Visual Inertial Stereo Odometry Based on Vertical Lines

Jesus Pestana[1], Thomas Holzmann[1] and Friedrich Fraundorfer[1]

***Abstract*— This paper introduces a novel visual inertial odometry approach, which uses a direct stereo visual odometry algorithm and tightly couples it with inertial measurements. Our improved direct stereo visual odometry algorithm estimates the pose of consecutive frames by minimizing the photometric error of patches around vertical lines. Coupled with pre-integrated inertial measurements, the system state is optimized in a joint visual-inertial optimization framework. In an experimental evaluation we show that tightly coupled inertial measurements significantly improve the pose estimation quality compared to vision-only algorithms, and that reliable pose estimation is also possible in poorly textured environments. Further, we demonstrate comparable results to a state-of-the-art method.**

I. INTRODUCTION

Two so far separated trends recently emerged to solve the visual ego-motion estimation problem, tightly coupled visual-inertial methods [1], [2] and direct methods [3], [4], [5]. Tightly coupled visual-inertial methods estimate the combined state of the Inertial Measurement Unit (IMU) and vision parameters while minimizing a cost function, which is the addition of IMU residuals and image residuals. Direct methods on the other hand perform direct or dense image alignment without a feature extraction or feature matching step.

In this paper we propose to combine these two recent techniques, in particular to tightly couple direct visual odometry using a stereo camera with inertial measurements. The visual odometry (VO) part of our work is based on our prior work proposed in [10], which is a direct VO approach that focuses on vertical lines, motivated by their abundance in man-made environments. For estimating the combined visual-inertial state local-window based optimization, fixed-lag smoothing is used. The cost function to be optimized contains inertial residuals as well as visual photometric residuals.

II. RELATED WORK

Traditional Visual Odometry (VO) and visual Simultaneous Localization and Mapping (SLAM) algorithms detect and match visual features (preferably point features) in order to estimate the pose of successive camera frames. A popular point feature-based SLAM approach is PTAM [6]. As it is a full SLAM system, it does not just estimate the relative motion from frame to frame, but estimates the pose of the camera with respect to a simultaneously created global map. It is a monocular approach, which just uses one camera.

This project has been supported by the Austrian Science Fund (FWF) in the project V-MAV (I-1537)

[1]Institute for Computer Graphics and Vision, Graz University of Technology (Austria)
{pestana, holzmann, fraundorfer}@icg.tugraz.at

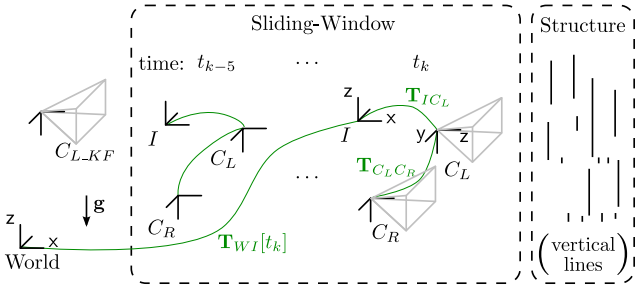

Fig. 1. Overview of our system. We solve a tightly-coupled IMU and direct image alignment problem by tracking only gravity-aligned lines. Our acquisition system consists of an IMU, frame I, rigidly attached to a stereo camera, left-camera C_L and right camera C_R frames. The internal state of our VINS, includes a keyframe image, frame $C_{L_{KF}}$, from the left camera and a sliding-window of recent vision-only relative pose measurements, IMU states and measurements.

Contrary, Libviso [7] is a stereo approach, which uses stereo images of a calibrated stereo setup as input. It is a pure VO method and just estimates the relative motion from frame to frame. Similarly to PTAM, it also uses point features. Instead of point features, also line features can be used for pose estimation. For example, Elqursh et al. [8] propose to use lines of a special line configuration to estimate the relative pose of two cameras. With this method, it is possible to estimate camera motion where nearly no texture exist. However, as line detection is quite expensive in terms of computational cost, it is hard to run such a method in real-time on a lightweight computer.

Recently, direct VO and SLAM methods have become popular. Instead of detecting and matching features in order to estimate a camera pose, direct methods directly minimize the photometric error of the whole or parts of an image to find an optimal camera pose.

One of the first real-time capable direct approaches was DTAM [4]. This method tracks a camera given a 3D model of the scene by minimizing the photometric error of the whole current frame. However, to work in real-time, a computer with a powerful GPU was necessary. A similar approach is used in LSD-SLAM [3]. It tracks a camera using direct methods and simultaneously creates the model of the scene. As it is not using the whole image but just parts of the image with sufficiently high gradient values, it can significantly reduce the computation costs. Therefore, it can run in real-time even on smartphones.

An approach which uses both, feature detection and direct pose estimation, is proposed in [5]. It detects point features in keyframes and directly estimates the pose of frames between keyframes by minimizing the photometric error of patches around the detected feature points. As it uses feature-

based and direct methods, it is called Semi-Direct Visual Odometry (SVO).

Nowadays, many devices are also equipped with an IMU and can simultaneously acquire images. Therefore, there arises an interest on methods that use both, image and IMU measurements, to estimate the pose of a combined sensor head. The resulting estimation frameworks are referred as Visual Inertial Navigation Systems (VINS) and are commonly subdivided depending on the type of optimization that is applied, in batch non-linear minimization and recursive filtering; and depending on the depth of the integration between the visual and inertial parts, in loosely and tightly coupled approaches.

Loosely-coupled VINS are characterized by utilizing a VO or SLAM framework as a black box, which is not directly utilizing the IMU measurements in the visual optimization problem. Konolige et al. [9] proposed a loosely-coupled VINS in which the IMU was utilized as an inclinometer, providing information about the gravity direction and the heading to a VO framework, to estimate the pose of a ground vehicle. Weiss et al. [10] used PTAM to deliver pose measurements to an Extended Kalman Filter (EKF) that utilized IMU measurements to propagate the state of a flying robot. This specific type of VINS is amenable to precise mathematical representations from Control Theory, enabling the realization of non-linear observability analysis based on the Lie derivatives. Based on these mathematical grounds, the works [11], [12] establish which motions the IMU must undertake to render the inter-sensor spacial calibration [11] and the scale of the visual SLAM position estimates [12] observable on the acquired data.

Tightly-coupled VINS solve a joint optimization problem, where error terms originate from the IMU measurements and from the camera images. Regarding recursive filtering approaches, the Multi-State Constraint Kalman Filter (MSCKF) [2] integrates tracked monocular visual features from multiple images on the IMU-propagated state of an EKF, without requiring the addition of the observed feature positions into the filter state. More recently, ROVIO [13], a novel VINS method proposed the usage of direct pixel intensity errors and the inclusion of the related image patch locations in the filter state expressed in robot frame coordinates. Regarding tightly-coupled batch optimization VINS approaches, OKVIS [1] utilizes visual reprojection errors and specified in detail the relative weighting of visual and inertial residuals. OKVIS stores a set of keyframes and a sliding-window with recent images and IMU measurements to jointly optimize the related IMU states. Lupton et al. [14] proposed a method to pre-integrate the IMU state and covariance propagation, and utilized it to speed-up a batch optimization VINS algorithm. This IMU pre-integration method has been recently used [15] to add IMU constraints in a graph-SLAM framework, using a purely visual approach as odometry method.

The contributions of this work are the following. Aside of our particular line-based visual odometry algorithm [16], our work presents important differences from current state-of-the-art algorithms:

1) Compared to [1], [14]: we propose the utilization of statistically independent vision-only pose measurements instead of considering the corresponding visual residuals from recent images.
2) Compared to [1]: we propose the utilization of IMU pre-integration in order to speed-up the re-evaluation of inertial residuals.
3) Compared to [15]: our inertial residuals are used in the odometry VIO batch optimization, instead of in the front-end graph-SLAM framework.

III. SYSTEM OVERVIEW

In this paper, we propose a tightly-coupled visual-inertial odometry algorithm. The vision part is a direct visual odometry algorithm, which detects gravity-aligned lines with a fast line detection algorithm aided by an IMU and estimates the pose of the camera by minimizing the photometric error of patches around lines, as originally proposed in [16]. Inspired on OKVIS [1], our VINS utilizes batch optimization to jointly minimize the following error terms: direct visual residuals on the current image [16], and a set of recent inertial and relative pose measurement residuals stored in a local sliding-window, see Fig. 1.

IV. VISUAL ODOMETRY

The visual odometry algorithm used in our work is based on the visual odometry approach proposed in [16]. It is a direct stereo visual odometry method which uses lines and is therefore especially suited for applications in man-made environments. We have made some adaptations to this algorithm to make it more robust and accurate compared to the original version.

A. Direct Stereo Visual Odometry Based on Lines

In this section, we will briefly discuss the visual odometry algorithm introduced in [16]. For more details, we refer the reader to the original paper. Our visual odometry algorithm uses stereo images and IMU measurements as input. In selected keyframes, 3D information (i.e., lines) is triangulated within the stereo setup. This information is then used to directly estimate the pose of the camera by aligning patches around lines in order to minimize the photometric error.

1) Vertical Line Detection and Matching: Using the IMU measurements, it is possible to align the images with the gravity direction, which means that lines parallel to the gravity direction in reality are mapped to vertical lines in the image. Fig. 2 illustrates the alignment process.

Having gravity aligned images, vertical lines can be detected by analyzing the gradient in the horizontal direction. For this, we convolve the image with a Sobel filter and create a histogram of gradient values for each column of the image. Columns having enough similar gradient values are assumed to contain a vertical edge, which we refer as line. Start and end point of the line are detected and finally a non-maximum suppression is applied. An example image with detected lines can be seen in Fig. 3.

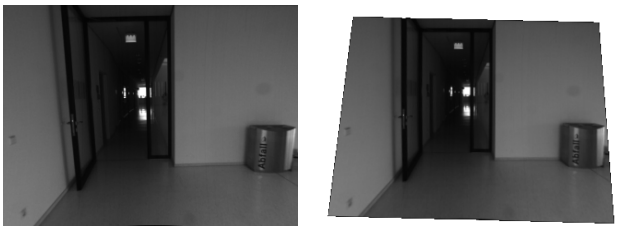

Fig. 2. Image alignment with the gravity direction. *Left:* Unalinged image. *Right:* Image aligned to the gravity direction. As can be seen, lines parallel to the gravity direction in the scene are projected to vertical lines in the image. Images taken from [16].

The matching of lines is done only in the calibrated stereo setup. Corresponding lines are searched within a predefined disparity range by comparing the gradient histogram bin responsible for line detection and by computing the Sum of Absolute Differences (SAD) of the patches around lines of a potential match. The lines with the lowest SAD value and an SAD value below a predefined threshold are accepted as match. This line is then triangulated within the calibrated stereo setup in order to retrieve the corresponding 3D line.

2) Direct Pose Estimation: Direct pose estimation is the process of estimating the pose of a camera by directly minimizing the photometric error of the new image with respect to a previous image. In our algorithm, we want to estimate the movement of the camera with respect to the previous keyframe. For this, we apply a non-linear least squares optimization on the image intensities. We keep the 3D lines fixed in the optimization step, as we assume to have already accurately triangulated 3D information from the stereo setup.

We aim to minimize the quadratic photometric error of patches around lines (as illustrated in Fig. 3). The quadratic photometric error per pixel is defined as:

$$r_i^2(\mathbf{T}) = \left(\boldsymbol{I}_{kf}(\boldsymbol{p}_i) - \boldsymbol{I}\left(\pi\left(\boldsymbol{p}_i, d_{kf}(\boldsymbol{p}_i), \mathbf{T}\right)\right)\right)^2, \quad (1)$$

where $\boldsymbol{I}_{kf}(\boldsymbol{p}_i)$ is the intensity value of a pixel in a keyframe and $\boldsymbol{I}(\pi(\boldsymbol{p}_i, d_{kf}(\boldsymbol{p}_i), \boldsymbol{\xi}))$ is the intensity value of the pixel of keyframe pixel warped to the current frame by the warping function $\pi(\cdot)$. d_{kf} defines the depth of the pixel p_i in the keyframe and $\mathbf{T} = \mathbf{T}_{C_{L\_KF}C_{Lk}}$ is the transformation between keyframe and current frame.

As some pixels might be outliers and have a significantly higher residual error, we downweight these outliers by using a Cauchy loss function.

3) Keyframe Selection: Having exceeded a certain amount of movement, a new keyframe has to be selected. Selecting too many keyframes can result in a higher drift, while too few keyframes may result in poor pose estimates. Compared to [16], we introduce a novel keyframe selection scheme which is explained in detail in IV-B.5.

4) Map Propagation: Before triangulating new 3D information at a keyframe, the lines of the previous keyframe are projected into the current one and are reused as lines also in this frame when they fulfill a sanity check (see Sec. IV-B.1). Afterwards, new lines are just detected in the parts of the image where no accepted lines of the previous image exist.

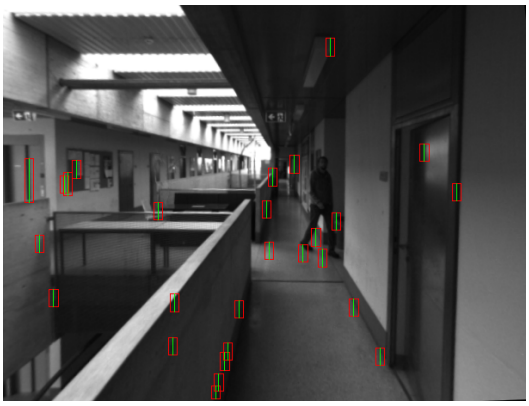

Fig. 3. Detected lines (green) and patches around lines (red) used in direct pose estimation. Not all lines are actual lines in the scene. However, as they are positioned within regions with high gradient, they are well suited for direct pose estimation.

B. Improvements and Adaptations

Compared to the version described in [16] we made some improvements and adaptations to the algorithm, which we describe in this section.

1) Improved Line Tracking: In the original version, existing 3D lines from the previous keyframe are projected into the new keyframe and assigned to a nearly detected line. This may lead to inaccurate 3D-2D line matches, as lines are not necessarily detected at exactly the same location (especially when the image is not perfectly aligned with gravity). Therefore, we propose a new approach to track already triangulated lines: We project the lines of the previous keyframe into the current keyframe and compare the patch around the line using SAD. If the SAD error is lower than half of the error defined for line matching in the stereo setup, the projected line is accepted as line in the new keyframe. Otherwise, the line is rejected.

2) Line Feature Selection: In high-gradient regions of the image, a lot of small lines might be detected, which contain mainly redundant information which does not improve the pose optimization but just makes it slower. Therefore we aim to reduce this redundant information: We partition the keyframe into 5x5 equally-sized cells and just add new lines to a cell, where less than 5 already triangulated lines exist.

3) Subpixel Refinement for Line Matching: As we triangulate the 3D information in the calibrated stereo setup, we assume an accurately determined baseline. However, as this baseline is usually just between $10 - 15\ cm$, the triangulation result gets already quite inaccurate after a few meters of depth when using just pixel accurate line matching. Therefore, we propose a simple match refining technique: Having computed a line match, we shift the detected line between -1 and $+1$ pixel along the epipolar line with steps of 0.1 pixel, interpolate the patch around the shifted line using bilinear interpolation and compute the SAD of each shifting step with the matched line. The shift with the lowest SAD error is used as correction for the line match.

4) Multi-View Depth Optimization: Even though when using subpixel refinement, the estimated depth still might not be exact due to the small baseline. Therefore, we introduce a

multi-view depth optimization scheme inspired by [17]: For every line, we propagate a depth uncertainty and the currently estimated depth to the consecutive frames. In every new frame, the depth and the uncertainty gets updated depending on the triangulation angle. If the uncertainty falls under a threshold, the depth gets fixed and will not be updated anymore.

5) Keyframe Selection: Originally, our approach was selecting keyframes by the average photometric error per pixel after optimization and after a certain amount of rotational and translational movement. Instead of the average photometric error, we propose to use a different measure: We propose to use the ratio between intensity residuals which were not downweighted by the loss function (called inliers) and intensity residuals which were downweighted by the loss function (called outliers). We select a new keyframe, if less than 20 % of the residuals are inliers. Additionally, we select a new keyframe if less than 5 lines are used in the pose optimization. In comparison to the original approach, this approach makes the parameter selection easier, as a good ratio is easier to determine than an average cost threshold. Further, it is slightly more robust against outliers, as outliers still influence the average cost, but not this ratio.

6) Constant Velocity Assumption: Direct methods generally have problems when the inter-frame motion gets too big. However, as it can be assumed that the velocity stays relatively constant between two consecutive frames [18], we can use the motion of the two previous frames as the initialization for the current direct pose estimation step. Using this assumption, also bigger motions can be handled easier, as the pose estimation already starts at a better estimate. As this assumption can also deliver a wrong initialization, we just use it if more than 5 lines are available for direct pose estimation. This reduces the cases of complete failure (trajectory jumping due to wrong initialization) significantly. We use the constant velocity assumption only in the vision-only odometry part used in the comparison.

7) No Point Features Used: Due to our improvements, using lines only does not drastically decrease the accuracy of the pose estimates compared to using lines and points. Additionally, as we aim to use a lightweight vision solution for combining it with IMU measurements, we chose to use only lines in the experiments for this paper.

V. VISUAL-INERTIAL ODOMETRY

In this section we propose a cost function whose derivation is based on the Maximum a Posteriori (MAP) method, which is closely related to the Maximum Likelihood principle. We consider the VINS shown in Fig. 1, from whom we want to estimate its IMU pose, velocity and internal state over time.

Rotation parameterization: in this work, Hamilton convention quaternions [19] are used to represent rotations, i. e. $\mathbf{R}_{WI} = \mathbf{R}\{\mathbf{q}_{WI}\} \in SO(3)$ and $\mathbf{q}_{WI} \in \mathbb{Q}$, resulting in the IMU pose $\mathbf{T}_{WI} = \{\mathbf{q}_{WI};\, \mathbf{p}_I|_W\} = \{\mathbf{q}_{WI};\, \mathbf{p}_I\} \in SE(3)$. We denote the coordinate transformation composition as $\mathbf{T}_{WC_L} = \mathbf{T}_{WI}\mathbf{T}_{IC_L}$. The notation $\cdot|_F$ specifies that a vector is expressed with respect to frame F.

All angular errors are specified in global coordinates, i. e. $\delta\boldsymbol{\theta} = \delta\boldsymbol{\theta}|_W$. The conversion between angular errors and quaternions is cumbersome, specifically in the definition of optimization residuals. For this reason, and using the global rotation perturbation convention, we introduce the operator $\ominus$ specified in Eq. (2c). For real numbers and vectors this operator is equivalent to the subtraction operation.

$$\mathbf{q}\{\boldsymbol{\theta}\} = [q_w; \mathbf{q}_v] = [\cos(\theta/2); \sin(\theta/2)\mathbf{u}_\theta] \tag{2a}$$

$$\boldsymbol{\theta}\{\mathbf{q}\} = \{\theta = 2\arctan2\left(||\mathbf{q}_v||_2, q_w\right),\, \mathbf{u}_\theta = \mathbf{q}_v/||\mathbf{q}_v||_2\} \tag{2b}$$

$$\mathbf{q}_2 \ominus \mathbf{q}_1 \doteq \boldsymbol{\theta}\{\mathbf{q}_2 \otimes \mathbf{q}_1^{-1}\} = \boldsymbol{\theta}_{12},\ \mathbf{q}_2 = \mathbf{q}\{\boldsymbol{\theta}_{12}\} \otimes \mathbf{q}_1 \tag{2c}$$

Clock synchronization: The following simplified notation, $\cdot[t_k] = \cdot_k$, is utilized in this paper for all variables at the same timestamp t_k, independent of their origin, e. g. for the IMU state $\mathbf{x}[t_k] = \mathbf{x}_k$. Its physical validity relies on the clock synchronization across sensors, which renders their generated timestamps comparable with each other.

IMU measurement model: The IMU measures the rotation velocity $\boldsymbol{\omega}_m \approx \boldsymbol{\omega}|_I$ and the specific acceleration $\mathbf{a}_m \approx (\mathbf{a} - \mathbf{g})|_I$ of frame I. The resulting measurements are affected by Gaussian noises, $\boldsymbol{\omega}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\omega n})$ and $\mathbf{a}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{an})$; and slowly time-varying biases, $\boldsymbol{\omega}_b$ and $\mathbf{a}_b$. The biases are modeled as random walk noises, i. e. $\dot{\boldsymbol{\omega}}_b = \boldsymbol{\omega}_{bn}$ with $\boldsymbol{\omega}_{bn} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\omega bn})$ and $\dot{\mathbf{a}}_b = \mathbf{a}_{bn}$ with $\boldsymbol{a}_{bn} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{abn})$. The result is the following IMU measurement model:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \boldsymbol{\omega}_b + \boldsymbol{\omega}_n \tag{3a}$$

$$\mathbf{a}_m = \mathbf{R}_{IW}(\mathbf{a} - \mathbf{g}) + \mathbf{a}_b + \mathbf{a}_n \tag{3b}$$

IMU state and error-state: The state, $\mathbf{x}_k$, is composed by the IMU pose, $\mathbf{T}_{WI}$, velocity, $\mathbf{v}_I = \mathbf{v}_I|_W \in \mathbb{R}^3$ and the accelerometer and gyroscope IMU biases, respectively $\mathbf{a}_b \in \mathbb{R}^3$ and $\boldsymbol{\omega}_b \in \mathbb{R}^3$; resulting in, $\mathbf{x}_k = [\mathbf{p}_I; \mathbf{v}_I; \mathbf{q}_{WI}; \mathbf{a}_b; \boldsymbol{\omega}_b]$. The gravity vector, $\mathbf{g} = \mathbf{g}|_W$, is estimated as a separate parameter, shared by all IMU states during integration of the inertial measurements. The IMU error-state, $\delta\mathbf{x} = [\delta\mathbf{p}_I; \delta\mathbf{v}_I; \delta\boldsymbol{\theta} = \delta\boldsymbol{\theta}_{WI}|_W; \delta\mathbf{a}_b; \delta\boldsymbol{\omega}_b; \delta\mathbf{g}]$, includes the gravity in order to consider the effect of its uncertainty into the overall IMU state covariance propagation. The estimation covariance of the IMU state, $\boldsymbol{\Sigma} = \mathbb{E}\left[\delta\mathbf{x}^\intercal \delta\mathbf{x}\right] \in \mathbb{R}^{18\times18}$, is expressed and propagated using the error-state kinematics, similarly to how this process is performed in an Error-State Kalman Filter (ESKF) [19].

IMU state prediction through direct-integration: The integration of the differential equations (4a-e), starting at a given IMU state $\mathbf{x}_k$ by means of the intervening IMU measurements, results in a prediction of the IMU state at the end of the integration interval.

$$\dot{\mathbf{p}}_I = \mathbf{v}_I \tag{4a}$$

$$\dot{\mathbf{v}}_I = \mathbf{R}_{WI}(\mathbf{a}_m - \mathbf{a}_b - \mathbf{a}_n) + \mathbf{g} \tag{4b}$$

$$\dot{\mathbf{q}}_{WI} = \frac{1}{2}\mathbf{q}_{WI} \otimes [0; (\boldsymbol{\omega}_m - \boldsymbol{\omega}_b - \boldsymbol{\omega}_n)] \tag{4c}$$

$$\dot{\mathbf{a}}_b = \mathbf{a}_{bn} \tag{4d}$$

$$\dot{\boldsymbol{\omega}}_b = \boldsymbol{\omega}_{bn} \tag{4e}$$

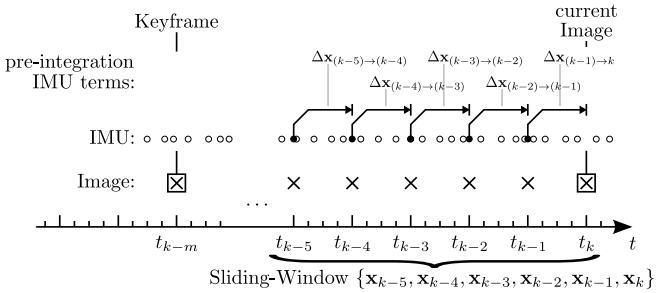

Fig. 4. Overview of the IMU pre-integration. An IMU state is associated to each image, so that the visual-based residuals can be related at these timestamps to the dynamic state of the IMU frame I. The IMU measurements are pre-integrated between consecutive image timestamps, resulting in the IMU pre-integration terms $\{\Delta\mathbf{x}_{(k-1)\rightarrow k}; \dots\}$.

A. Pre-integration of IMU measurements

The IMU pre-integration, introduced by [14], allows to predict the IMU state and estimation covariance between two timestamps, t_k and t_{k+1}, from a varying starting point $\mathbf{x}_k$. We use this technique in order to accelerate the calculation of the IMU residuals. The application of the pre-integration can be summarized as:

$$\mathbf{x}_{k+1|k} = \int_{t_k}^{t_{k+1}} \frac{d\mathbf{x}}{dt}(\mathbf{x}(t), \boldsymbol{\omega}_m, \mathbf{a}_m)dt \approx \mathbf{x}_k \oplus \Delta\mathbf{x}_{k\rightarrow(k+1)} \quad (5)$$

where the integrand is specified in Eq. (4) and the pre-integration terms are summarized as $\Delta\mathbf{x}_{k\rightarrow(k+1)}$. The calculation of these terms requires a single time integral calculation and, when applied, it is equivalent to the direct-integration for variable pose and velocity at t_k, $\{\mathbf{p}_I, \mathbf{v}_I, \mathbf{q}_{WI}\}_k$, and a first order approximation for variable biases at t_k, $\{\mathbf{a}_b, \boldsymbol{\omega}_b\}_k$. Fig. 4 shows a depiction of the pre-integration process and its relationship to the sliding-window of IMU states.

The bias correction is based on the aggregated state transition matrix of the error-state kinematics associated to the dynamic system specified by Eqs. (4). The discretized state transition matrix, $\mathbf{\Phi}_\mathbf{x}(\mathbf{x}) \in \mathbb{R}^{18\times18}$, and the noise perturbation matrices, $\mathbf{Q} \in \mathbb{R}^{12\times12}$ and $\mathbf{G} \in \mathbb{R}^{18\times12}$, of the error-state kinematics are the following:

$$\mathbf{\Phi}_\mathbf{x} = \begin{bmatrix} \mathbf{I} & \mathbf{I}\Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{R}\,[\mathbf{a}]_\times \Delta t & -\mathbf{R}\Delta t & \mathbf{0} & \mathbf{I}\Delta t \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{R}\Delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (6a)$$

with $\mathbf{R} = \mathbf{R}_{WI} = \mathbf{R}\{\mathbf{q}_{WI}\}$ and $\mathbf{a} = (\mathbf{a}_m - \mathbf{a}_b)$

$$\mathbf{Q} = \text{diag}\left(\Delta t^2\,\boldsymbol{\Sigma}_{an}; \Delta t^2\,\boldsymbol{\Sigma}_{\omega n}; \Delta t\,\boldsymbol{\Sigma}_{abn}; \Delta t\,\boldsymbol{\Sigma}_{\omega bn}\right) \quad (6b)$$

$$\mathbf{G}^\intercal = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \quad (6c)$$

The IMU pre-integration terms are obtained from the direct time-integration over the interval $[t_k, t_{k+1}]$, see Fig. 4, of the IMU measurements using equations (4a-c) and from the aggregation of the error-state transition and covariance matrices, see Eqs. (6a-c). This integration process results in the iterative application of Eqs. (7a-e), and results on the calculation of the IMU pre-integration terms $\Delta\mathbf{x}_{k\rightarrow(k+1)} = \{[\mathbf{p}_{pi}; \mathbf{v}_{pi}; \mathbf{q}_{pi}; \mathbf{a}_{b\,pi}; \boldsymbol{\omega}_{b\,pi}],\ \mathbf{\Phi}_{pi},\ \boldsymbol{\Sigma}_{pi}\}$. The integration is started from a fictitious initial IMU state in the origin with no velocity, no gravity and zero covariance, i. e. $\mathbf{x}_{pi\,init} = [\mathbf{0}; \mathbf{0}; \mathbf{q}\{\mathbf{0}\}; \mathbf{a}_{bk}; \boldsymbol{\omega}_{bk}]$, $\mathbf{\Phi}_{pi} = \mathbf{I}_{18}$ $\boldsymbol{\Sigma}_{pi} = \mathbf{0}_{18}$:

$$\mathbf{p}_{pi\,(j+1)} = \mathbf{p}_{pi\,(j)} + \mathbf{v}_{pi\,(j)}\Delta t \quad (7a)$$

$$\mathbf{v}_{pi\,(j+1)} = \mathbf{v}_{pi\,(j)} + \mathbf{R}\{\mathbf{q}_{pi\,(j)}\}(\mathbf{a}_{m\,(j)} - \mathbf{a}_b)\Delta t \quad (7b)$$

$$\mathbf{q}_{pi\,(j+1)} = \frac{1}{2}\mathbf{q}_{pi\,(j)} \otimes \mathbf{q}\{(\boldsymbol{\omega}_{m\,(j)} - \boldsymbol{\omega}_b)\Delta t\} \quad (7c)$$

$$\mathbf{\Phi}_{pi\,(j+1)} = \mathbf{\Phi}_\mathbf{x}(\mathbf{x}_{pi\,(j)})\mathbf{\Phi}_{pi\,(j)} \quad (7d)$$

$$\boldsymbol{\Sigma}_{pi\,(j+1)} = \mathbf{\Phi}_\mathbf{x}(\mathbf{x}_{pi\,(j)})\boldsymbol{\Sigma}_{pi\,(j)}\mathbf{\Phi}_\mathbf{x}(\mathbf{x}_{pi\,(j)})^\intercal + \mathbf{G}\mathbf{Q}\mathbf{G}^\intercal \quad (7e)$$

Equations (6a,7a-e) are written using the Euler integration method for simplicity. In this work we use the Runge-Kutta method instead, aka RK4 method, for the integration of differential equations and for the calculation of the discretized state transition matrix.

The IMU pre-integration terms allow to predict the IMU state and its estimation covariance, $\{\mathbf{x}_k, \boldsymbol{\Sigma}_k\}$, from different starting conditions, i. e. IMU state $\mathbf{x}_k$ and covariance $\boldsymbol{\Sigma}_k$. This operation is specified in Eqs. (8a-e), and we summarize it as $\{\mathbf{x}_{k+1|k}, \boldsymbol{\Sigma}_{k+1|k}\} = \{\mathbf{x}_k, \boldsymbol{\Sigma}_k\} \oplus \Delta\mathbf{x}_{k\rightarrow(k+1)}$:

$$\mathbf{p}_{k+1|k} = \mathbf{p}_k + \mathbf{v}_k\Delta t + \mathbf{g}\frac{\Delta t^2}{2} + \mathbf{R}\{\mathbf{q}_{WIk}\}\left(\mathbf{p}_{pi} + \mathbf{\Phi}_{pi\,\{\mathbf{p},\mathbf{a}_b\}}(\mathbf{a}_{b\,k} - \mathbf{a}_{b\,pi}) + \mathbf{\Phi}_{pi\,\{\mathbf{p},\boldsymbol{\omega}_b\}}(\boldsymbol{\omega}_{b\,k} - \boldsymbol{\omega}_{b\,pi})\right) \quad (8a)$$

$$\mathbf{v}_{k+1|k} = \mathbf{v}_k + \mathbf{g}\Delta t + \mathbf{R}\{\mathbf{q}_{WIk}\}\left(\mathbf{v}_{pi} + \mathbf{\Phi}_{pi\,\{\mathbf{v},\mathbf{a}_b\}}(\mathbf{a}_{b\,k} - \mathbf{a}_{b\,pi}) + \mathbf{\Phi}_{pi\,\{\mathbf{v},\boldsymbol{\omega}_b\}}(\boldsymbol{\omega}_{b\,k} - \boldsymbol{\omega}_{b\,pi})\right) \quad (8b)$$

$$\mathbf{q}_{k+1|k} = \mathbf{q}_k \otimes \mathbf{q}\{\mathbf{\Phi}_{pi\,\{\boldsymbol{\theta},\boldsymbol{\omega}_b\}}(\boldsymbol{\omega}_{b\,k} - \boldsymbol{\omega}_{b\,pi})\} \otimes \mathbf{q}_{pi} \quad (8c)$$

$$\mathbf{a}_{b(k+1)|k} = \mathbf{a}_{bk}, \quad \boldsymbol{\omega}_{b(k+1)|k} = \boldsymbol{\omega}_{bk} \quad (8d)$$

$$\boldsymbol{\Sigma}_{k+1|k} = \mathbf{A}_{WIk}\left(\mathbf{\Phi}_{pi}\mathbf{A}_{WIk}^\intercal\boldsymbol{\Sigma}_k\mathbf{A}_{WIk}\mathbf{\Phi}_{pi}^\intercal + \boldsymbol{\Sigma}_{pi}\right)\mathbf{A}_{WIk}^\intercal$$
with $\mathbf{A}_{WIk} = \text{diag}\left(\mathbf{R}_{WIk}, \mathbf{R}_{WIk}, \mathbf{R}_{WIk}, \mathbf{I}, \mathbf{I}, \mathbf{R}_{WIk}\right)$ (8e)

Sliding-window length: as pointed out by the VIO initialization process devised by Lupton et al. [14], three poses in conjunction with the intervening IMU pre-integration terms can produce an estimation of the gravity vector. This justifies our motivation to use a sliding-window with at least 3 IMU states.

B. Inertial Residuals:

The IMU state prediction residual, $\mathbf{r}_{isp}$, introduces a cost between consecutive states comparing the current estimate of $\mathbf{x}_{k+1}$ to its prediction based on the first state $\mathbf{x}_k$ by means of the intervening IMU measurements:

$$\mathbf{r}_{isp\,(k)\rightarrow(k+1)}(\mathbf{x}_k, \mathbf{x}_{k+1}, \mathbf{g}) = \mathbf{x}_{k+1|k} \ominus \mathbf{x}_{k+1} \quad (9a)$$

$$\text{with } \{\mathbf{x}_{k+1|k}, \boldsymbol{\Sigma}_{k+1|k}\} = \{\mathbf{x}_k, \boldsymbol{\Sigma}_k\} \oplus \Delta\mathbf{x}_{k\rightarrow(k+1)} \quad (9b)$$

C. Prior Residuals:

The IMU state prior residual, $r_{s\,prior}$, defines a cost on unnecessary variations of the first state of the sliding-window, $\mathbf{x}_{k-5}$, compared to its value after the last batch

optimization execution, $\mathbf{x}_{(k-5)\,prior}$:

$$\mathbf{r}_{s\,prior}\left(\mathbf{x}_{k-5}\right) = \mathbf{x}_{(k-5)\,prior} \ominus \mathbf{x}_{k-5} \tag{10}$$

The keyframe position, the inter-sensor spatial calibration and the gravity prior residuals are defined in a similar manner as a function of their value in the last batch optimization execution, $\mathbf{T}_{WC_{L\_KF}\,prior}$, $\mathbf{T}_{IC_L\,prior}$ and $\mathbf{g}_{prior}$:

$$\mathbf{r}_{\mathbf{T}_{prior}}\left(\mathbf{T}\right) = \mathbf{T}_{prior} \ominus \mathbf{T} \tag{11a}$$

$$\mathbf{r}_{g\,prior}\left(\mathbf{g}\right) = \mathbf{g}_{prior} \ominus \mathbf{g} = \mathbf{g}_{prior} - \mathbf{g} \tag{11b}$$

D. Vision-Only Relative Pose Measurement Residuals

In addition to the IMU states, the sliding-window stores vision-only based pose measurements along with their covariances, $\left\{\left\{\mathbf{T}_{WC_L m\,(i)}, \mathbf{\Sigma}_{\mathbf{T}pm\,(i)}\right\}\right\}_{i=(k-5),\ldots,(k-1)}$, see Sect. V-E. These measurements are better understood as relative poses with respect to nearby pose measurements. For this reason they are utilized in pairs resulting in the relative pose measurement residual $r_{rpm\,(k)\to(k+1)}$:

$$\mathbf{r}_{rpm\,(k)\to(k+1)}\left(\mathbf{T}_{IC_L},\, \mathbf{T}_{WI\,(k)},\, \mathbf{T}_{WI\,(k+1)}\right) = \tag{12}$$
$$\mathbf{T}_{WC_L m(k+1)} \ominus \left(\mathbf{T}_{WC_L m\,(k)} \mathbf{T}_{IC_L}^{-1} \mathbf{T}_{WI(k)}^{-1} \mathbf{T}_{WI(k+1)} \mathbf{T}_{IC_L}\right)$$

E. Visual-Inertial Cost Function and Batch Optimization

Our VINS algorithm relies on a cost function combining the image and inertial residuals discussed previously. In order to obtain an optimal estimate, we seek to ground the derivation of this cost on a strong statistical background, i. e. the application of the MAP method.

The proposed cost function, Eq. 13a, can be interpreted as a weighted Sum of Squared Errors (SSE), which are calculated from the sum of the squared norms of the residuals. However, for the SSE to be consistent with a solid statistical interpretation based on the ML principle, the measurement-based residuals have to be properly weighed against their intrinsic measurement uncertainties. For this reason, our SSE is defined as the sum of the squared Mahalanobis norms of the residuals, i. e. $\|\mathbf{r}\|^2_{\mathbf{\Sigma}} = \mathbf{r}^\intercal \mathbf{\Sigma}^{-1} \mathbf{r}$.

In addition, to the weighted image, inertial and relative pose measurement residuals, $E_{patch\,VO}$, r_{isp} and r_{rpm} respectively; several priors are added for part of the parameter estimates. These priors provide information about our current best estimates for the inter-sensor spatial calibration, the current keyframe pose, the gravity and the first IMU state. These prior residuals are respectively, $\mathbf{r}_{\mathbf{T}_{prior}}\left(\mathbf{T}_{IC_L}\right)$, $\mathbf{r}_{\mathbf{T}_{prior}}\left(\mathbf{T}_{WC_{L\_KF}}\right)$ $\mathbf{r}_{g\,prior}$, and $\mathbf{r}_{s\,prior}$. The addition of these properly weighed priors leads to the MAP based cost function proposed in Eq. 13a. A depiction of the involved priors and parameters, and which residual cost evaluation they affect is shown graphically on the diagram of Fig. 5.

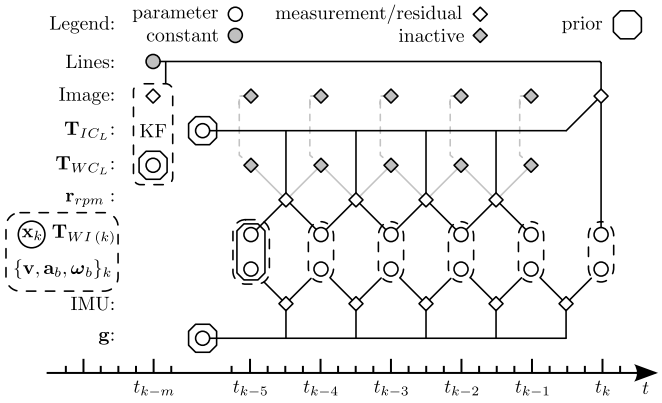

Fig. 5. Diagram showing the parameters that contribute to the E_{VIO} cost, Eq. 13a. The residuals are shown with diamonds, and their evaluation requires the parameters, shown with circles, which are drawn connected to them with black solid lines. Constant parameters and inactive residuals are shown greyed out. Grey dashed lines show potential data correlations. I. e. the image residuals used to derive a pose measurement are highly correlated with it, thus they are not both utilized at the same time. Grey solid lines show the down-grading of pose measurements to relative pose measurements. The prior residuals are shown with octagons around the effected parameters.

$$\begin{aligned}
&E_{VIO}\left(\mathbf{T}_{WC_{L_KF}}, \mathbf{T}_{IC_L}, \{\mathbf{x}_i\}_{i=(k-5),\ldots,k}, \mathbf{g}\right) = \qquad (13a)\\
&+ \left\|\mathbf{r}_{\mathbf{T}_{prior}}\left(\mathbf{T}_{WC_{L_KF}}\right)\right\|^2_{\mathbf{\Sigma}_{\mathbf{T}_{WC_{L_KF}}}} + \left\|\mathbf{r}_{\mathbf{T}_{prior}}\left(\mathbf{T}_{IC_L}\right)\right\|^2_{\mathbf{\Sigma}_{\mathbf{T}_{IC_L}}}\\
&+ \sum_{i=1}^{n_{patches}} \left(\frac{\sigma_{VO}}{\sigma_{VIO}}\right)^2 E_{patch\,VO}\left(\mathbf{T}_{WC_{L_KF}}^{-1} \mathbf{T}_{WI\,(k)} \mathbf{T}_{IC_L}\right)\\
&+ \sum_{i=(k-5)}^{(k-2)} \left\|\mathbf{r}_{rpm\,(i)\to(i+1)}\left(\mathbf{T}_{IC_L}, \mathbf{T}_{WI\,(i)}, \mathbf{T}_{WI\,(i+1)}\right)\right\|^2_{\mathbf{\Sigma}_{\mathbf{T}pm\,(k)}}\\
&+ \left\|\mathbf{r}_{g\,prior}\left(\mathbf{g}\right)\right\|^2_{\mathbf{\Sigma}_{\mathbf{g}}} + \left\|\mathbf{r}_{s\,prior}\left(\mathbf{x}_{k-5}\right)\right\|^2_{\mathbf{\Sigma}_{k-5}}\\
&+ \sum_{i=(k-5)}^{(k-1)} \left\|\mathbf{r}_{isp\,(i)\to(i+1)}\left(\mathbf{x}_{(i)}, \mathbf{x}_{(i+1)}, \mathbf{g}\right)\right\|^2_{\mathbf{\Sigma}_{k+1|k}}
\end{aligned} \tag{13b}$$

The calculation of the independent pose measurements is performed based on the optimization of the cost in Eq. 14b, whose residuals are defined and correspond to the same image patches which are used in the VIO cost. Since these vision residuals are never used again directly on the VIO cost, the resulting pose measurement is relatively uncorrelated from the other measurements contributing to the E_{VIO} cost in later iterations. The advantage of the pose measurements, is that compared to the addition of additional images to the cost function, they allow for a faster computation of the cost. About the new σ parameters, it is noted that while the σ_{VO} parameter sets the strength of the visual outlier rejection, σ_{VIO} serves as calibration parameter to set the strength of the visual residuals.

$$E_{patchVO}\left(\mathbf{T}_{C_{L\_KF}C_L}\right) = \tag{14a}$$
$$\left\| \sqrt{\sum_{i=1}^{n_{pixels}} r_i(\mathbf{T}_{C_{L\_KF}C_L})^2} \right\|_{\boldsymbol{\sigma}_{VO}^2 n_{pixels}}$$

$$E_{VO}\left(\mathbf{T}_{WC_{L\_KF}}, \mathbf{T}_{IC_L}, \mathbf{T}_{WI\,(k)}\right) = \tag{14b}$$
$$\sum_{i=1}^{n_{patches}} \left(\frac{\sigma_{VO}}{\sigma_{VIO}}\right)^2 E_{patch\,VO}\left(\mathbf{T}_{WC_{L\_KF}}^{-1}\mathbf{T}_{WI\,(k)}\mathbf{T}_{IC_L}\right)$$

F. Workflow of the Algorithm

The initialization of the filter is performed following the method proposed by Lupton [14] utilizing the first three vision-only relative pose measurements. Otherwise upon reception of a new image, the steps performed by our VINS are the following:

1) Pre-integration of the IMU and initialization of the new IMU pose to the resulting estimate at $\mathbf{x}_k$,
2) Minimization of the E_{VIO} cost, which results on the initialization for the VO-only optimization,
3) Minimization of the E_{VO} cost, and calculation of the new vision-only based pose measurement,
4) Update the sliding-window with the new parameter and covariance estimates,
5) Decide whether the current image is selected as new keyframe, and if so, the scene structure is updated.

The IMU state and the position measurement estimation covariance matrices, $\{\{\boldsymbol{\Sigma}_i\}_{i=(k-5),\ldots,k}, \boldsymbol{\Sigma}_{\mathbf{g}}, \boldsymbol{\Sigma}_{\mathbf{T}_{IC_L}}\}$ and $\{\boldsymbol{\Sigma}_{\mathbf{T}pm\,i}\}_{i=(k-5),\ldots,(k-1)}$, are obtained, respectively, after the VIO and the VO batch optimizations have converged. They result from a direct application of the *backward transport of covariance* theorem. When an image is selected as new keyframe, then its associated pose measurement estimation covariance is also utilized for the keyframe pose, i. e. $\boldsymbol{\Sigma}_{\mathbf{T}_{WC_{L\_KF}}} = \boldsymbol{\Sigma}_{\mathbf{T}pm\,k}$. The implementation of our VINS rely on the ceres-solver [20] for the realization of the non-linear optimization and, afterwards, for the calculation of the optimization parameter covariances.

VI. EXPERIMENTAL RESULTS

In our experimental evaluation, we used the publicly available Rawseeds dataset [21] and datasets acquired with our own camera-IMU setup. We compared the results of our VIO algorithm (referred as VIO) with a state-of-the-art stereo VIO algorithm (OKVIS [1]) and with our own algorithm using the vision measurements only (referred as VO-only). We show that we can significantly improve the results compared to VO-only and deliver comparable results to OKVIS. We have selected OKVIS for the experimental evaluation of our VINS, because it is a state-of-the-art visual-inertial stereo approach which has common characteristics with our design.

A. Rawseeds

The Rawseeds datasets are publicly available indoor and outdoor datasets acquired with a ground floor robot which is equipped with multiple sensors (like multiple cameras, IMU, laser scanner). For the indoor datasets, the robot was also

Fig. 6. Images from the Rawseeds Bicocca_2009-02-27a dataset. Most of the images contain narrow corridors and broader hallways in between. However, also images just containing untextured walls are included in this sequence.

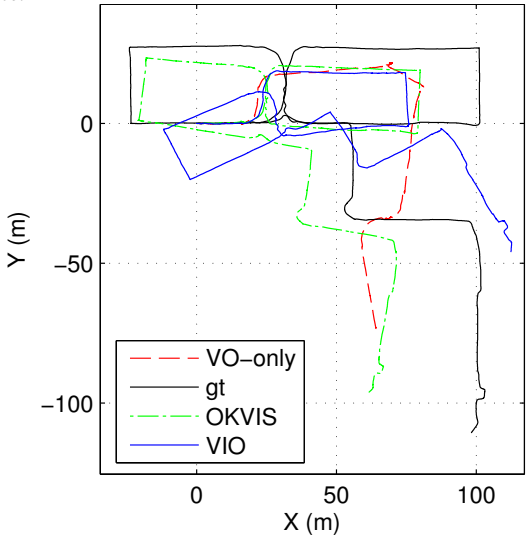

Fig. 7. Trajectory estimations of the Rawseeds Bicocca_2009-02-27a dataset. As can be seen, VO-only (red) fails in the corners of the corridor, where nearly no texture exist. Therefore it was also not plotted for the whole sequence in this plot. Contrary, VIO (blue) performs much better in these areas. Compared to OKVIS, VIO has a slightly higher rotational error.

partly tracked by an external tracking system. To get a ground truth for the full trajectories, an extended ground truth was made available which also includes poses computed from data acquired by the onboard laser scanner.

In our experiments, we only used one subset, namely Bicocca_2009-02-27a (see Fig 6). We used the left and right camera from the forward-looking trinocular camera and the IMU measurements. The camera images have a resolution of 640x480 pixels and were captured with a framerate of 15 FPS. The IMU is an Xsens MTi and acquired measurements with a rate of 128 Hz. In our VINS, we used the noise characteristics and bias definitions mentioned in the manual of the IMU.

In Fig. 7 one can see the trajectories of the algorithms which are compared. It can be observed, that VIO results in a better tajectory than VO-only, as motion in the poorly textured corridor corners can be estimated better with the help of the IMU.

As there is also ground truth available for this trajectory, we can compute the relative pose errors, which are discussed in Tab. I. As can be observed, especially the rotational errors are much lower for VIO compared to VO-only. This is due to nearly textureless parts of the sequence, where the robot is rotating in front of untextured walls. Compared to OKVIS,

TABLE I

RELATIVE TRANSLATIONAL AND ROTATIONAL ERROR OF THE RAWSEEDS SEQUENCE AS DEFINED IN [22] PER SECONDS OF MOVEMENT OF OUR APPROACHES (VO-ONLY AND VIO) AND OKVIS.

	VO-only	VIO	OKVIS
Transl. RMSE (m/s)	0.5216	0.1988	0.1499
Transl. Median (m/s)	0.3835	0.1637	0.1305
Rot. RMSE (deg/s)	5.4824	1.9943	1.9444
Rot. Median (deg/s)	0.0135	0.0127	0.0084

our approach has a similar rotational and translational error. However, OKVIS optimizes multiple keyframes and their corresponding 3D points in a local bundle adjustment step, which leads to a slightly reduced rotational and translational drift.

B. Own Dataset

Our own dataset was acquired with a stereo camera mounted on a micro aerial vehicle (MAV) (more precisely, the Asctec Pelican). The MAV was moved along a circular trajectory within a poorly textured indoor environment. In Fig. 8, one can see some images from the captured sequence. This trajectory was selected because it should demonstrate the benefits of VIO for a fast moving camera, where direct methods (like our VO-only approach) might have problems.

The stereo camera consists of two Matrix Vision BlueFox-MLC202b cameras with a baseline of approximately $13.5\ cm$. With our camera lens setup, each camera has a horizontal field of view of $81.2\ deg$, captured with a framerate of $20\ FPS$ and a resolution of $640x480$ pixels. To synchronize the two cameras, they are triggered by the Asctec Autopilot, which also simultaneously delivers IMU measurements. The IMU data rate was set to $80\ Hz$.

The system was calibrated in three steps. First, the stereo camera intrinsic and extrinsic, $\mathbf{T}_{C_L C_R}$, parameters were calibrated. Second, a long-term stand-still acquisition of IMU measurements was acquired. This dataset was used to obtain a stochastic characterization of the IMU bias drift and measurement noises, based on the obtained experimental estimates of the Power Spectral Density (PSD) and the Allan Variance (AV) of the noises, as specified by the relevant IEEE standards. Third, both of these calibrations were utilized to estimate the inter-sensor spatial calibration, i. e. the reference transform $\mathbf{T}_{WC_L}$, using the Kalibr toolbox [23]. The data acquisition in our VINS is synchronized, so that the IMU measurements and image timestamps can be compared without assuming any error in the calculations.

In Fig. 9 one can see the results of our VIO algorithm in comparison to our VO-only algorithm and OKVIS. It can be observed that VO-only has a drift in height, while VIO can significantly reduce this due to the inclusion of IMU measurements. Compared to OKVIS, VIO performs comparable. However, as we do not have a ground truth, we cannot quantitatively compare the approaches against each other.

Fig. 8. Example images from our own test sequence. The acquisition platform was moved in circular movements in an indoor environment.

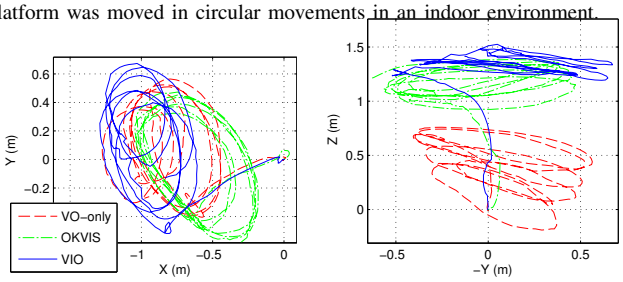

Fig. 9. Results of our own trajectory. *Left:* Top view, *Right:* Side view. As can be seen, VIO significantly reduces height drift compared to VO-only Compared to OKVIS, VIO has a slightly higher horizontal drift and a comparable height drift.

VII. CONCLUSION

In this paper, we have shown that the results of our direct visual odometry algorithm are significantly improved by tightly coupling it with IMU measurements. Our experiments have shown that our approach achieves comparable accuracy to a state-of-the-art method, thus, demonstrating the potential of the usage of the independent pose measurements residuals as sources of information. We have based the derivation of our cost function on the application of a sound statistical methods, namely MAP, which we have used to support the inclusion of these residuals. Future work will include extending the visual odometry algorithm to jointly optimize multiple keyframes and, simultaneously, 3D lines. This could significantly reduce drift in the combined VINS.

REFERENCES

[1] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, 2015.

[2] A. Mourikis, N. Trawny, S. Roumeliotis, A. E. Johnson, A. Ansar, L. Matthies, *et al.*, "Vision-aided inertial navigation for spacecraft entry, descent, and landing," *Robotics, IEEE Transactions on*, 2009.

[3] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*, 2014.

[4] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Proceedings International Conference on Computer Vision*, 2011, pp. 2320–2327.

[5] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *International Conference on Robotics and Automation*, 2014.

[6] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Proceedings International Symposium on Mixed and Augmented Reality*, 2007.

[7] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, 2012.

[8] A. Elqursh and A. M. Elgammal, "Line-based relative pose estimation." in *Proceedings IEEE Conference Computer Vision and Pattern Recognition*. IEEE Computer Society, 2011, pp. 3049–3056.

[9] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," in *Robotics research*. Springer, 2010, pp. 201–212.

[10] S. Weiss, M. W. Achtelik, M. Chli, and R. Siegwart, "Versatile distributed pose estimation and sensor self-calibration for an autonomous mav," in *International Conference on Robotics and Automation*, 2012.

[11] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *The International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[12] F. M. Mirzaei and S. I. Roumeliotis, "A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation," *Robotics, IEEE Transactions on*, 2008.

[13] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 298–304.

[14] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 61–76, 2012.

[15] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Robotics: Science and Systems XI*, 2015.

[16] T. Holzmann, F. Fraundorfer, and H. Bischof, "Direct stereo visual odometry based on lines," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2016.

[17] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in *Proceedings International Conference on Computer Vision*, 2013.

[18] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *International Conference on Robotics and Automation*, 2013.

[19] J. Sola, "Quaternion kinematics for the error-state kf," *LAAS-CNRS, Toulouse, France, Tech. Rep*, 2012.

[20] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[21] A. Bonarini, W. Burgard, G. Fontana, M. Matteucci, D. G. Sorrenti, and J. D. Tardos, "Rawseeds: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets," in *International Conference on Intelligent Robots and Systems*, 2006.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *International Conference on Intelligent Robots and Systems*, 2012.

[23] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013.