to 19 %). This is because depth error only propagates to the translation magnitude in our approach while it propagates to all estimated parameters in the 6-DoF estimator.

Using the same data from previous experiment, we also report the accumulated keyframe-to-frame average errors per number of frames in Figure 6. This is a very important result as it shows our approach has a low and bounded error for the entire trajectory (between 1 % and 3 %) while the classic 6-DoF estimator accuracy linearly deteriorates as the baseline (number of frames) increases. The final error for the classic estimator is one order of magnitude worse after 37 frames (between 13 % and 15 %).

These experiments confirm the validity of our approach, demonstrating that it can be a good replacement for the classical 6-DoF estimator thanks to the fact it has better properties, specially during initialization

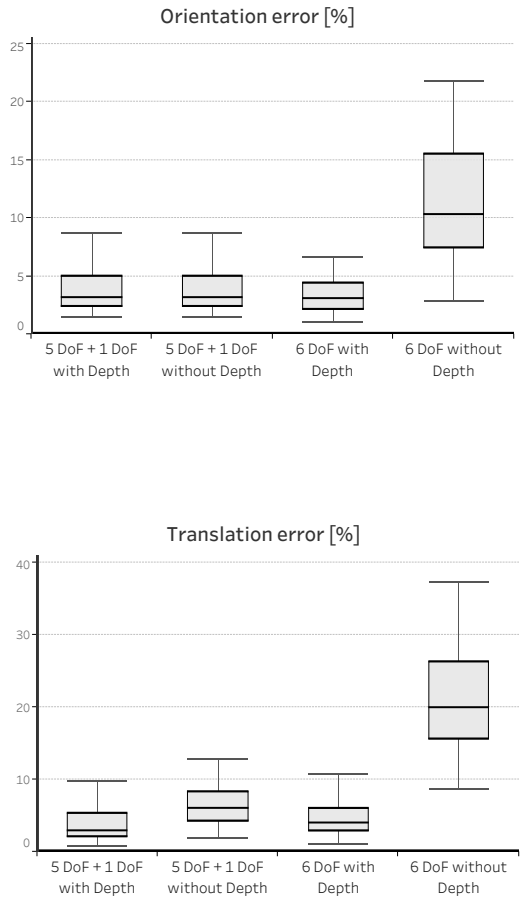

Fig. 5. Comparison between classical 6-DoF estimators and our proposal (5-DoF + 1-DoF estimator). We distinguish between two cases: 1) general case, where depth is available and 2) depth is not available – i,e. during initialization phase. Errors are reported as the ratio [%] between the maximum error (orientation or translation) and the maximum displacement (rotation or translation) between any two points in the trajectory. Our approach outperforms the classical 6-DoF estimator when depth is not available (more details in the text). The whisker plots representation is as follows: P5, P25, median, P75 and P95.

where we face low-parallax motions and features cannot be triangulated accurately.

The interested reader can also confirm the validity of our approach by looking at our video from the supplementary material (`https://youtu.be/ZGmzXK-dj1Y`).

7.2 Comparison against the state-of-the-art

We have compared our approach against state-of-the-art approaches for the relative pose problem. The TUM benchmark has been used for the evaluation.

7.2.1 Dataset for the relative pose problem

We release a dataset that we have generated from TUM benchmark (Sturm et al. [28]) to evaluate relative pose estimators and therefore facilitate the comparison against future solutions for the relative pose problem. To this purpose, we use six recordings from TUM benchmark – two from each different *Freiburg* Kinect sensor. We run ORB-Slam (Mur-Artal et al. [24]) in every recording and we store the keyframe-to-frame bearing correspondences –after removing potential outliers with our RANSAC approach– into text files. The corresponding ground-truth pose from keyframe-to-frame is also provided for quantitative evaluation. We also generate the same feature correspondences but without noise to facilitate the debugging when first using this dataset. We produce the noiseless correspondences by projecting the 3D map points associated to those correspondences into the keyframe and the frame. We subsample the dataset to have an affordable number of keyframe-to-frame pairs per recording, resulting in around 300 pairs per recording on average. The dataset can be downloaded from `https://github.com/alejocb/relative_pose_dataset`.

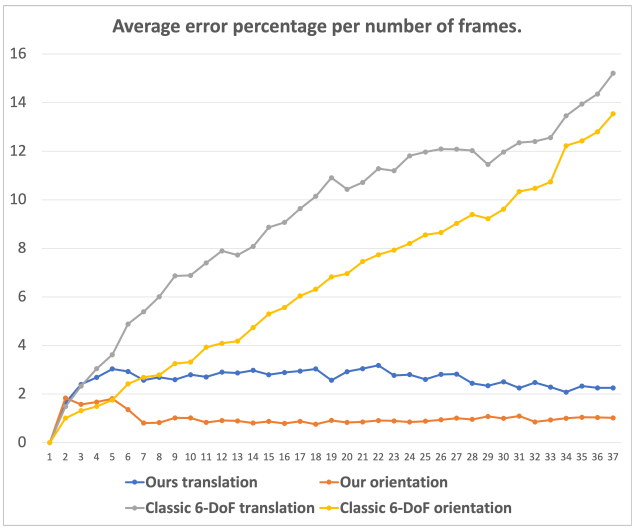

Fig. 6. Orientation and translation error per number of frames. We compare our approach against the classic 6-DoF estimator when depth is not available. Out approach has a bounded average error while the error of the classic 6-DoF estimator linearly increases with the frame number. Errors are reported as the ratio [%] between the average absolute error in a frame and the maximum displacement (orientation or rotation) between any two points in the trajectory.

7.2.2 Evaluation with TUM dataset [28]

In this section, we compare our proposal for the relative pose estimator problem against the OpenGV implementation (`https://laurentkneip.github.io/opengv/`) provided by Kneip and Lynen [18] and QCQP solution from Briales et al. [4]. QCQP code is not available online but it was provided by the authors. We use the dataset that we explained in the previous subsection 7.2.1 for this evaluation.

Current implementation of the QCQP solution from Briales et al. [4] does not admit initial guesses while our proposal and Kneip and Lynen [18] proposal do admit initial guesses. As a first experiment, we evaluate the importance of the initial guesses. We compare the three approaches in a recording from TUM benchmark (Sturm et al. [28]) and report the results we obtained with different accuracy levels of the initial guesses.

The initial guesses are generated as follows:

$$R_{\text{guess}} = \exp_{so(3)}\left(\log_{SO(3)}(R_{\text{gt}}) * (1.0 - \Gamma)\right) \tag{13}$$

Where R_{gt} is the ground-truth relative pose between both cameras and Γ goes from 0.0 (perfect initial guess) to 1.0 (inaccurate initial guess). In our figures, an initial guess error of $X\%$ means $\Gamma = 0.01X$ in this equation.

As it can be observed in Figure 7, our approach and Kneip and Lynen [18] approach are quite inaccurate when the initial guesses are not good. This is expected as these approaches are non-linear optimizations that normally find a local minimum close to the initial guess. QCQP solution from Briales et al. [4] principal advantage is that is a solution that comes with global optimality guarantees making this solution the preferred one, specially if initial guesses are inaccurate.

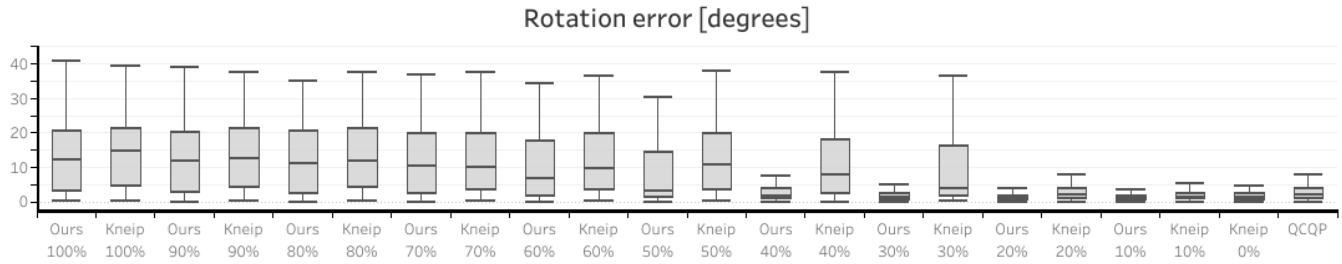

Fig. 7. Comparison between QCQP approach from Briales et al. [4], Kneip and Lynen [18] and our approach. We have used the TUM dataset "fr3 structure texture near" for this comparison. The percentage number specifies the error of the initial guess (see equation 13 for the definition of the initial guess error). Our approach consistently outperforms Kneip and Lynen [18] and it also outperforms QCQP if the initial guess error is 30% or smaller. The whisker plots representation is as follows: P5, P25, median, P75 and P95.

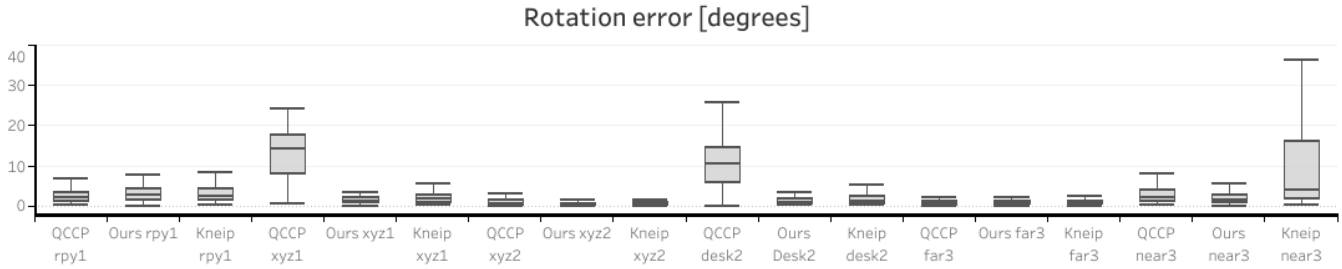

Fig. 8. Comparison between QCQP approach from Briales et al. [4], Kneip and Lynen [18] and our approach for 6 datasets from TUM benchmark (Sturm et al. [28]). Our approach outperforms both QCQP approach from Briales et al. [4] and Kneip and Lynen [18] in most of the recordings. We have used an initial guess error of 30% in this experiment. See equation 13 for the definition of the initial guess error. The whisker plots representation is as follows: P5, P25, median, P75 and P95. Legend of the datasets is as follows: rpy1 (fr1 rpy), xyz1 (fr1 xyz), xyz2 (fr2 xyz), desk2 (fr2 desk2), far3 (fr3 structure texture near) and near3 (fr3 structure texture far).

However, the global minimum is not guaranteed to be the one that achieves the most accurate pose, specially in ill-posed problems. For this reason, Kneip and Lynen [18] and our approach obtain comparable or even better results than QCQP approach from Briales et al. [4] if the initial guesses are good enough. According to these experiments, our approach can handle initial guess errors of up to 30%, which is good enough as the initial guesses are computed from the previous frame, meaning the potential Γ we need to handle is equal to $(N-1)/N$ and therefore will be significantly bigger than 0.3. Where N is the number of frames between keyframe and frame. Note that our approach has better convergence properties than Kneip and Lynen [18], it is able to converge to an accurate solution using worse initial guesses. This is mostly due to our second contribution of the paper, which is combining the functional and its Jacobians in the residual which helps finding the global minimum easier. We analyze this weight in more detail in the section 7.3.

In a second experiment (see Figure 8), we compare the three approaches in six recordings from the TUM benchmark. In this experiment, we have used the initial guesses with 30% of error. Note that our approach outperforms both QCQP approach from Briales et al. [4] and Kneip and Lynen [18] in most of the recordings.

7.3 Validation of the proposed residual

The previous experiment from section 7.2 validates our proposal for the residual of the iterative relative pose problem. However, we include a second experiment to further analyze the influence of our contribution in the formulation. The influence of our proposal can be tuned with the weight W. For this experiment, we use synthetic data and same camera model as in the experiments from section 7.1. We create a random set of 2D-2D correspondences between two different camera views (with uncertainty equal to 0.75 pixels). We perturb the poses and the initial guesses for the rotation and translation direction contain an error of around 10 degrees and 50 degrees in average, respectively. We run the relative pose estimator and compute the error as a function of the weight W (which ranges from 0 to 1000). We run 50 experiments per weight.

Observe in Figure 10 that the best results are obtained with weights bigger than zero. However, note that we get big errors if we use huge values for the weight. In this case, the problem becomes ill-posed as the functional (1-D) will have a way higher influence than the Jacobians (5-D) and the residual will become effectively 1-D while the number of parameters to estimate is still 5. Having a weight equal to zero results in the original residual (with only the Jacobians), resulting in high errors in average when compared against our approach.

The new residual is particularly helpful when the initial guess of the relative pose is far from the optimal one. In this case, the convergence properties are improved as the functional is included in the residual. As expected, if we repeat this test with good initial guesses for the relative pose, the results obtained for both residuals are good and very close to each other.

We do an additional experiment with real data to further validate the inclusion of the functional in the residual. The reader is referred to Figure 9 for more details.

7.4 Complexity of our approach

Our pipeline runs in real-time in both low-end and high-end devices. We have measured the compute of our full visual odometry (tracker and pose estimator) in a Samsung-S10 and in a Huawei-P20. We have obtained average compute times of $\sim$10 and $\sim$25 ms respectively. The frame to frame tracker and the pose estimator represent the $\sim 60\%$ and $\sim 40\%$ of the total compute respectively. While Kneip and Lynen [18] estimator also works in real time and has a similar complexity as our approach, QCQP approach from Briales et al. [4] is implemented in *Matlab* and it takes several seconds of compute per frame.

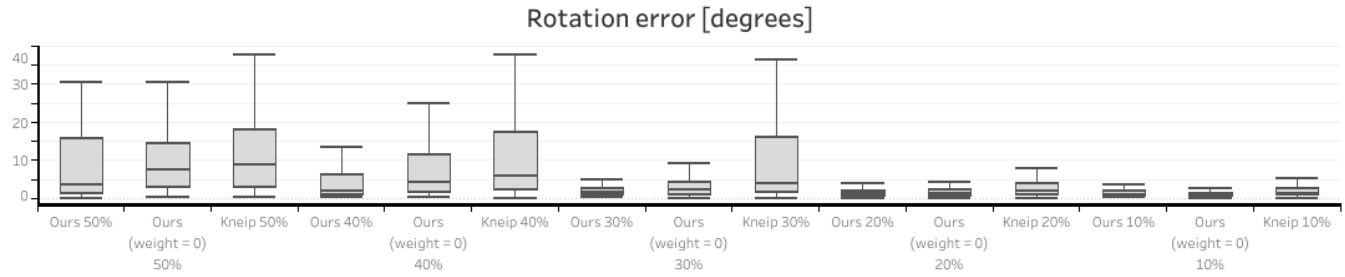

Fig. 9. Evaluation of the addition of the functional in the residual. Experiment using TUM dataset "fr3 structure texture near". The percentage number indicates the initial guess error (see equation 13). Note our results are worse if the functional is not included in the residual (weight = 0). Our approach can converge for higher errors of the initial guess when compared against Kneip and Lynen [18] approach. The whisker plots representation is as follows: P5, P25, median, P75 and P95.

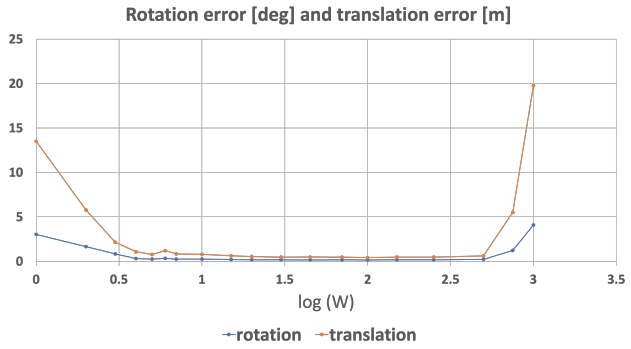

Fig. 10. Positional and rotational errors as a function of the weight W proposed in equation 7. The Figure shows that our proposal does improve the results significantly in optimizations where the initial guess is far from the actual solution. The improvements are up to a factor of 10. Also, the estimator is quite robust to this weight, values between 15 and 250 obtain a similar level of accuracy.

7.5 Failure cases.

As demonstrated in the experimental section, the main failure case of our approach is the low accuracy achieved when the initial guess is far from the true value. Even though we tried to mitigate this issue by robustifying our residual, we still have convergence issues and working on a more robust solution remains for future work.

Similarly to most relative pose estimators in the literature, our relative pose estimator is very sensitive to outliers. Even though we do not handle outliers in the estimator itself, we wrap the relative pose estimator inside of a RANSAC scheme to mitigate this issue.

8 Conclusions and Future Work

In this paper, we have proposed a novel monocular visual odometry that initializes without motion parallax and estimates a 6-DoF pose from first frame on. This is achieved by combining a relative pose estimator and a translation magnitude estimator. We have shown this estimator outperforms the classical 6-DoF estimator in initialization stages with low parallax motion. We believe the usage of relative pose estimators has a great potential in monocular SLAM due to their ability of not failing with pure rotational motions or with poor map estimation. For the relative pose problem, we have also proposed a new residual combining the Jacobians of the functional and the functional itself. The residual is minimized using a Levenberg–Marquardt optimizer. We demonstrate that minimizing both errors is more accurate than minimizing only the Jacobians.

There are multiple opportunities for future work:

- finding an analytic solution for the introduction of feature uncertainties and robust cost functions to down-weight outliers in the functional to minimize in the relative pose estimator.
- estimating both translation magnitude and depth of the features in the 1-DoF estimator to achieve more accurate results.
- merging the translation magnitude and relative pose estimators into a single probabilistic estimation module that can consume both features with depth and without depth instead of decoupling the estimation in two different modules.

9 Appendix

In this section we derive the Jacobians of the functional with respect to the so(3) rotation parameters $\frac{\partial \boldsymbol{f}}{\partial \theta_i} = \frac{\partial \boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^\intercal}{\partial \theta_i}$ and the translation direction 2-DoF manifold $\frac{\partial \boldsymbol{x}\boldsymbol{C}\boldsymbol{x}^\intercal}{\partial \beta_i}$ that are used as part of the residual to minimize in equation 7.

We apply the chain rule to derive them. First, we compute the derivative with respect to $\boldsymbol{x}$:

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = 2\boldsymbol{C}\boldsymbol{x} \tag{14}$$

The Kronecker product (Van [30]) is used to get the derivative with respect to the translation direction $\boldsymbol{u}$ and the vectorized form of the rotation ($\boldsymbol{r} = \text{vec}(\boldsymbol{R})$).

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{r}} = \text{kroneckerProduct}\,(\boldsymbol{u}, \boldsymbol{I}_{9x9}) \tag{15}$$

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}} = \text{kroneckerProduct}\,(\boldsymbol{I}_{3x3}, \boldsymbol{r}) \tag{16}$$

Relative of the rotation matrix with respect to each axis $i = x, y, z$:

$$\frac{\partial \boldsymbol{r}}{\partial \theta_i} = \text{vec}\,(\boldsymbol{R}[\text{unit}_i]_x) \tag{17}$$

$[]_x$ is the skew-symmetric matrix and unit_i is the unit vector in the direction of the i axis.

The translation direction is parametrized using the z-axis (third column) of a rotation matrix ($\boldsymbol{R}_{\boldsymbol{u}} \in SO3()$). Therefore, its Jacobian is computed as follows:

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\beta}} = \text{vec}\left([\boldsymbol{R}_{\boldsymbol{u}}[\text{unit}_z]_x]_{\text{topBlock(3,2)}}\right) \tag{18}$$

The final Jacobians are computed applying the chain rule:

$$\frac{\partial \boldsymbol{f}}{\partial \theta_i} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{r}} \frac{\partial \boldsymbol{r}}{\partial \theta_i} \tag{19}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\beta}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\beta}} \tag{20}$$

REFERENCES

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] A. M. Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

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

[4] J. Briales, L. Kneip, and J. Gonzalez-Jimenez. A certifiably globally optimal solution to the non-minimal relative pose problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 145–154, 2018.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[6] A. Concha and J. Civera. Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5686–5693. IEEE, 2015.

[7] A. Concha and J. Civera. Rgbdtam: A cost-effective and accurate rgb-d tracking and mapping system. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 6756–6763. IEEE, 2017.

[8] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

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

[10] J. M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera. Cam-convs: Camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11826–11835, 2019.

[11] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza. Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor. In *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 1722–1729. IEEE, 2015.

[12] M. E. Fathy, A. S. Hussein, and M. F. Tolba. Fundamental matrix estimation: A study of error criteria. *Pattern Recognition Letters*, 32(2):383–391, 2011.

[13] O. D. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(03):485–508, 1988.

[14] 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*, 24(6):381–395, 1981.

[15] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2016.

[16] M. Garcia-Salguero, J. Briales, and J. Gonzalez-Jimenez. Certifiable relative pose estimation. *Image and Vision Computing*, 109:104142, 2021.

[17] J. Huang, R. Liu, J. Zhang, and S. Chen. Fast initialization method for monocular slam based on indoor model. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2360–2365. IEEE, 2017.

[18] L. Kneip and S. Lynen. Direct optimization of frame-to-frame rotation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2352–2359, 2013.

[19] S. H. Lee and J. Civera. Rotation-only bundle adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 424–433, 2021.

[20] J. Li, H. Bao, and G. Zhang. Rapid and robust monocular visual-inertial initialization with gravity estimation via vertical edges. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6230–6236. IEEE, 2019.

[21] H. C. Longuet-Higgins. The reconstruction of a plane surface from two perspective projections. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 227(1249):399–410, 1986.

[22] J. J. Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pp. 105–116. Springer, 1978.

[23] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[24] 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*, 33(5):1255–1262, 2017.

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

[26] J. Park and S. Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. *arXiv preprint arXiv:1703.07870*, 2017.

[27] J. Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM review*, 6(4):422–430, 1964.

[28] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580. IEEE, 2012.

[29] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pp. 298–372. Springer, 1999.

[30] C. F. Van Loan. The ubiquitous kronecker product. *Journal of computational and applied mathematics*, 123(1-2):85–100, 2000.