An Open Source, Fiducial Based, Visual-Inertial State Estimation System

Michael Neunert, Michael Blösch and Jonas Buchli

***Abstract*— Many robotic tasks rely on the estimation of the location of moving bodies with respect to the robotic workspace. This information about the robots pose and velocities is usually either directly used for localization and control or utilized for verification. Often motion capture systems are used to obtain such a state estimation. However, these systems are very costly and limited in terms of workspace size and outdoor usage. Therefore, we propose a lightweight and easy to use, visual inertial Simultaneous Localization and Mapping approach that leverages paper printable artificial landmarks, so called fiducials. Results show that by fusing visual and inertial data, the system provides accurate estimates and is robust against fast motions. Continuous estimation of the fiducials within the workspace ensures accuracy and avoids additional calibration. By providing an open source implementation and various datasets including ground truth information, we enable other community members to run, test, modify and extend the system using datasets or their own robotic setups.**

I. Introduction

A. Motivation

For many tasks in mobile robotics, it is important to estimate a robot's state with respect to its workspace, i.e. its pose and velocities expressed in an inertial coordinate frame aligned to the robot's workspace. Such tasks include navigation, motion planning or manipulation.

One way to measure the position and orientation of a robot is to use a motion capture system (such as e.g. Vicon[1], Optitrack[2] or PTI Visualeyez[3]). These systems are usually highly accurate and provide a state estimate in a fixed calibrated frame. However, these systems can be very costly and the user might be limited to a certain workspace size. Furthermore, many common systems such as Vicon and Optitrack use passive markers that reflect infrared light which limits there usability in outdoor scenarios. Also, most systems require a tedious calibration procedure that needs to be repeated frequently to maintain accuracy. Since these systems do not have access to inertial data, they have to apply finite differences on position estimates to compute velocity information, which usually leads to highly quantized data (compare Figure 12).

Another approach to state estimation is Visual Odometry (VO), which is sometimes also combined with inertial data. VO can provide very accurate local estimates of the robot motion. However, usually only a finite number of previous observations (frames) are included during the pose estimation step and no loop closure is performed. While this can lead to a leaner, computationally less demanding implementation, VO can drift over time and does not provide a globally consistent path. Compared to Visual Odometry Simultaneous Localization and Mapping (SLAM) introduces the notion of a global map and therefore can ensure consistency by performing loop closure. Since the set of observed features in a scene can vary, SLAM often relies on a large number of features and landmarks. Therefore, map building, storing and loop-closure detection can be computationally and memory demanding. Furthermore, the detection and re-observation of the features can be erroneous or fail and thus impair the robustness and accuracy of the overall system.

In this work, we propose a monocular, visual-inertial SLAM system that only relies on inertial measurements and artificial visual landmarks, also known as "fiducials" which constitute the map. By using artificial landmarks that provide rich information, the estimation, mapping and loop closure effort is minimized. In this implementation, we use AprilTags [1] as our fiducials. Since these tags also provide a unique identification number, they can be robustly tracked and estimated in the applied Extended Kalman Filter (EKF). Therefore, loop closure is handled implicitly and no additional loop closure detection step is required. Furthermore, a single observation of a tag is sufficient to compute the relative transformation between tag and robot. Hence, the system can work with very few tags while still remaining accurate. As a result the map size and also the filter state size remains low which lowers computational demands. Therefore, the complexity of the proposed system is much lower than common SLAM approaches while still providing accurate, globally consistent estimates. By also including inertial measurements, robust performance during fiducial occlusion or motion blur from fast motions is ensured. While inserting the paper-based landmarks in the workspace of a robot is simple, the proposed approach is not applicable to all robotic tasks. Hence, instead of a replacement for SLAM solutions, we see the developed system as an additional, lightweight tool that can be used e.g. for verification of other state estimation systems, as an inexpensive outdoor-capable motion capture system or for absolute localization in a given workspace.

Ideally, such a tool should be inexpensive and easy to set up. Therefore, the proposed approach relies only on sensors that are already available on most robotic platforms, i.e. an inertial measurement unit (IMU) and a single camera. The

Michael Neunert and Jonas Buchli are with the Agile & Dexterous Robotics Lab. Michael Blösch is with the Autonomous Systems Lab. Both labs are at the Institute of Robotics and Intelligent Systems, ETH Zurich, Switzerland. {neunertm, bloeschm, buchlij}@ethz.ch

[1] http://www.vicon.com

[2] https://www.naturalpoint.com/optitrack/

[3] http://www.ptiphoenix.com/

fiducials are paper-based and hence inexpensive to produce. Since the fiducials positions and orientations do not have to be known a priory but get estimated online, the tags can be placed in arbitrary orientations and locations in the workspace and previous calibration is not required. To facilitate an easy integration, we are providing the implementation as an open source software package seamlessly integrated into Robot Operating System (ROS)[2]. Additionally, we provide data sets including ground truth measurements for verification of the system and possible improvements to it.

Therefore, we hope to provide the community with an easy to use, inexpensive and lightweight, yet accurate state estimation or motion capture system that frees the user from possible indoor limitations, tedious calibration processes or purchasing expensive hardware.

B. Related Work

The proposed approach combines fiducial-based localization and feature-based, monocular, visual-inertial SLAM. If considered separately, both topics are studied extensively and therefore are well covered in literature.

Many existing fiducial-based localization systems are targeted for augmented reality or were designed to be used with cameras only. Hence, many systems use vision data only (e.g. [3], [4], [1], [5], [6]). These systems have two major drawbacks over the presented system. Firstly, they fail to provide any estimate during occlusion or motion blur. Secondly, linear velocities and body rates can only be computed based on the position and orientation estimates and are thus highly quantized. While this might be negligible for virtual reality applications, it can cause issues when closing a control loop over these estimates. To mitigate these issues, a motion model can be assumed [7]. However, this makes the approach specific to the system dynamics implemented in the motion model. While these vision-only systems fail during motion blur or occlusion, our approach relies on the fiducials and the according detectors developed in these approaches.

The motion estimation and map building elements of the presented approach are closely related to monocular, visual-inertial SLAM, which has proven to be very effective [8], [9], [10], [11], [12]. The difference between our approach and fiducial-free visual-inertial SLAM solutions is that we rely on artificial landmarks that result in highly robust and unique features in image space. As a result, our landmarks can be easily redetected and their detections are virtually outlier free which increases the robustness of the approach. Additionally, each landmark has a 6-DoF pose (position and orientation) rather than the usual 3-DoF point landmarks used in most SLAM approaches. Therefore, a single landmark is sufficient for estimating the pose. This also allows for a simple yet accurate landmark initialization which usually is a problem in monocular SLAM approaches [13]. Furthermore, fiducials allow to align the map and the estimates to a certain location in the workspace. While SLAM also provides an estimate and a map of the workspace, they are not aligned to a given reference in the workspace. Therefore, when e.g. performing manipulation task separate alignment or object recognition is required.

While both, fiducial-based localization and SLAM are well studied problems, not many approaches exist that combine both. One approach where fiducials are combined with SLAM is presented in [14]. However, the inertial measurements are not used to estimate velocities but only used for a fall-back pose estimation if all fiducials are occluded. Thus, this work does not fully leverage the potential of the inertial measurements. Another similar system as the one presented in this work has been described in [15]. Since this work is part of the development of a commercial product (InterSense IS-1200[4]) the authors remain relatively vague about their sensor fusion algorithm as well as the achievable performance of their system. Furthermore, dedicated hardware is required which poses additional costs for the user and contradicts the goals of this project to provide a cost-efficient, open source framework. A third visual-inertial, fiducial based localization system is presented in [16]. While also here inertial measurements and visual data are fused in an EKF, the approach does not include measurements from an accelerometer which can be helpful during fast linear motions and can provide a notion of gravity. Additionally, it is assumed that the poses of the tags are perfectly known in a world frame. Therefore, one can only place the tags in known configuration, e.g. in a single plane or one has to pre-calibrate the setup using a different approach. Also, imperfect calibration will lead to offsets or non-smooth estimates when transitioning between tags.

C. Contributions

In this paper, we present a monocular visual-inertial EKF-SLAM system based on artificial landmarks. This work successfully combines two proven concepts, namely SLAM and fiducial-based state estimation by extending the common 3D point landmark formulation of SLAM to 6 DoF landmarks. The approach provides estimates to possibly known locations in the workspace and thus can be used as a localization system. Yet, in contrast to existing combined approaches, the presented approach fuses visual measurements, landmark poses and inertial data in a single estimator. Therefore, the approach does not require pre-calibration of the workspace.

Furthermore, we provide the system as free to use open source software including data sets for verification and testing. Through integration in ROS the system can be easily interfaced with common hardware. Special care is taken to provide a lean implementation with low computationally requirements and little dependencies to make the system suitable for low power platforms as well. The source code, the datasets as well as a more detailed technical manual can be found at `https://bitbucket.org/adrlab/rcars`.

D. Notation and Conventions

In the following sections, scalars are indicated with small letters (e.g. f_x). Vectors are indicated with small, bold letters

[4]http://www.intersense.com/categories/2/

(e.g. $\boldsymbol{r}$). Matrices are indicated by non-bold capital letters (e.g. K).

A capital subscript leading a variable name describes the frame that the position vector or rotation quaternion is expressed in.

Position vectors are usually denoted by r. The trailing subscript describes the direction of the vector from its origin to its goal position (read from left to right), e.g. ${}_A r_{OP}$ is a position vector expressed in frame A that points from point O to point P.

Quaternions are usually denoted by q. The trailing subscript defines the (passive) rotation of the latter system around the former one, e.g. q_{AB} defines the rotation of the coordinate system B around the coordinate system A. Hence, to rotate a position vector expressed in B to A, we would compute ${}_A r_{OP} = q_{AB}({}_B r_{OP})$.

II. System Description

The present localization system consists of two main elements, a detector for the fiducials and an extended Kalman filter (EKF) that continuously estimates the robot's state as well as the fiducials' poses.

In a first step, the image acquired by the camera is rectified and if required converted to gray scale. Afterwards, the detector is run on the image which outputs the corner coordinates as well as a unique identifier number (id) associated with each detected tag. Furthermore, it estimates the relative transformation between each tag and the camera. This estimation is based on an iterative optimization minimizing the reprojection errors between the projected 3D corner points and its detections in image space.

In a second step, the EKF uses the information from redetected tag corners to estimate the robot's state, including pose, linear velocity and body rates. Additionally, the filter continuously estimates the position and orientation of the tags with respect to a gravity aligned inertial frame. When a tag is seen for the first time, its pose is initialized using the current pose estimate for the camera and the optimized, relative transformation between the camera and the tag as provided by the detector. After this initialization, the tag pose will be refined within the EKF by using the reprojection errors of its corners in each subsequent re-observation.

A. Fiducials

Over the past years, a large variety of fiducial systems have been developed. Very popular implementations include ARToolKit [17], ARTag [4], CyberCode [18] and multiring color fiducials [19]. In our implementation, we are using AprilTags [1] which are 2-dimensional, printable bar codes. The reason for this choice was the achievable high accuracy [1] and the number of available detector implementations in C/C++. In our system, we are using the detector implemented in cv2cg[5]. In our evaluations, this implementation has proven to be relatively fast, while still providing accurate and robust tag detections.

[5]http://code.google.com/p/cv2cg/

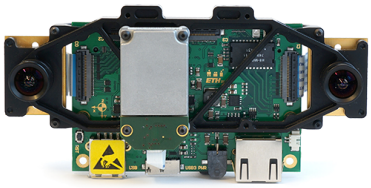

Fig. 1. Skybotix VI-Sensor used for development and verification. The sensor provides time stamped stereo camera and IMU data. In this experiment, the stereo capabilities are neglected and only the left camera is used.

B. Hardware

The proposed system requires a camera and an inertial measurement unit (IMU). The calibration of the sensor rig, i.e. the intrinsics of the camera as well as the transformation between camera and IMU have to be known.

In our setup, we are using a Skybotix[6] VI-Sensor[20] shown in Figure 1. This sensor consists of two cameras in a stereo configuration and an IMU. While the sensor can be used as a stereo camera we are only relying on the left camera in this work. The sensor is set up to output images at 20 Hz and IMU data at 200 Hz.

The VI-Sensor comes factory calibrated, both for the camera intrinsics as well as the extrinsic calibration between each camera and the IMU. Thus, in our case no additional calibration needs to be done.

C. Camera Model

In this project, we assume a pinhole camera model. The pinhole camera model is frequently used and thus should allow for an easy integration with existing camera hardware. The pinhole camera model is based on a 3x3 projection matrix K that transforms a 3D point in camera coordinates described by the position vector ${}_C\boldsymbol{r}_{OC}$ to its projection point ${}_P\boldsymbol{r}_{OC}$

$$ {}_P\boldsymbol{r}_{OC} = K\ {}_C\boldsymbol{r}_{OC} \tag{1} $$

$$ = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}\ {}_C\boldsymbol{r}_{OC} \tag{2} $$

where f_x and f_y denote the focal lengths and $c = (c_x, c_y)$ denotes the camera's principle point. To obtain the pixel coordinates of the projection point ${}_P\boldsymbol{r}_{OC}$ in the image plane (z = 1) a second projection is applied, that normalizes the projection point with its z-coordinate. We refer to the overall projection consisting of the camera projection 2 and the subsequent projection onto the image plane as the camera projection map π.

The expected input for the detector and filter is a rectified image, i.e. an undistorted image. Therefore, the user is free to choose a distortion model as long as an undistorted image is provided. In the case of the VI-Sensor we are using a radial tangential distortion model based on the factory calibration.

[6]http://www.skybotix.com

D. Filter

In order to fuse the information gained from the observed tags together with the on-board inertial measurement we implement an extended Kalman filter. Relying on appropriate sensor models, this filter uses the inertial measurements in order to propagate the robot's state and performs an update step based on the available tag measurements. We would like to mention that, although we only make use of raw IMU measurements (proper acceleration and rotational rates) and do not have access to absolute attitude measurements, the proposed system is able to estimate the absolute inclination angles of the sensor with respect to gravity.

In the following paragraphs we will explain the sensor models used and derive the required filter equations. For readability, this derivation is carried out for the case of a single tag, but can be easily extended to the case of multiple tags.

1) Coordinate Systems: In our filter setup we assume different coordinate frames. First, we assume an inertial world coordinate system W. We assume that gravity points in negative z-direction in this frame. Furthermore, we define the body coordinate system B. For the sake of readability, we assume our sensor measurements to be expressed in this frame. In practice two additional frames, the camera and the IMU frame are introduced, which are linked to the body frame over a constant transformation (extrinsic calibration). Finally, we define a coordinate system T for each tag which coincides with the geometrical center of the tag.

2) Sensor Models: First, we introduce the sensor model used for the IMU. It assumes Gaussian noise as well as additive bias terms for accelerometer and gyroscope measurements. This can be formulated as follows:

$$\tilde{\boldsymbol{f}} = \boldsymbol{f} + \boldsymbol{b}_f + \boldsymbol{w}_f, \tag{3}$$

$$\dot{\boldsymbol{b}}_f = \boldsymbol{w}_{bf}, \tag{4}$$

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{b}_\omega + \boldsymbol{w}_\omega, \tag{5}$$

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega}, \tag{6}$$

where $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{\omega}}$ are the actual measurements of the proper acceleration and rotational rates, $\boldsymbol{b}_f$ and $\boldsymbol{b}_\omega$ are the additive bias terms, and all terms of the form $\boldsymbol{w}_*$ represent continuous white Gaussian noise.

Amongst the IMU data, we will also include measurements of the observed tags. For this measurement we propose the use of a re-projection based visual model. Given the position and attitude of the camera, $_{\boldsymbol{W}}\boldsymbol{r}_{\boldsymbol{WB}}$ and $\boldsymbol{q}_{\boldsymbol{BW}}$, as well as the same quantities of a specific tag, $_{\boldsymbol{W}}\boldsymbol{r}_{\boldsymbol{WT}}$ and $\boldsymbol{q}_{\boldsymbol{TW}}$, we can compute the position of the i^{th} tag corner $_{\boldsymbol{T}}\boldsymbol{r}_{\boldsymbol{TC}_i}$ (fixed to the target coordinate frame $\boldsymbol{T}$) as viewed from the camera:

$$_{\boldsymbol{B}}\boldsymbol{r}_{\boldsymbol{BC}_i} = \boldsymbol{q}_{\boldsymbol{BW}}\left(_{\boldsymbol{W}}\boldsymbol{r}_{\boldsymbol{WB}} + _{\boldsymbol{W}}\boldsymbol{r}_{\boldsymbol{WT}} + \boldsymbol{q}_{\boldsymbol{TW}}^{-1}(_{\boldsymbol{T}}\boldsymbol{r}_{\boldsymbol{TC}_i})\right). \tag{7}$$

By using the camera projection map π, we can project the above quantity onto the image plane and derive the corresponding pixel coordinate measurement $\tilde{\boldsymbol{p}}_i$, where we again assume an additive Gaussian noise model ($\boldsymbol{n}_p \sim \mathcal{N}(0, \boldsymbol{R}_p)$):

$$\tilde{\boldsymbol{p}}_i = \pi(_{\boldsymbol{B}}\boldsymbol{r}_{\boldsymbol{BC}_i}) + \boldsymbol{n}_{p,i}. \tag{8}$$

3) Filter States and Prediction Model: The above visual sensor model assumes the knowledge of the tag pose. Instead of using fixed values, which could quickly lead to inconsistencies, we propose to include the pose of the tag into the filter state. Therefore, the filter will be able to refine the tag pose. When multiple tags are used, this will lead to an optimization of the poses of all tags observed within the workspace. Thus, we call this autocalibration.

Together with a robocentric representation of the sensor state, the full filter state will look as follows:

$$\boldsymbol{x} := \left(\boldsymbol{r}, \boldsymbol{v}, \boldsymbol{q}, \boldsymbol{b}_f, \boldsymbol{b}_\omega, \boldsymbol{r}_{\boldsymbol{T}}, \boldsymbol{q}_{\boldsymbol{T}}\right) \tag{9}$$

$$:= \left(_B\boldsymbol{r}_{WB}, _B\boldsymbol{v}_B, \boldsymbol{q}_{WB}, _B\boldsymbol{b}_f, _B\boldsymbol{b}_\omega, _W\boldsymbol{r}_{WT}, \boldsymbol{q}_{WT}\right). \tag{10}$$

In the above state, $\boldsymbol{r}$, $\boldsymbol{v}$, and $\boldsymbol{q}$ are the robocentric position, velocity, and attitude of the sensor. Computing the total derivatives of the selected state and inserting the IMU model (3)-(6) yields:

$$\dot{\boldsymbol{r}} = -\left(\tilde{\boldsymbol{\omega}} - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega\right)^\times \boldsymbol{r} + \boldsymbol{q}(\boldsymbol{v}) + \boldsymbol{w}_r, \tag{11}$$

$$\dot{\boldsymbol{v}} = -\left(\tilde{\boldsymbol{\omega}} - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega\right)^\times \boldsymbol{v} + \tilde{\boldsymbol{f}} - \boldsymbol{b}_f - \boldsymbol{w}_f + \boldsymbol{q}^{-1}(\boldsymbol{g}), \tag{12}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{q}(\tilde{\boldsymbol{\omega}} - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega), \tag{13}$$

$$\dot{\boldsymbol{b}}_f = \boldsymbol{w}_{bf}, \tag{14}$$

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega}, \tag{15}$$

$$\dot{\boldsymbol{r}}_{\boldsymbol{T}} = \boldsymbol{w}_{rt}, \tag{16}$$

$$\dot{\boldsymbol{q}}_{\boldsymbol{T}} = \boldsymbol{w}_{qt}. \tag{17}$$

We include additional continuous white Gaussian noise processes $\boldsymbol{w}_r$, $\boldsymbol{w}_{rt}$, and $\boldsymbol{w}_{qt}$ in order to excite the full filter state and for modeling errors caused by the subsequent discretization of the states. For all white Gaussian noise processes $\boldsymbol{w}_*$, the corresponding covariance parameters $\boldsymbol{R}_*$ describe the magnitude of the noise. Except for $\boldsymbol{R}_r$, $\boldsymbol{R}_{rt}$, and $\boldsymbol{R}_{qt}$ which are the main tuning parameters, all covariance parameters can be chosen by considering the corresponding sensor specifications. Using a simple Euler forward integration scheme we can derive the following discrete time prediction equations:

$$\boldsymbol{r}_k = \left(I - \Delta t_k \left(\tilde{\boldsymbol{\omega}}_k - \boldsymbol{b}_{\omega,k-1} - \boldsymbol{w}_{\omega,k}\right)^\times\right)\boldsymbol{r}_{k-1} + \Delta t_k \left(\boldsymbol{q}_{k-1}(\boldsymbol{v}_{k-1}) + \boldsymbol{w}_{r,k}\right), \tag{18}$$

$$\boldsymbol{v}_k = \left(I - \Delta t_k \left(\tilde{\boldsymbol{\omega}}_k - \boldsymbol{b}_{\omega,k-1} - \boldsymbol{w}_{\omega,k}\right)^\times\right)\boldsymbol{v}_{k-1} + \Delta t_k\Big(\tilde{\boldsymbol{f}}_k - \boldsymbol{b}_{f,k-1} - \boldsymbol{w}_{f,k} + \boldsymbol{q}_{k-1}^{-1}(\boldsymbol{g})\Big), \tag{19}$$

$$\boldsymbol{q}_k = \boldsymbol{q}_{k-1} \otimes \exp\left(-\Delta t_k(\tilde{\boldsymbol{\omega}}_k - \boldsymbol{b}_{\omega,k-1} - \boldsymbol{w}_{\omega,k})\right), \tag{20}$$

$$\boldsymbol{b}_{f,k} = \boldsymbol{b}_{f,k-1} + \Delta t_k \boldsymbol{w}_{bf,k}, \tag{21}$$

$$\boldsymbol{b}_{\omega,k} = \boldsymbol{b}_{\omega,k-1} + \Delta t_k \boldsymbol{w}_{b\omega,k}, \tag{22}$$

$$\boldsymbol{r}_{\boldsymbol{T},k} = \boldsymbol{r}_{\boldsymbol{T},k-1} + \Delta t_k \boldsymbol{w}_{rt,k}, \tag{23}$$

$$\boldsymbol{r}_{\boldsymbol{T},k} = \boldsymbol{r}_{\boldsymbol{T},k-1} + \Delta t_k \boldsymbol{w}_{qt,k}, \tag{24}$$

with $\Delta t_k = t_k - t_{k-1}$.

4) Update Model: The update step is performed by directly employing the reprojection error as the Kalman filter innovation term. For each tag corner i and based on equation (8) we can write:

$$\boldsymbol{y}_i = \tilde{\boldsymbol{p}}_i - \pi(\boldsymbol{_B r_{BC_i}}). \tag{25}$$

This is performed for every tag detected in the current camera frame. For each newly observed tag the state is augmented by an additional tag pose (position and attitude). The augmentation uses the estimated relative pose from the tag tracker in order to initialize the state with a good linearization point. The corresponding covariance matrices are initialized to large values and typically converge very quickly.

III. Results

In order to assess the performance of the proposed approach, we define two test procedures. In a first test, we verify the accuracy of the fiducial pose estimation. In a second test, we then compare the state estimation computed by our EKF to ground truth data obtained from a high class motion capture system.

In order to collect data with the motion capture system, both the sensor as well as the tags are equipped with passive, reflective markers. These markers are manually placed on the tags and a reference coordinate system is fitted to the marker in the motion capture software provided by the motion capture vendor. Since both processes are manual steps, they are subject to inaccuracies. This will result in small static offsets in position and orientation measurements. However, measured velocities and rotational rates should still be fairly accurate, and be independent of such inaccuracies if expressed in the robocentric frame.

A. Datasets

For both tests, we are using two of the datasets that are available for download with the source code. The first dataset "table" consists of three tags that are placed flat on a table at the same orientation as shown in Figure 2. The distances between the tags are chosen to be of similar magnitude.

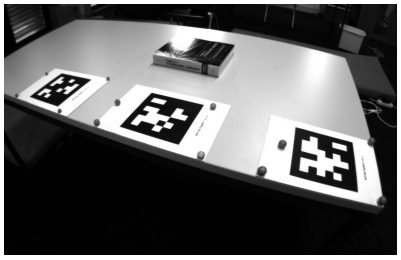

Fig. 2. Setup for dataset "table" as seen by the camera. The tags are aligned and placed flat on a table. This allows for obtaining manual measurements as ground truth information.

The second dataset "dataset_1" also contains three tags. This time, we tried to create a challenging dataset, where the tags are sparsely distributed around a larger workspace of about 4x4x4m. Furthermore, the tags are intentionally oriented in such a way that the viewing angle for the camera is not ideal. Additionally, the sensor is moved fast, such that occasionally motion blur occurs. This fast motion combined with the large workspace also reduces the time that two neighboring tags are visible in the same image. Hence, this increases the level of difficulty in estimating the tags' locations. While one could easily improve or resolve at least some of the issues demonstrated in this dataset, we use it as a bad-case example to illustrate the robustness but also the limitations of the proposed approach. An on-board image taken by the camera, showing the challenging setup as well as the motion blur mentioned above is shown in Figure 3

Fig. 3. Setup for dataset "dataset_1" as seen by the camera. This image shows the large scale of the workspace, as well as the sparsely distributed tags at non-ideal viewing angles. Furthermore, motion blur due to fast motions can be observed.

To provide comparable results, no test specific parameter tuning has been performed, i.e. the same parameters are used in all tests.

B. Fiducial Estimation Test

For the verification of the continuous fiducial estimation procedure, we use the "table" dataset. In this dataset, manual measuring the offset between the tags is simple. Thus, we can use these measurements as ground truth information and compare it to the estimates of the motion capture system. This allows us also to evaluate the accuracy of the marker and coordinate system placement during the set up of the motion capture system.

We compare the norm of the relative translation, i.e. the distance between tag 0 and tag 1 as well as between tag 0 and tag 2 with the manual distance measurements. This error plots are shown in Figure 4 and 5. The plots shows two interesting aspects. The errors in the beginning of the sequence is quite small. This indicates that the initial guess obtained from the reprojection error optimization on the first frame is fairly accurate. Over time, our EKF filter then further refines the poses, reaching approximately millimeter accuracy which is of equal magnitude as manual measuring errors.

The figures also shows the error with respect to the motion capture measurement. Here, the error is shifted by about 1mm for the translation between tag 0 and tag 2. Since a zero mean error curve would be expected, this constant offset most likely results from inaccurate marker and reference coordinate system placement.

Since the tags in this dataset are placed flat on a table and aligned with the table's edge, we can also analyze the rotation error of our tag pose estimates. To do so, we compute

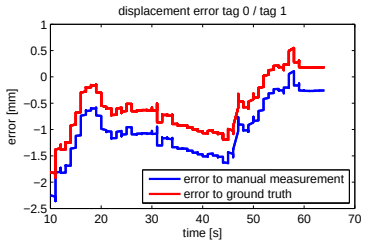

Fig. 4. Displacement error between estimated tag positions and manual as well as motion capture references. As can be seen, the error decreases over time, since the tags' positions are iteratively refined by the EKF. Finally, submillimeter accuracy is achieved.

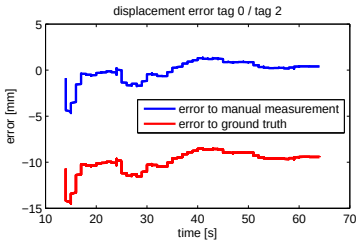

Fig. 5. Displacement error between estimated tag positions and manual as well as motion capture references. Also in this test, the position error decreases over time. In this plot, we can see a significant difference between the error with respect to the motion capture data and the manually measured data. This is most likely to inaccurate marker and coordinate system placement in the motion capture system which creates a steady state offset.

the relative rotation between two tags. We then convert this rotation to an axis-angle representation and use the angle as our error measurement. Due to the tag alignment, the relative rotation between two tags can be assumed to be the identity rotation. This is also confirmed by the motion capture system up to at least the fourth decimal of the relative rotation angle. Therefore, we simply take the identity rotation as the reference. Figure 6 shows the error between estimated rotation and the identity rotation for the relative rotation between tag 0 and tag 1. As can be seen, the error is initially around 0.5 degrees. Through continuous refinement of the tag poses by the EKF filter, this error reduces to around 0.2 degrees over time. This error is of same magnitude as printing and measurement accuracy.

The experiments described above show the high achievable fiducial estimation accuracy for both translations and rotations of the proposed approach. Furthermore, these results underline that tag pose refinement significantly reduces displacement and rotational errors present in the initial

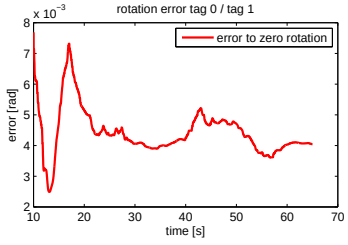

Fig. 6. Rotation error between estimated tag positions and zero rotation. The error is obtained by converting the relative rotation to angle-axis representation of which the angle is plotted. As can be seen, the error decreases over time, since the tags' rotations are iteratively refined by the EKF. The error starts at around 0.5 degrees and reduce to about 0.2 degrees.

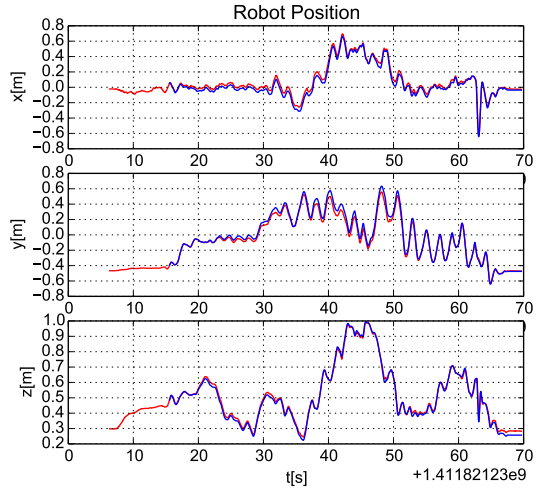

Fig. 7. Comparison between estimated robot position (blue) and ground truth position (red) for the dataset "table". As can be seen, the maximum position offset between both measurements lie only within a centimeter scale which is the same magnitude as the achievable measurement accuracy in this setup.

single frame pose estimate used for initialization. This will eventually improve the consistency of the relative tag poses and thus should also improve the robot's pose estimation results.

C. State Estimation Test

1) Dataset Table: To assess the performance of the state estimation, we use both datasets described above. The goal of our estimation framework is to localize against our workspace, where we choose tag 1 as the origin. This choice is arbitrary and one could choose any tag as the reference tag defining the workspace location and orientation. Since our estimator automatically estimates the orientation of the workspace with respect to gravity, no manual alignment is required.

Figure 7 shows a comparison of the position estimates of the filter and ground truth data from the motion capture system for the *table* dataset. This plot nicely illustrates the robust tracking behavior of the system. Even though the reference tag is not detectable at every instance of the dataset, the estimated fiducials provide a stable reference for the filter to localize against, such that tracking errors remain a few centimeters.

In Figure 8 the estimated orientation and ground truth orientation for the same datasest are compared. Also here, the estimator shows a robust tracking with minimal deviations. The maximum error observed in pitch direction and is about 0.05 rad which corresponds to less than 3 degrees.

Since the ground truth reference data is a relative pose between the sensor and the reference tag computed from the individual poses, the error magnitudes observed above lie within the measurement accuracy of the ground truth data. While this underlines the performance of the approach, it does not give any indication about its limits. Therefore, we tried to push the system to its limits using *dataset_1* which contains several artificial challenges as described above.

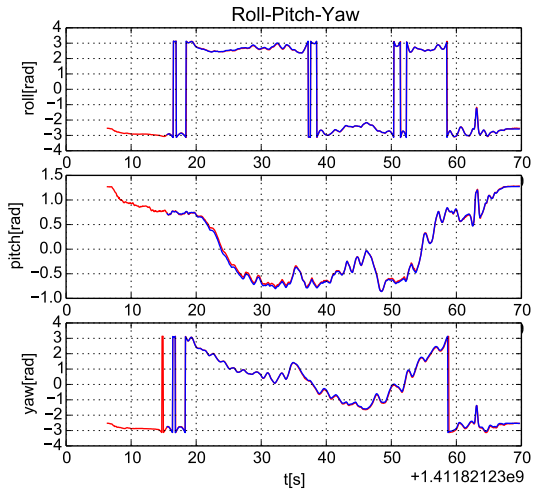

Fig. 8. Comparison between estimated robot orientation (blue) and ground truth orientation (red) for the dataset "table". Due to the wrap-around at +/-π the plot is discontinuous. However, since quaternions are used for the internal representation of the filter, the output of the filter is smooth. As also seen in the position data, estimated and ground truth rotations agree up to measurement uncertainty.

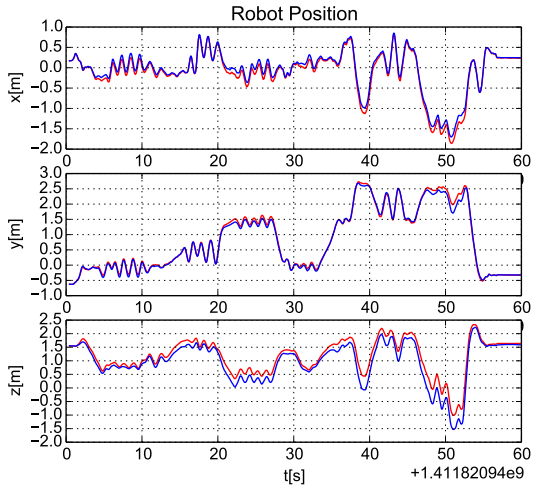

Fig. 9. Comparison between estimated robot position (blue) and ground truth position (red) for the dataset "dataset_1". This dataset has been made artificially difficult with sparse tag coverage and fast motions to show the robustness of the filter. While the estimates diverges when only the briefly observed tag on the very left can be used for localization, it converges back to the ground truth information when localizing against the other tags again.

2) Dataset Dataset_1: In this experiment, again the estimated position is compared to ground truth data and the results are shown in Figure 9. As the plots show, the position starts to deviate from ground truth in the last third of the sequence.

While results are not as good as in the *table* dataset, *dataset_1* can be seen as a worst-case benchmark scenario. Most of the difficulties for the algorithm are artificially posed and the performed motion is faster than in many robot applications. Due to the sparse tag placement and fast motions, the detector was unable to detect any tag in many of the images of the sequence. This is shown in Figure 11 where these instances are marked with the value 1. In total,

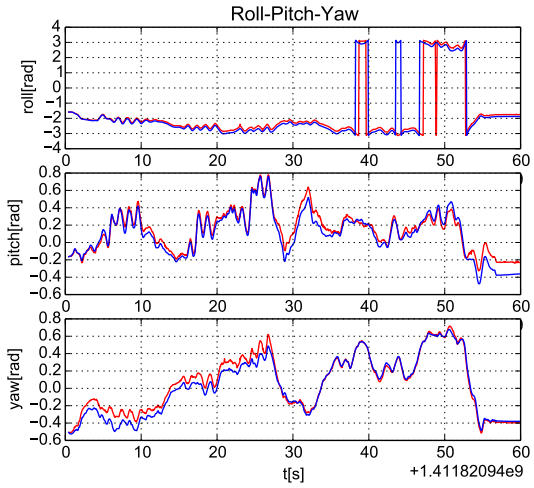

Fig. 10. Comparison between estimated robot orientation (blue) and ground truth orientation (red) for the dataset "orientation_1". Due to the wrap-around at +/-π the plot is discontinuous. However, since quaternions are used for the internal representation of the filter, the output of the filter is smooth.

Fig. 11. Plot indicating whether one or multiple tags were detected (indicated as 0) or no tag was detected (indicated as 1) for *dataset_1*. Overall, in almost 20% of all images no tag could be detected.

the filter is provided only with inertial measurements for almost 20 % of the sequence. Additionally, tag 0 is only seen together with another tag for in total 9 frames. Thus, little localization information is provided for this tag, leading to a high uncertainty of the tags pose. Still, it is the only visible tag for about 15 % of the dataset. Thus, the filter is only provided with uncertain vision information and noisy inertial measurements during these parts. However, the filter remains stable and is able to converge close to ground truth data again when the other tags are visible again.

Also in the orientation, the effects of sparsely distributed tags combined with fast motions are visible. Figure 10 shows the difference between ground truth and estimated orientation for *dataset_1*. As can be seen, the orientation estimate is fairly accurate throughout the dataset with a slight deviation in yaw at the beginning of the trajectory and a small deviation of pitch of about 9 degrees towards the end.

When looking at the linear velocity estimates for this dataset shown in Figure 13, one can see that the estimates agree well with the velocity data obtained by using finite differences on the ground truth data. Interestingly, the estimated velocities are virtually outlier free while the finite differences show occasional peaks. This effect still occurs, even though a high quality motion capture system has been used. This underlines the limitations of using finite differences for velocity estimates and encourages the use of inertial data.

This effect is even more pronounced when looking at Figure 12 which shows the rotational velocity estimates and their counterparts computed using finite differences on

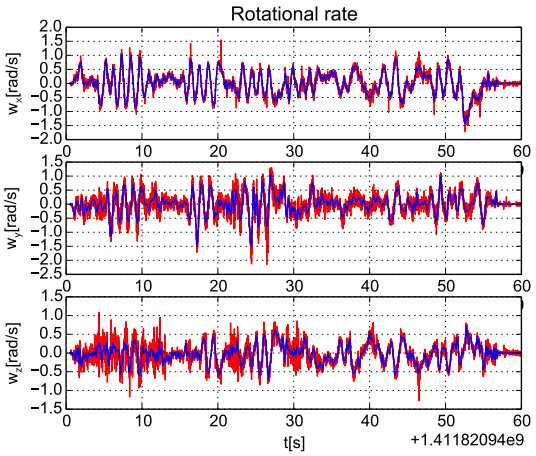

Fig. 12. Comparison of rotational velocity estimates (blue) and rotational velocities calculated by using finite differences of the ground truth orientation data (red). As also with the linear velocities shown in Figure 13, the estimation matches the ground truth data. Here the significance of using inertial measurements for low-noise estimates over finite differences on pose information is even more prominent.

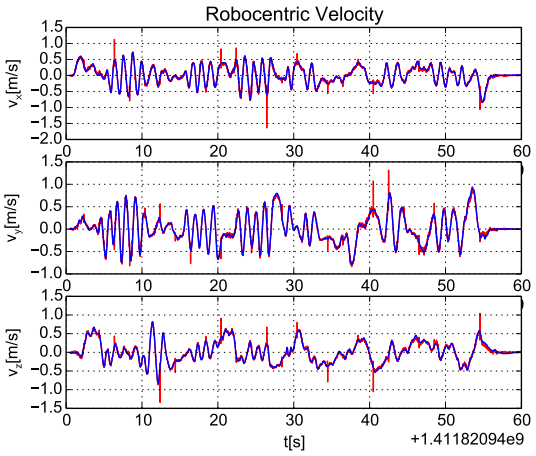

Fig. 13. Comparison of linear velocity estimates (blue) and linear velocities calculated by using finite differences of the ground truth position data (red). While the estimated velocities agree well with the velocities computed from ground truth data, they are virtually outlier free. While a high quality motion capture system is used, this data still shows the limitations of finite differences for velocity estimates.

the ground truth orientation. The difference in noise level between both measurements is significant. One reason is that the IMU directly measures rotational rates using gyroscopes. Furthermore, rotations tend to be more difficult to estimate for motion capture systems. This effect gets amplified when differentiating this noisy signal.

In conclusion, the low tag detection rate underlines the difficulty of the dataset but also the importance of using inertial measurements to provide continuous estimates and guarantee the stability of the filter. Another reason for using inertial measurement data is the high noise obtained when differentiating pose information. This noise is also present in data obtained from high quality motion capture.

While fast motions are unavoidable in some applications, these artificial difficulties can be easily avoided in real world applications. First, it is recommended to have a higher tag density in the workspace. This increases the likelihood of detecting a tag. Furthermore, we reinitialize our estimation before each test. In practice, the estimation quality would improve over time since the accuracy of the estimated tag poses also increases (see Figure 4). For highly sensitive tasks, one could even consider a short calibration procedure, where the sensor is moved through the workspace while pre-running the estimation.

IV. Conclusion

In this paper, we have presented an open-source, visual-inertial state estimation system, that combines the benefits of SLAM and fiducial based estimation. By relying on standard hardware already present on most robots the system can be applied cost efficiently. Due to its good accuracy and high robustness, it could replace an expensive motion capture systems in applications that do not require the high precision or update rates only offered by highly expensive motion capture systems.

Due to the continuous estimation of the fiducials, no calibration procedure is required when switching or changing workspaces. This also allows for placing fiducials based on optimal visibility. As results show, good workspace coverage of fiducials can significantly improve the estimation quality. Usually more tags mean more calibration effort. In contrast, using a EKF-SLAM approach, the system frees the user from this burden, eliminating the trade-off between fiducial density and calibration complexity.

Furthermore, the EKF-based estimator combines localization and calibration in a near optimal but yet computationally very efficient way. This ensures smoothness of the estimates and helps to leverage information from simultaneously observing multiple tags. Experiments under fast motions and sparse tag coverage of the workspace underline the importance of including inertial measurements. This additional information ensures the robustness of the estimator and provides interruption-free state estimates. Furthermore, the inertial measurements ensure high quality translational and rotational velocity estimates.

By granting full access to our source code and datasets, as well as a seamless integration into ROS, we provide all tools for running the system on other hardware, verifying the results presented in this paper and improving the system.

V. Future Work

Results have shown that good coverage of fiducials is important for a good estimation quality. Ideally, one could use differently sized fiducials such that their size can be optimized for their intended location. In this context, it could be investigated to estimate the tag size, eliminating yet another parameter.

As seen in experiments under fast motions and sparse tag coverage, the estimated position will slightly start to drift if no tag is visible. This is due to the integration of noisy inertial measurements. To reduce this drift, estimates from non-artificial features could be integrated in the filter as well.

This would also allow the robot for leaving the workspace for longer time periods.

Especially when using a separate camera and IMU, obtaining a good extrinsic calibration can be tricky. To make the system less sensitive to these parameters and also easier to use, we will investigate on estimating the extrinsic calibration online.

ACKNOWLEDGEMENT

The authors would like to thank Sammy Omari, the Autonomous Systems Lab and Skybotix for their support with the motion capture system and the VI-Sensor. This research has been funded partially through a Swiss National Science Foundation Professorship award to Jonas Buchli.

REFERENCES

[1] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3400–3407.

[2] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[3] S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso, "Ssl-vision: The shared vision system for the robocup small size league," in *RoboCup 2009: Robot Soccer World Cup XIII*. Springer, 2010, pp. 425–436.

[4] M. Fiala, "Artag revision 1, a fiducial marker system using digital techniques," 2004.

[5] M. Faessler, E. Mueggler, K. Schwabe, and D. Scaramuzza, "A monocular pose estimation system based on infrared leds."

[6] A. Breitenmoser, L. Kneip, and R. Siegwart, "A monocular vision-based system for 6d relative robot localization," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept 2011, pp. 79–85.

[7] H. Lim and Y.-S. Lee, "Real-time single camera slam using fiducial markers," in *ICCAS-SICE, 2009*, Aug 2009, pp. 177–182.

[8] A. Mourikis, S. Roumeliotis, *et al.*, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3565–3572.

[9] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," *Int. Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, Nov. 2011.

[10] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[11] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of imu and vision for absolute scale estimation in monocular slam," *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 287–299, 2011.

[12] M. Li and A. I. Mourikis, "High-precision, consistent ekf-based visual–inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[13] J. Sola, T. Vidal-Calleja, J. Civera, and J. M. M. Montiel, "Impact of landmark parametrization on monocular ekf-slam with points and lines," *International journal of computer vision*, vol. 97, no. 3, pp. 339–368, 2012.

[14] M. Maidi, F. Ababsa, and M. Mallem, "Vision-inertial tracking system for robust fiducials registration in augmented reality," in *Computational Intelligence for Multimedia Signal and Vision Processing, 2009. CIMSVP '09. IEEE Symposium on*, March 2009, pp. 83–90.

[15] E. Foxlin and L. Naimark, "Vis-tracker: a wearable vision-inertial self-tracker," in *Virtual Reality, 2003. Proceedings. IEEE*, March 2003, pp. 199–206.

[16] S. You and U. Neumann, "Fusion of vision and gyro tracking for robust augmented reality registration," in *Virtual Reality, 2001. Proceedings. IEEE*. IEEE, 2001, pp. 71–78.

[17] I. P. H. Kato, M. Billinghurst, and I. Poupyrev, "Artoolkit user manual, version 2.33," *Human Interface Technology Lab, University of Washington*, vol. 2, 2000.

[18] J. Rekimoto and Y. Ayatsuka, "Cybercode: designing augmented reality environments with visual tags," in *Proceedings of DARE 2000 on Designing augmented reality environments*. ACM, 2000, pp. 1–10.

[19] Y. Cho, J. Lee, and U. Neumann, "A multi-ring color fiducial system and an intensity-invariant detection method for scalable fiducial-tracking augmented reality," in *In IWAR*. Citeseer, 1998.

[20] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Y. Siegwart, "A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam." IEEE International Conference on Robotics and Automation (ICRA 2014), 2014.