AI-Based Multi-Object Relative State Estimation with Self-Calibration Capabilities

Thomas Jantos[1], Christian Brommer[1], Eren Allak[1], Stephan Weiss[1] and Jan Steinbrener[1]

***Abstract*— The capability to extract task specific, semantic information from raw sensory data is a crucial requirement for many applications of mobile robotics. Autonomous inspection of critical infrastructure with Unmanned Aerial Vehicles (UAVs), for example, requires precise navigation relative to the structure that is to be inspected. Recently, Artificial Intelligence (AI)-based methods have been shown to excel at extracting semantic information such as 6 degree-of-freedom (6-DoF) poses of objects from images.**

In this paper, we propose a method combining a state-of-the-art AI-based pose estimator for objects in camera images with data from an inertial measurement unit (IMU) for 6-DoF multi-object relative state estimation of a mobile robot. The AI-based pose estimator detects multiple objects of interest in camera images along with their relative poses. These measurements are fused with IMU data in a state-of-the-art sensor fusion framework. We illustrate the feasibility of our proposed method with real world experiments for different trajectories and number of arbitrarily placed objects. We show that the results can be reliably reproduced due to the self-calibrating capabilities of our approach.

I. INTRODUCTION

Mobile robots, such as unmanned aerial vehicles (UAVs), rely on the information of their on-board sensors to autonomously navigate the world. Semantic information, i.e. the higher-level meaning of sensor data, can improve a robot's ability to navigate in its surroundings and allows for more complicated tasks [1]. In semantic navigation, the robot moves depending on context or task, in many cases with respect to objects of interest in the scene. Such tasks include infrastructure inspection [2] or object tracking [3]. While the goal for the latter is to keep the moving object in the field of view of the camera, infrastructure inspection requires accurate positioning of the robot with respect to a typically static object of interest. Semantic information extracted from the robot's sensor data, namely the detection of the object of interest and its pose relative to the robot are important elements to achieving this task. For example, monitoring power pole insulators for possible damages requires a UAV to fly around the desired insulator and take high resolution images from specific positions to allow for detection of damage or changes over time.

Current autonomous mission execution is typically based on global navigation satellite system (GNSS) for localization of the UAV. GNSS always provides a global position and not a relative position with respect to an object of interest. Moreover, the accuracy is too low for precise, centimeter-range navigation and GNSS is prone to signal loss in proximity to large structures. In this case often other sensor modalities are considered, e.g. visual-inertial odometry (VIO) [4]. With VIO, a local pose can be estimated by combining the movement of geometrical features (edges, corners) in monocular camera images with data from an inertial measurement unit (IMU). Classical, feature extraction based algorithms are not well suited for semantic navigation as they rely on raw features that do not provide information about any objects in the scene and they struggle with fast or slow motion [5].

Recent advancements in artificial intelligence (AI) led to a breakthrough in the extraction of semantic information from raw sensor measurements like path detection using camera images [6], object recognition with laser scanners [7], semantic segmentation for scene understanding [8], and recently, 6 degree-of-freedom (6-DoF) pose estimation of objects for robotic grasping [9]. Furthermore, the availability of AI capable edge computing devices enables the usage of such methods on mobile robots.

In this paper, we investigate the suitability of AI-based pose estimation for full 6-DoF, object relative state estimation for mobile robotics. We consider a minimal sensor configuration consisting of a single monocular camera and an IMU in line with size, weight and computational power constraints of mobile robotic platforms such as UAVs. We utilize an AI-based pose estimator to detect, classify and

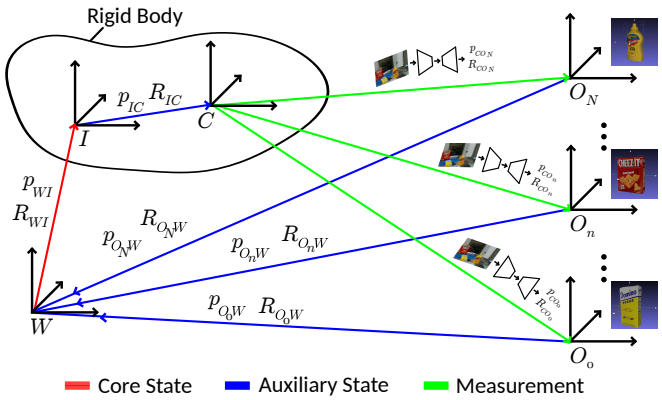

Fig. 1. Visualization of the coordinate frames in this work. We estimate the state of a fixed rigid body consisting of IMU I and camera C relative to up to N different objects O_k with respect to a fixed but arbitrary navigation world W. In addition to the core states (red), we also estimate the calibration between IMU and camera (blue). We also estimate the pose of the object frames with respect to the world (blue). Our pose sensor consists of AI-based 6-DoF relative pose measurements between camera and objects (green).

[1]The authors are with the Control of Networked Systems Group, University of Klagenfurt, 9020 Klagenfurt am Wörthersee, Austria firstname.lastname@ieee.org

This work was supported by the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) under the grant agreement 881082 (MUKISANO).

Pre-print version, DOI: 10.1109/ICRA48891.2023.10161375 ©IEEE.

estimate the 6-DoF poses of objects of interest contained in each camera image, and then fuse the information with IMU measurements in a state-of-the-art sensor fusion framework to infer the 6-DoF object-relative pose of the robot. A schematic overview of our approach is presented in Fig. 1. Our main contributions can be summarized as follows:

- Extracting semantic information from images with AI and fusing this relative pose information with IMU data for accurate, 6-DoF, object relative state estimation.
- Formulating a filter-based method to estimate the state of the mobile robot and the pose of multiple, different objects based on 6-DoF relative pose measurements.
- Providing a self-calibrating formulation of the filter that does not require any assumptions about the global or relative positions of the different objects in a scene.
- Validating the proposed approach with several real world experiments using objects of a popular 6-DoF object pose challenge data set to show that our method works for different trajectories and different number of objects with reproducible performance.

The remainder of the paper is organized as follows. In Section II, we summarize the related work. In Section III, we present how we integrate object relative pose measurements into a state estimation framework. In Section IV, the experiments and the corresponding results are discussed. Finally, the paper is concluded in Section V.

II. RELATED WORK

For state estimation in mobile robotics, typically IMU data and one or more pose sensors such as GNSS are fused together. GNSS provides global position information but not 3D orientation information. In the absence of GNSS signals, VIO, the combination of a monocular camera and IMU data, can estimate the pose of a robot by triangulating the position of the camera given geometrical features from an image and estimating the remaining scale factor with inertial data [4], [10]. Fusing multiple sensors yields a more robust and reliable estimate of the robot's state. There exist mainly two different approaches for sensor fusion: filter-based, recursive and optimization-based methods. The latter can yield more accurate state estimates but is computationally more demanding due to optimizing across several sensor measurements [11]. In comparison, filtering-based methods, such as Extended Kalman Filters (EKF) [12], [13], are computationally more efficient and thus are well-suited for mobile robotics.

GNSS and classical VIO do not provide object relative pose measurements and thus are not suitable for object relative state estimation. However, image-based 6-DoF relative object pose estimation methods can be utilized as a pose sensor in state estimation frameworks. There exist classical approaches and AI methods based on deep learning. Classical approaches are either template-based, where the object pose is determined by finding a matching template for the current image [14], or feature-based, where keypoints are extracted from the image and then matched to the 3D object model [15]. On the other hand, deep learning-based approaches are mostly end-to-end learned methods, where the 6-DoF pose is directly estimated from the input image using convolutional neural networks (CNNs). Deep learning-based methods can be divided even further depending on the amount of additional information used. [9], [16] take a single RGB image as input to their network and employ symmetry aware losses during training that make use of 3D object model information. 3D object models can also be provided as an additional input to the network [17], utilized for refining an initial pose estimate [18], [19], [20] or for matching keypoints, which were regressed by the network [21]. Other forms of additional information consist of taking multiple images [22], [23] or depth maps [24], [25]. Recently, we have proposed PoET [26] a 6-DoF multi-object pose estimation framework that achieves state-of-the-art results on benchmark datasets and only takes a single RGB image as input and does not require any additional information during training or inference.

An alternative to object relative state estimation is simultaneous localization and mapping (SLAM) on an object level. [27] uses depth images to extract 6-Dof object pose information. Similar to us, [28] use an AI-based pose estimator to predict 6-DoF relative object poses from images. Both approaches fuse the 6-DoF relative pose information from multiple view points together and combine it with graph optimization to estimate the pose of the camera and objects with respect to a map. However, they do not use any other sensors, such as IMU, in their approach. [29] combines IMU measurements, geometric features from images, and 6-DoF object poses in a SLAM approach. Graph optimization still needs to be performed for a graph containing all object poses. In general, the requirement for a map and optimization in SLAM results in a higher computational load for the mobile robot. Our approach still allows for object relative state estimation without this requirement.

Object relative state estimation for mobile robotics has been shown in [30], where a UAV localizes itself with respect to cylinder shaped infrastructure by extracting geometrical features from images and assuming a known radius. Similarly, [31] used a color-based ellipse-detection algorithm to first detect the object of interest in the image and then used the knowledge about object size, visual appearance and camera parameters to calculate the relative pose to the object. Meanwhile, [32] investigated different classical 6-DoF object pose estimation approaches for object-relative state estimation. They also investigated the use of machine learning to detect the presence of objects by training simple classifiers on classical features extracted from images.

In contrast to that, we propose here a fully AI-based method to extract semantic information from camera images. We do not need to define a geometric model, keypoints or templates to map object appearances in images to relative 6-DoF poses. Moreover, AI-based models are not limited to specific geometric object shapes and remove the need for handcrafted features. In our previous work [26], we have introduced PoET for 6-DoF pose estimation of objects in RGB images using state-of-the-art AI methods. We mainly

focused on the definition, the training, a thorough ablation study and comparison to other deep learning-based methods on benchmark datasets for 6-DoF multi-object pose estimation. In this work, we present a detailed investigation of the suitability of our AI-based object pose estimator as pose sensor for 6-DoF object-relative state estimation of a mobile robot using a state-of-the-art sensor fusion framework with multiple real world experiments.

III. METHOD

In this section, we present the design of our approach. First, we explain the notation used for the measurement equations and transformations of coordinate frames. Second, we reason about the choice of frameworks for 6-DoF pose estimation and state estimation. Finally, we describe how the estimated 6-DoF of several known objects can be combined to estimate the 6-DoF pose of the robot. This includes a detailed description of how our choice of sensor fusion algorithm is extended to include 6-DoF pose measurements of each individual object.

A. Notation

Throughout this paper we use the following notation: given three coordinate frames A, B and C, the transformation ${}_A\mathbf{T}_{BC}$ defines frame C with respect to frame B expressed in frame A. If the left subscript A is omitted, the transformation is defined in frame B. Furthermore, the transformation ${}_A\mathbf{T}_{AB}$ can be split up into two parts namely ${}_A\mathbf{p}_{AB}$ and $\mathbf{R}_{AB}$, which describe the translation and rotation respectively. Alternatively, the rotation can also be expressed by a quaternion $\mathbf{q}_{AB}$. Each quaternion $\mathbf{q}$ can be represented by $\mathbf{q} = [\mathbf{q}_\mathbf{v}\ q_w]^T = [q_x\ q_y\ q_z\ q_w]^T$. The quaternion multiplication is represented by $\otimes$. $\mathbf{I}_3$ and $\mathbf{0}_3$ refer to the identity and the null matrix in $\mathbb{R}^{3x3}$, respectively. $[\omega]_\times$ is the skew-symmetric operator as defined in [33].

B. Pose and State Estimation Frameworks

Mobile robots, in particular UAVs, are subject to payload constraints, which impose not only limitations on the size and amount of sensors a robot can carry, but also on the computational power available for data processing. Hence, the necessity arises for efficient and computationally light algorithms. Therefore, we chose our object pose estimation framework PoET [26] as a 6-DoF pose sensor as it only uses RGB images and does not rely on any depth information and 3D object models, removing the need for additional hardware components and reducing the computational load by not having to process 3D models. In a first step, PoET detects all objects it was trained to detect in an image and also predicts their classes. Afterwards, the predicted bounding boxes and multi-scale feature maps are fed to a transformer architecture to predict the relative, up-to scale 6-DoF pose between the camera and each object. The predicted rotation and translation are unique for non-symmetric objects. For objects with one or more symmetry axes, the rotation or translation for some object poses becomes ambiguous with more than one possible solution. The obvious negative effects of this ambiguity on the pose estimation of the robot can be minimized by considering multiple objects in heterogeneous configuration and fusing individual measurements in a proper sensor fusion framework. This mimics an inspection workflow where typically several distinct parts of interest of the structure to be inspected are visible at the same time.

For the sensor fusion framework, we use MaRS [12] for multi-sensor fusion and state estimation due to being lightweight and computationally efficient as it was developed specifically with mobile robotics in mind. MaRS was designed for modularity and separates the propagation of the core state variables based on inertial data from the state updates based on the measurements of the individual sensors. It also uses abstract sensor classes that are type agnostic. This allows for straightforward integration of new sensor modules. For our method, we define a multi-pose sensor, where a single measurement consists of a single RGB image. From each image, we then extract the 6-DoF relative poses of all detected objects with PoET and use them for the EKF update step as described below.

C. EKF State and Update

As reference frame for the mobile robotic system, we chose the frame of its IMU (I). Thus, our goal is to estimate the pose of the IMU (I) with respect to the world (W) by measuring the 6-DoF relative poses between the camera (C) and a set of objects (O_k). The different coordinate frames are visualized in Fig. 1. As mentioned earlier, the relative poses of the objects with respect to the camera are extracted from the image by our AI-based pose estimator dubbed PoET. PoET will only consider objects that it was trained for. Given a single RGB image, a 6-DoF pose $\mathbf{T}_{CO_k}$ is predicted for each detected object of interest and the assignment of the predicted pose to an object is based on the predicted class. For details about the architectural choices in PoET, we refer the reader to [26]. Depending on the total number of objects N in a scene, the full state vector $\mathbf{X}$ is then defined as:

$$\begin{aligned}\mathbf{X} = [&\mathbf{p}_{WI}^T, \mathbf{v}_{WI}^T, \mathbf{q}_{WI}^T, \mathbf{b}_\omega^T, \mathbf{b}_a^T, \\ &\mathbf{p}_{IC}^T, \mathbf{q}_{IC}^T, \mathbf{p}_{O_0W}^T, \mathbf{q}_{O_0W}^T, \ldots, \mathbf{p}_{O_NW}^T, \mathbf{q}_{O_NW}^T]^T\end{aligned} \tag{1}$$

The core states necessary for state propagation are the position $\mathbf{p}_{WI}$ of the IMU, its velocity $\mathbf{v}_{WI}$ and its orientation $\mathbf{q}_{WI}$ as well as the gyroscopic bias $\mathbf{b}_\omega$ and the accelerometer bias $\mathbf{b}_a$. The pose and velocity dynamics are given as [4]

$$\dot{\mathbf{p}}_{WI} = \mathbf{v}_{WI} \tag{2}$$

$$\dot{\mathbf{v}}_{WI} = \mathbf{R}_{WI}\ (\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a) - \mathbf{g} \tag{3}$$

$$\dot{\mathbf{q}}_{WI} = \frac{1}{2}\Omega(\omega_b - \mathbf{b}_\omega - \mathbf{n}_\omega)\ \mathbf{q}_{WI} \tag{4}$$

where $\mathbf{a}_m$ is the measured acceleration in the IMU frame, $\mathbf{n}_a$ is the accelerometer noise parameter, $\mathbf{g}$ is the gravity vector in W, ω_b is the measured angular velocity in the IMU frame, $\mathbf{n}_\omega$ is the gyroscopic noise parameter, and $\Omega(\omega)$ is the quaternion multiplication matrix of ω. The IMU biases are modeled as random walks.

Furthermore, we estimate the calibration between the IMU and the camera given by $\mathbf{p}_{IC}$ and $\mathbf{q}_{IC}$. Due to implementation

reasons we assume the number of objects in a scene to be known a priori, but neither the global poses of the objects nor the relative poses between objects is known. Therefore, we additionally estimate an object-world which describes the transformation $(\mathbf{p}_{O_kW}, \mathbf{q}_{O_kW})$ between the object frame and the navigation world. Both the camera-IMU extrinsics $\mathbf{p}_{IC}, \mathbf{q}_{IC}$ as well as the object poses $\mathbf{p}_{O_kW}, \mathbf{q}_{O_kW}$ are modeled to remain constant over time.

For each image, every measured relative pose is treated as an independent measurement with which the pose of the camera can be estimated. To calculate the required Jacobians for the update step, we consider the inverted relative pose measurements $\mathbf{T}_{O_kC}$, i.e. the camera pose relative to the object frame. Based on the relationship between the different coordinate frames and the independent relative position $z_{\mathbf{p}_{O_k}}$ and orientation $z_{\mathbf{q}_{O_k}}$ measurements, the residuals for position $\tilde{z}_{\mathbf{p}_{O_k}}$ and orientation $\tilde{z}_{\mathbf{R}_{O_k}}$ can be written as:

$$\begin{aligned}\tilde{z}_{\mathbf{p}_{O_k}} &= z_{\mathbf{p}_{O_k}} - \hat{z}_{\mathbf{p}_{O_k}} \\ &= \mathbf{p}_{O_kC} - (\mathbf{p}_{O_kW} + \mathbf{R}_{O_kW}(\mathbf{p}_{WI} + \mathbf{R}_{WI}\ \mathbf{p}_{IC})) \end{aligned} \tag{5}$$

$$\tilde{z}_{\mathbf{R}_{O_k}} = 2\ \frac{\tilde{z}_{\mathbf{q}_{\mathbf{v}},O_k}}{\tilde{z}_{q_w,O_k}} \tag{6}$$

$$\tilde{z}_{\mathbf{q}_{O_k}} = \hat{z}^{-1}_{\mathbf{q}_{O_k}} \otimes z_{\mathbf{q}_{O_k}} = (\mathbf{q}_{O_kW} \otimes \mathbf{q}_{WI} \otimes \mathbf{q}_{IC})^{-1} \otimes \mathbf{q}_{O_kC} \tag{7}$$

$$\tilde{z}_{O_k} = \begin{bmatrix} \tilde{z}_{\mathbf{p}_{O_k}} \\ \tilde{z}_{\mathbf{R}_{O_k}} \end{bmatrix} \tag{8}$$

Given these residuals and depending on a single pose measurement from object O_k, the Jacobian for the position $H_{\mathbf{p}}$ and orientation $H_{\mathbf{R}}$ with respect to the states is [33]:

$$H_{\mathbf{p},\mathbf{p}_{WI}} = \mathbf{R}_{O_kW} \tag{9}$$

$$H_{\mathbf{p},\mathbf{R}_{WI}} = -\mathbf{R}_{O_kW}\mathbf{R}_{WI}[\mathbf{p}_{IC}]_\times \tag{10}$$

$$H_{\mathbf{p},\mathbf{p}_{IC}} = \mathbf{R}_{O_kW}\mathbf{R}_{WI} \tag{11}$$

$$H_{\mathbf{p},\mathbf{p}_{O_kW}} = \mathbf{I}_3 \tag{12}$$

$$H_{\mathbf{p},\mathbf{R}_{O_kW}} = -\mathbf{R}_{O_kW}[\mathbf{p}_{WI}]_\times - \mathbf{R}_{O_kW}[\mathbf{R}_{WI}\ \mathbf{p}_{IC}]_\times \tag{13}$$

$$H_{\mathbf{R},\mathbf{R}_{WI}} = \mathbf{R}^T_{IC} \tag{14}$$

$$H_{\mathbf{R},\mathbf{R}_{IC}} = \mathbf{I}_3 \tag{15}$$

$$H_{\mathbf{R},\mathbf{R}_{O_kW}} = \mathbf{R}^T_{IC}\mathbf{R}^T_{WI} \tag{16}$$

where, e.g. $H_{\mathbf{p},\mathbf{p}_{WI}}$ only considers the part of the residual $\tilde{z}_{\mathbf{p}_{O_k}}$ that depends on the state $\mathbf{p}_{WI}$. The rest of the Jacobians are equal to $\mathbf{0}_3$. As relative pose measurements for different objects are independent of each other, the Jacobians for the other $(i \neq n)$ object-world states, i.e. $H_{\mathbf{p},\mathbf{p}_{O_iW}}$, $H_{\mathbf{p},\mathbf{R}_{O_iW}}$, $H_{\mathbf{R},\mathbf{p}_{O_iW}}$, $H_{\mathbf{R},\mathbf{R}_{O_iW}}$, are all equal to $\mathbf{0}_3$. For a single object O_k, the Jacobian is given by stacking the individual components:

$$\begin{aligned} H_{\mathbf{p},O_k} = [&H_{\mathbf{p},\mathbf{p}_{WI}}\ H_{\mathbf{p},\mathbf{v}_{WI}}\ H_{\mathbf{p},\mathbf{R}_{WI}}\ H_{\mathbf{p},\mathbf{b}_\omega}\ H_{\mathbf{p},\mathbf{b}_a} \\ &H_{\mathbf{p},\mathbf{p}_{IC}}\ H_{\mathbf{p},\mathbf{R}_{IC}}\ H_{\mathbf{p},\mathbf{p}_{O_0W}}\ H_{\mathbf{p},\mathbf{R}_{O_0W}} \\ &\ldots H_{\mathbf{p},\mathbf{p}_{O_NW}}\ H_{\mathbf{p},\mathbf{R}_{O_NW}}] \end{aligned} \tag{17}$$

$$\begin{aligned} H_{\mathbf{R},O_k} = [&H_{\mathbf{R},\mathbf{R}_{WI}}\ H_{\mathbf{R},\mathbf{v}_{WI}}\ H_{\mathbf{R},\mathbf{R}_{WI}}\ H_{\mathbf{R},\mathbf{b}_\omega}\ H_{\mathbf{R},\mathbf{b}_a} \\ &H_{\mathbf{R},\mathbf{p}_{IC}}\ H_{\mathbf{R},\mathbf{R}_{IC}}\ H_{\mathbf{R},\mathbf{p}_{O_0W}}\ H_{\mathbf{R},\mathbf{R}_{O_0W}} \\ &\ldots H_{\mathbf{R},\mathbf{p}_{O_NW}}\ H_{\mathbf{R},\mathbf{R}_{O_NW}}] \end{aligned} \tag{18}$$

$$H_{O_k} = \begin{bmatrix} H_{\mathbf{p},O_k} \\ H_{\mathbf{R},O_k} \end{bmatrix} \tag{19}$$

Depending on the current image, the final residual z and observation matrix $\mathbf{H}$ for the state update is determined by vertically stacking the residuals and Jacobians, individually, from Eq. (8) and Eq. (19), respectively, for each object that was detected for the current update step. Similar to hardware sensors, our AI-based pose sensor might return faulty or inaccurate measurements. In a similar fashion as described in [34], we conduct a χ^2 test based on the EKF innovation $\mathbf{S}$ and the residual to detect outlier measurements. This is applied for the measurement of each object individually. Outlier measurements for object O_k are then rejected and the final residual and Jacobian have to be rebuild by masking the corresponding rows. The correction is then calculated based on this final total residual and associated Jacobian. In the update step, measurement uncertainties for each measurement can be considered. For the present work, these uncertainties have been fixed to 10 cm and 20 degrees for the position and orientation measurement of each object, respectively. These numbers were determined based on the standard deviation of PoET across a video sequence reported in [26].

The proper initialization of the individual frames is an important aspect to consider. At the beginning of the recording, we initialize an arbitrary but fixed navigation world W. Without loss of generality, the IMU frame is initialized at the origin of W. The initial extrinsic calibration between the IMU and camera is determined through visual-inertial calibration [35]. Each object-world is initialized when the corresponding object is seen by the camera for the first time. The object frame is then placed with respect to the world frame by taking the currently estimated pose of the camera and the relative pose measurement:

$$\mathbf{R}_{O_kW} = \mathbf{R}_{O_kC}\mathbf{R}^T_{IC}\mathbf{R}^T_{WI} \tag{20}$$

$$\mathbf{p}_{O_kW} = \mathbf{p}_{O_kC} - \mathbf{R}_{O_kW}(\mathbf{R}_{WI}\ \mathbf{p}_{IC} + \mathbf{p}_{WI}) \tag{21}$$

In our problem formulation, the robot's pose I is estimated relative to a set of object-worlds O_k through relative pose measurements. As the robot's pose is relative to a world frame and the measurements are relative to the corresponding object frames, the object-worlds can be placed freely with respect to the world frame, which does cause observability issues. By fixing the state of one object-world reference frame, the system is rendered observable. This fixed object, from now on called the main object O_m, serves as the anchor point for the object-relative 6-DoF state estimation. The position of the main object's frame with respect to the navigation world is not changed, i.e. the corresponding Jacobians $H_{\mathbf{p},\mathbf{p}_{O_mW}}$ and $H_{\mathbf{R},\mathbf{p}_{O_mW}}$ are set to $\mathbf{0}_3$. Measurements in which the main object is not visible in the picture are directly rejected. Otherwise, estimating the object-world of additional objects while the anchor is not visible leads to ambiguous updates.

The propagation step is performed at the rate of the IMU sensor readings, while the update step happens with the frequency of the camera images.

Fig. 2. Object configuration that was used for sequence 4 (left) and object poses as estimated by PoET (right). Note the difference in object package coloring between real-world objects (left) and YCB-V objects used for training PoET (right). The left image, as shown here, is directly fed into our pose estimation framework to get the 6-DoF relative pose measurements.

IV. EXPERIMENTS & RESULTS

In this section, we present the experiments conducted and discuss the results in detail. We trained PoET on the YCB-V dataset [9] as described in [26], a benchmark dataset for 6-DoF pose estimation, and chose a subset of objects to serve as objects of interest in our experiments. The images and IMU data were recorded using an Intel Realsense D435i with an RGB resolution of 1280x720 and 30 FPS. After undistorting the images, they were cropped to a resolution of 640x480, which is the standard resolution of the YCB-V dataset. We record our own real data by placing the objects in our motion capture room and moving around the objects with the camera while tracking the body of the camera. This mimics the inspection of a set of objects of interest with a mobile robotic platform with 6-DoFs. An example object configuration and image can be found in Fig. 2. Because we do not record any information regarding the global position of the objects in the room, the trajectory derived from the object-relative state estimation has to be aligned with the ground truth trajectory to calculate the error metrics. It is important to note the differences between the benchmark data and our own real data. The camera which was used to record the YCB-V dataset has a different recording resolution, field of view, and focal point than the camera used during our experiments. In addition, some real world objects had slightly different appearance (size and coloring) than the ones used for the data set that PoET was trained with. The results reported here have been obtained with the YCB-V trained model of PoET. We did not perform any retraining or fine-tuning of the model to adapt to these differences.

We conducted two different experiments. First, we investigated the performance of our approach for 8 different sequences. The sequences varied with respect to the number of objects present, the constellation of the objects and the trajectory performed by the camera around the objects. We calculate the root mean square error (RMSE) for the position and Euler angles by comparing the estimated trajectory with the respective ground truth trajectory for a single recording of that sequence. Moreover, we calculate the average RMSE and the standard deviation (std) over all trajectories. The results are summarized in Table I. The overall performance across all sequences shows that our method is able to sufficiently estimate the state given AI-based, relative 6-DoF pose measurements for a variety of scenarios. For most sequences, the position RMSE is around 10 cm or less, which is an acceptable error for this rather complex task. Furthermore, these results indicate that our method is applicable for object relative state estimation. In some cases, i.e. sequence 2 and 3, the achieved performance is worse than in others. This is due to outliers in the predictions of the 6-DoF pose estimator. Objects being partially out of the image or ambiguous viewpoints as well as motion blur in the images lead to wrong pose estimates. Especially the latter one results in the predictions for all objects in a single image to be wrong. While the χ^2 test helps to reject such frames, multiple consecutive images with noisy or wrong measurements will still affect the state estimation as it can result in phases with no updates and only IMU propagation leading to

TABLE I

RMSE FOR POSITION AND ORIENTATION ACROSS THE WHOLE TRAJECTORY FOR DIFFERENT SEQUENCES.

Sequence	#objects	(x, y, z) [m]	(roll, pitch, yaw) [deg]
1	3	[0.066, 0.163, 0.034]	[3.00, 6.81, 4.52]
2	3	[0.162, 0.358, 0.233]	[54.38, 19.39, 20.27]
3	2	[0.163, 0.338, 0.136]	[25.34, 16.57, 15.13]
4	4	[0.045, 0.157, 0.108]	[15.38, 5.58, 10.02]
5	1	[0.075, 0.031, 0.054]	[61.12, 9.66, 26.19]
6	3	[0.061, 0.117, 0.029]	[11.00, 5.77, 7.41]
7	4	[0.093, 0.159, 0.099]	[19.38, 12.38, 27.24]
8	3	[0.127, 0.165, 0.104]	[26.19, 11.95, 10.70]
mean	-	[0.099, 0.186, 0.099]	[26.97, 11.0, 15.19]
± std		± [0.043, 0.102, 0.062]	± [19.18, 4.75, 8.01]

TABLE II

RMSE FOR POSITION AND ORIENTATION ACROSS THE WHOLE TRAJECTORY FOR DIFFERENT RUNS OF SEQUENCE 4.

Run	(x, y, z) [m]	(roll, pitch, yaw) [deg]
1	[0.094, 0.107, 0.119]	[4.72, 7.08, 13.16]
2	[0.077, 0.121, 0.103]	[1.33, 7.24, 8.05]
3	[0.049, 0.096, 0.092]	[3.14, 1.72, 14.49]
4	[0.065, 0.119, 0.101]	[7.75, 5.25, 12.90]
5	[0.069, 0.119, 0.099]	[2.79, 3.35, 11.57]
6	[0.047, 0.099, 0.099]	[2.53, 3.63, 12.21]
7	[0.059, 0.090, 0.088]	[1.92, 5.01, 10.89]
8	[0.057, 0.093, 0.083]	[3.68, 2.92, 11.37]
9	[0.074, 0.090, 0.096]	[6.85, 4.69, 13.11]
10	[0.076, 0.101, 0.118]	[7.19, 6.02, 12.10]
mean	[0.067, 0.104, 0.010]	[4.19, 4.69, 11.99]
± std	± [0.014, 0.012, 0.011]	±[2.20, 1.71, 1.64]

TABLE III

RMSE FOR POSITION AND ORIENTATION ACROSS THE WHOLE TRAJECTORY FOR DIFFERENT RUNS OF SEQUENCE 6.

Run	(x, y, z) [m]	(roll, pitch, yaw) [deg]
1	[0.061, 0.117, 0.029]	[11.00, 5.77, 7.41]
2	[0.092, 0.122, 0.052]	[17.17, 16.97, 7.16]
3	[0.072, 0.112, 0.025]	[13.08, 24.64, 5.93]
4	[0.083, 0.110, 0.037]	[12.11, 8.24, 4.58]
5	[0.074, 0.089, 0.023]	[16.02, 5.99, 6.15]
6	[0.067, 0.079, 0.031]	[12.29, 11.97, 7.19]
7	[0.064, 0.126, 0.025]	[16.64, 12.26, 7.72]
8	[0.059, 0.093, 0.036]	[15.86, 6.31, 6.41]
9	[0.067, 0.092, 0.046]	[12.38, 4.69, 4.49]
10	[0.057, 0.150, 0.031]	[12.95, 16.12, 6.89]
mean	[0.069, 0.109, 0.034]	[13.95, 11.29, 6.39]
± std	± [0.010, 0.020, 0.009]	±[2.11, 6.09, 1.07]

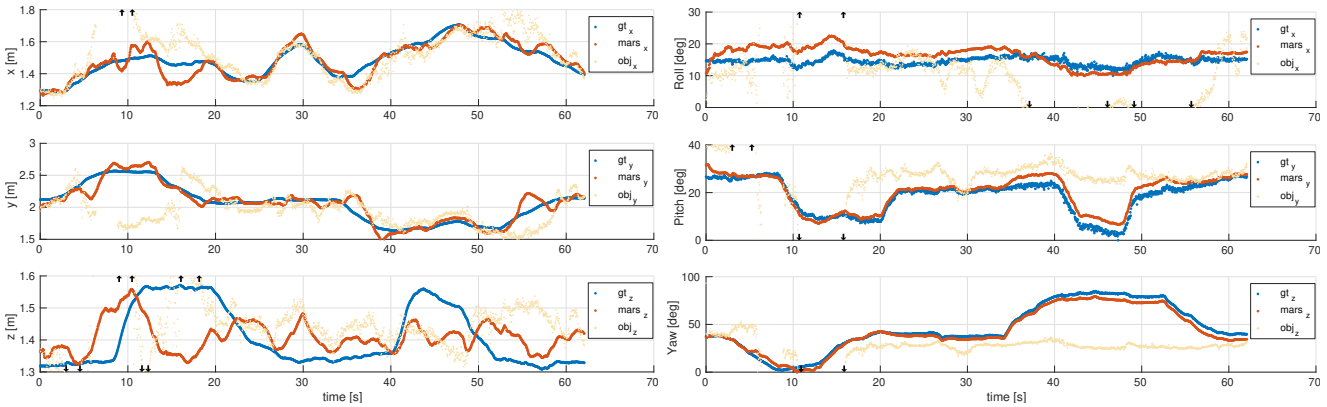

Fig. 3. Comparison of estimated position and orientation in Euler angles (mars) and the ground (gt) for run 8 of sequence 4. The components of the position (x, y, z) and orientation (roll, pitch, yaw) are plotted individually for the whole sequence. Additionally, we compare the reprojected IMU pose given the raw PoET estimates for object 3 (obj). The black arrows enclose a section in which the reprojected IMU pose is out of plotting range. Important to note, the object was not visible in the camera images between 6.4s and 8.8s.

deadreckoning. In such cases, a prediction of uncertainties for each object and measurement would lead to better results rather than working with the fixed values described above. Integration of aleatoric and epistemic uncertainties for the predictions of PoET is subject to future work.

To illustrate the reproducibility of our approach, we chose two out of the 8 sequences (sequence 4 and sequence 6) and repeated each sequence 10 times resulting in similar but not exactly the same trajectories. For each sequence, the RMSE across the whole trajectory for each run and the average RMSE and std across all runs are summarized in Table II and Table III, respectively. For both sequences, we are able to reproduce the performance of our method across 10 independent runs with a low standard deviation. This shows an AI-based component can be reliably incorporated into the state estimation of a robot.

Moreover, we compare the estimated and the ground truth position and orientation across the whole trajectory for an example recording in Fig. 3. This example shows that our approach reliably estimates the position and orientation for the whole duration of the recording. Furthermore, the graphs show that the raw measurements of a single object sometimes lead to a reprojected IMU pose that does not align with the ground truth trajectory. However, by fusing IMU information with pose measurements from multiple objects our method is able to reliably estimate the trajectory, despite outlier measurements. In addition to that, we show in Fig. 4 an example for the self-calibration capabilities of our approach with respect to the object-world states. The object-world is wrongly initialized after it was first observed due to a possible noisy measurement. Nonetheless, the state converges after 5 seconds.

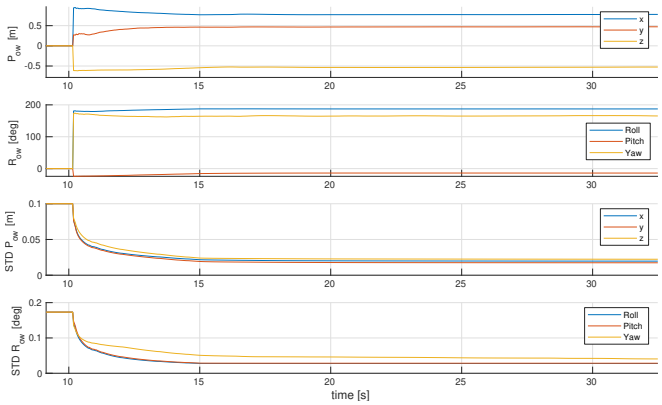

Fig. 4. Visualization of the estimated object-world state $(\mathbf{p}_{O_k W}, \mathbf{q}_{O_k W})$ and the corresponding state covariance represented by the std for a non-main object for run 8 of sequence 4. The position is split up into (x, y, z), while the orientation is represented by the Euler angles. The states are plotted from the point of time the object is first observed (at about 10s) until the states converge. At the beginning the object state is wrongly initialized due to perhaps a noisy measurement. However, after about 5 seconds the state converges and the uncertainty becomes minimal.

V. CONCLUSIONS

In this paper, we investigated object relative state estimation for mobile robots with an AI-based method to extract semantic information (object class and pose) from single RGB images. We defined a minimal sensor configuration, consisting of an RGB camera and IMU, and an experimental scenario in which object relative state estimation is required, mimicking the task of inspection and monitoring. We derived and implemented a filter-based solution for full state estimation of a mobile robot given 6-DoF relative pose measurements. Additionally, our method does not require any initial information about the global and relative poses of the objects. By defining object-world states, the coordinate frame of each object is estimated concurrently with respect to a common navigation world by using one of the objects as an anchor point. Our experiments with own real data showed that our method can be used for state estimation of the mobile robot in different scenarios and that the results can be reliably reproduced. Our results show that AI-based, semantic information from a single sensor is sufficient in combination with IMU data for accurate state estimation. Future work will consider incorporating aleatoric and epistemic uncertainties of the AI-based predictions in the sensor fusion framework for improved outlier rejection as well the integration of our proposed approach on a real UAV for closed loop experiments.

REFERENCES

[1] J. Crespo, J. C. Castillo, O. M. Mozos, and R. Barber, "Semantic information for robot navigation: A survey," *Applied Sciences*, vol. 10, no. 2, p. 497, 2020.

[2] S. Jordan, J. Moore, S. Hovet, J. Box, J. Perry, K. Kirsche, D. Lewis, and Z. T. H. Tse, "State-of-the-art technologies for uav inspections," *IET Radar, Sonar & Navigation*, vol. 12, no. 2, pp. 151–164, 2018.

[3] Y. Li, C. Fu, Z. Huang, Y. Zhang, and J. Pan, "Keyfilter-aware real-time uav object tracking," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 193–199.

[4] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, pp. 854–874, 2011.

[5] E. Allak, A. Hardt-Stremayr, and S. Weiss, "Key-frame strategy during fast image-scale changes and zero motion in VIO without persistent features," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2018.

[6] A. Giusti, J. Guzzi, D. C. Cireşan, F.-L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro, *et al.*, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2015.

[7] R. Dominguez, E. Onieva, J. Alonso, J. Villagra, and C. Gonzalez, "Lidar based perception solution for autonomous vehicles," in *2011 11th International Conference on Intelligent Systems Design and Applications*. IEEE, 2011, pp. 790–795.

[8] M. Hofmarcher, T. Unterthiner, J. Arjona-Medina, G. Klambauer, S. Hochreiter, and B. Nessler, "Visual scene understanding for autonomous driving using semantic segmentation," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019, pp. 285–296.

[9] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," in *Robotics: Science and Systems (RSS)*, 2018.

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

[11] J. Solà, J. Vallvé, J. Casals, J. Deray, M. Fourmy, D. Atchuthan, A. Corominas-Murtra, and J. Andrade-Cetto, "Wolf: A modular estimation framework for robotics based on factor graphs," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4710–4717, 2022.

[12] C. Brommer, R. Jung, J. Steinbrener, and S. Weiss, "MaRS: A modular and robust sensor-fusion framework," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 359–366, 2020.

[13] T. Moore and D. Stouch, "A generalized extended kalman filter implementation for the robot operating system," in *Intelligent autonomous systems 13*. Springer, 2016, pp. 335–348.

[14] Z. Cao, Y. Sheikh, and N. K. Banerjee, "Real-time scalable 6dof pose estimation for textureless objects," in *2016 IEEE International conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2441–2448.

[15] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-dof object pose from semantic keypoints," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 2011–2018.

[16] A. Amini, A. S. Periyasamy, and S. Behnke, "T6d-direct: Transformers for multi-object 6d pose direct regression," in *DAGM German Conference on Pattern Recognition*. Springer, 2021, pp. 530–544.

[17] G. Billings and M. Johnson-Roberson, "Silhonet: An rgb method for 6d object pose estimation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3727–3734, 2019.

[18] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1521–1529.

[19] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.

[20] Z. Li, G. Wang, and X. Ji, "Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7678–7687.

[21] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 611–16 621.

[22] C. Li, J. Bai, and G. D. Hager, "A unified framework for multi-view multi-class object pose estimation," in *Proceedings of the european conference on computer vision (eccv)*, 2018, pp. 254–269.

[23] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *European Conference on Computer Vision*. Springer, 2020, pp. 574–591.

[24] A. Krull, E. Brachmann, F. Michel, M. Y. Yang, S. Gumhold, and C. Rother, "Learning analysis-by-synthesis for 6d pose estimation in rgb-d images," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 954–962.

[25] F. Michel, A. Kirillov, E. Brachmann, A. Krull, S. Gumhold, B. Savchynskyy, and C. Rother, "Global hypothesis generation for 6d object pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 462–471.

[26] T. Jantos, M. A. Hamdad, W. Granig, S. Weiss, and J. Steinbrener, "PoET: Pose Estimation Transformer for Single-View, Multi-Object 6D Pose Estimation," in *Proceedings of the 6th Conference on Robot Learning*. PMLR, 2023.

[27] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1352–1359.

[28] N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang, "Symmetry and uncertainty-aware object slam for 6dof object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 901–14 910.

[29] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 1722–1729.

[30] J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "Visual servoing of quadrotors for perching by hanging from cylindrical objects," *IEEE robotics and automation letters*, vol. 1, no. 1, pp. 57–64, 2015.

[31] G. Loianno, V. Spurny, J. Thomas, T. Baca, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho, *et al.*, "Localization, grasping, and transportation of magnetic objects by a team of mavs in challenging desert-like environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1576–1583, 2018.

[32] K. Máthé, L. Buşoniu, L. Barabás, C.-I. Iuga, L. Miclea, and J. Braband, "Vision-based control of a quadrotor for an object inspection scenario," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2016, pp. 849–857.

[33] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[34] C. Brommer, C. Böhm, J. Steinbrener, R. Brockers, and S. Weiss, "Improved state estimation in distorted magnetic fields," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1007–1013.

[35] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.