

On-Board Vision-Based 3D Relative Localization System for Multiple Quadrotors

Duarte Dias^{1,2}

Rodrigo Ventura²

Pedro Lima²

Alcherio Martinoli¹

Abstract—This work proposes a novel relative localization system, based on active markers and an on-board camera, for tracking multiple quadrotors in a limited field of view. The system extracts the 3D poses of the markers including one that, by pulsating at a predefined frequency, provides a unique platform ID. We discuss how the camera field of view can be explored in presence of multiple targets, and what are the conditions on the system visibility that lead to the establishment of bidirectional sensing between robots with similar sensing capabilities. A visibility analysis is conducted to show that the developed relative localization system meets such requirements, and a closed-loop experiment is used to validate its performance under these conditions. Finally, its performance is compared with other results from the literature, and a metric is established with the intent of mapping different design solutions, facilitating design choices in presence of different requirements.

I. INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs) has been increasing, since their design has been simplified, enabling for smaller sizes, and the attitude stabilization techniques have been improved. Smaller size allows in turn for a more convenient exploration of multi-UAV systems in confined spaces, although at the cost of additional constraints in terms of on-board resources. Because of such possible miniaturization, UAV systems can also be used in environments without easy access to absolute localization information (e.g., indoors) or where GNSS-based technology simply does not provide enough accuracy for close range navigation.

Typical indoor localization solutions leverages off-board motion capture systems [3], [4], which might not be always feasible or desired, as they involve a deployment of potentially expensive infrastructure before any operation using the UAV system. In other approaches, each robot autonomously acquires its position in the environment using Simultaneous Localization and Mapping (SLAM) or Monte Carlo Localization (MCL) techniques, as in [1], [2]. However, for multi-robot coordination, those positions need to be shared between robots, so they can compute the relative positions of their neighbors. This can become unfeasible when high communication throughput is necessary to cope with the high

dynamics of the systems at hand. Relative sensing solves this issue, as this information can be collected without the use of communication channels. Additionally, the extracted information can be more robust to disturbances from the environmental background, since it is possible to engineer precisely the dedicated features to be sensed on each vehicle.

Relative localization for AUVs has been demonstrated using several physical channels, such as sound [16] or infrared [17]. However, the most mature approaches use camera systems, extracting range and bearing information based on the object size and position in the image. The simplest methods use single blob detection [11], [12], [5], which can be strongly dependent on the lighting conditions, shadows, or partial occlusions by the 3D structures, making range estimations unreliable. Although satisfactory results can be achieved using circle matching techniques [13], [12], [7], bearing information is still mainly preferred. This is the case of the formation control algorithms in [12], [5], which have to use additional information sharing and triangulation methods to allow formation control under these limitations. Also, robot IDs has to typically be extracted from colored markers, a technique that has limited scalability.

However, range accuracy and blob detection reliability can be improved using active multi-marker platforms. This work is based on a similar method to that reported in [6], where the full robot pose is extracted using the P3P algorithm. Such method has lead to high precision measurements, for large distance ranges, as reported in [9], [10]. As in [8], inertial measurements are also fused with the relative sensing data to obtain improved estimation and control performance. However, previous works consist on single platform observations. This work tackles the problem of multi-vehicle relative sensing, and studies the impact of the limited camera Field Of View (FOV), including the corresponding visibility restrictions. The system also implements an ID-recognition algorithm based on a pulsating marker to avoid using different configurations of the marker 3D layout, which hampers the platform design and adds computational complexities. Finally, this work proposes a metric to characterize the system performance according to the chosen design, with the intent of mapping the performance of different solutions, simplifying future design choices when facing different requirements, for example in terms of FOV or range. This work is developed using CMOS cameras, but most of the concepts can be applied on Dynamic Vision Sensors (DVS) [15], especially with the results in [14], which allows the system to distinguish active markers from the environment.

The paper starts by providing a description of the developed

*This work has been partially supported by ISR/LARSyS Strategic Funds from FCT [UID/EEA/5009/2013] and FCT SFRH/BD/51928/2012.

¹Duarte Dias (duarte.dias@epfl.ch) and Alcherio Martinoli (alcherio.martinoli@epfl.ch) are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne, Switzerland.

²Pedro Lima (pal@isr.tecnico.ulisboa.pt), Rodrigo Ventura (rodrigo.ventura@isr.tecnico.ulisboa.pt) and Duarte Dias are with the Institute for Systems and Robotics, IST, Universidade de Lisboa, Portugal.

hardware of the relative localization system in Section II, followed by a description of the dedicated tracking algorithm in Section III, which includes the definition of the performance metric and the considered sensor noise model. In Section IV experiments analyze the accuracy of the sensor measurements, and validate the proposed performance metric. Additionally, a visibility study on the developed localization system is performed, and multi-robot tracking is validated with a closed-loop experiment using two observed quadrotors. Finally, the work concludes with some remarks in Section V.

II. HARDWARE

The designed hardware platform, described in Fig. 1, is composed by a set of five omni-directional active markers of 2 cm radius, emitting light through a plastic diffuser. These markers are separated into two groups: four markers are used for the relative 3D pose extraction of the platform, emitting continuous light; one marker has pulsating capabilities, enabling a specific ID extraction by measuring the marker pulsating frequency. Regarding the first group, two active markers (M1 and M2) are placed on the quadrotor horizontal axes, and one (M3) at the vertical axis. The camera is placed in the vertical axis in between the vertical marker and the quadrotor frame origin, and it points in the direction shown in the Fig. 1 (b), considered to be the platform front. The length of the horizontal axes (24 cm) was chosen to be different from the length of the vertical axis (20 cm) to avoid platform symmetries. The last marker (M4) is placed below the camera and moved away from the vertical axis in the same direction of the camera's heading, but without entering its vertical FOV. The ID marker was placed below the previous marker, as shown in Fig. 1, in a location that is not important for localization.

The positions of the markers and the camera were designed so that the platform could rotate more than the FOV of the camera without generating any marker occlusions that could compromise its visibility to other sensors observing it. This visibility condition gives a quadrotor, equipped with this platform, the ability of adjusting its direction to optimize its FOV when tracking multiple targets, without compromising the possibility of maintaining bi-directional sensing connection with its neighboring robots. Note that this condition is often a requirement in multi-robot cooperation scenarios. On Section IV a study of the platform visibility shows that this platform meets these requirements.

Both marker groups have different lighting properties, to allow a clean decoupling between the ID extraction and platform localization. There are several advantages for this separation. Firstly, the relative localization accuracy and bandwidth are not affected by a temporary loss of the pulsating marker. Secondly, the platform ID is not generated using different geometrical configurations, which would require a careful choice of marker positions to prevent possible ID misclassification, and would increase the computational complexity of the used classification algorithms with the number of IDs. Finally, as discussed later, this additional

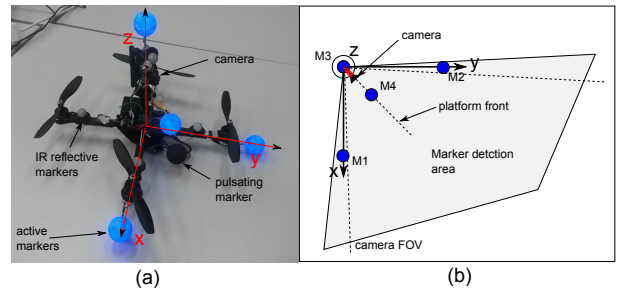


Fig. 1. Illustration of the sensing platform. (a) Complete set with camera and the active markers; (b) conceptual top view of the structure.

marker also allows for additional filtering capabilities, providing a way to confirm the existence of the observed platform. However, the initial ID extraction phase takes some time, which increases as the time used between marker pulses increases to allow for possible additional IDs. In this work, the localization markers emit blue light, but different lighting properties can be chosen, as for example infra-red (IR) used in [6], [8], or the Active Led Markers (ALM) on [14] in a DVS system. To increase the marker detection reliability in brighter scenarios, blue paint was added to take advantage of the reflections of incoming light on the markers. The ID marker emits red light, as it was found to be well detected by the cameras on the UAVs. A smaller layer of black paint was added on top to prevent external light in bright environments from overshadowing the marker emission.

III. PLATFORM TRACKING

Each quadrotor is equipped with the previously described platform. A tracking software runs on-board each quadrotor i , to extract the relative 3D position, velocity, and attitude, along with the ID, of other platforms observed using the on-board camera. Fig. 2 gives a description of the process. First, an algorithm associates a set of image blobs to a potential platform j observed in i 's camera frame. A relative position and orientation of the platform, \mathbf{x}_{ij}^c and \mathbf{R}_{ij}^c , is computed to justify the image blobs as 2D projections of the platform active markers in the camera image. The platform ID is extracted from the frequency of the pulsating marker. After the detection phase, measurements related to platform j are converted to a flying frame, to eliminate perturbations from quadrotor i roll (ϕ_i), and pitch (θ_i) inclinations, during horizontal movements. This frame, described in Fig. 3, is centered in the quadrotor i position, and has the same yaw (ψ_i), but its z axis is aligned with gravity (vertical), making its roll and pitch equal to zero. The resulting position measurements are filtered with an Extended Kalman Filter (EKF) to obtain accurate estimation of the 3D position, \mathbf{x}_{ij}^e , and velocity, \mathbf{v}_{ij}^e , which compose the state variables of the estimator for each platform j , defined in the flying frame. The next sections describe each part.

A. 3D Relative Pose Extraction

To obtain platform poses from the image, a blob detection algorithm, performed in all the image, identifies potential observed markers. Blobs with a size smaller than a certain

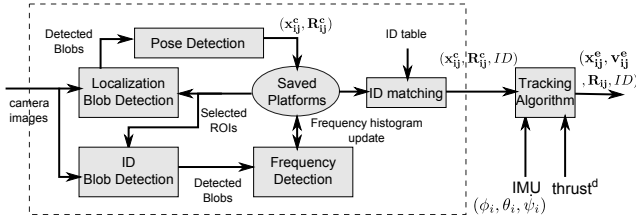


Fig. 2. Overall system architecture and information flow, for a specific quadrotor i . Sensor measurements are extracted from the camera image and fed into a tracking system block for velocity estimation and noise filtering. The block “Saved Platforms” saves the relative 3D pose and pulsating marker state of potential platforms observed on each image. This allows to compute the pulsating marker frequency across images for ID extraction, and to reduce the algorithm complexity on each image through the establishment of ROIs for each platform.

threshold are discarded. With the resulting blob set, all combinations of four blobs are tested as a potential platform j . Since the problem is combinatorial, additional pruning methods are performed to discard in advance wrong combinations.

For each accepted marker group j , an association between the image markers with the respective 3D platform marker positions is performed. All association combinations within each group of four markers are attempted. For each combination, three of the markers are fed into the P3P algorithm, described in [9], that is able to obtain four possible solutions for \mathbf{x}_{ij}^c and \mathbf{R}_{ij}^c justifying the current association. The fourth marker is then used to select the right solution and to evaluate its correctness, by comparing the distance error in pixels between the 2D positions of the observed marker in the image and the projection of the 3D marker associated to it in the image, computed using the transformation of the solution that is being tested. Marker configurations with an error bigger than a threshold are not considered. Valid configurations are saved with a local temporary identifier while the ID extraction is not concluded. Their extracted 3D poses are used to define a region of interest (ROI) and predict the blob positions for a more efficient detection in the next images.

B. ID Extraction

The saved marker configurations are then subjected to an ID extraction process, using the additionally fifth active marker with pulsating capabilities. A new blob detection procedure is launched for each currently saved configuration to detect blobs with the lighting properties chosen for these markers. However, since at this stage there is already an estimate for the platform pose, the procedure is done in a small ROI, with the center around the predicted position for the marker projection in the image, and its size computed from the predicted distance between the localization markers and the camera. Again, a threshold is used to remove blobs that are too small.

The platform ID is associated to a unique pulse frequency. To detect these frequencies, the time between two marker pulses is noted and used to create a pulse frequency histogram. This histogram is used to compute an average pulse frequency, which is then matched with a potential corresponding platform ID. Only relevant frequencies, defined by being recorded more than a few times (5 times was used in this work), are

considered to compute the average. This has the advantage that any false positive generated by the pose extraction will most likely be filtered out. Also, even if clutter affects the pulse frequency, the created histogram avoids losing the correct ID. The ID initialization takes the time needed to observe at least one relevant frequency. But after this period, ID information is always available.

C. Tracking Algorithm

Each measurement related to an observed platform j is forward to a tracker. This tracker keeps a list of platforms, each described by its ID, its estimated position and velocities states $(\mathbf{x}_{ij}^e, \mathbf{v}_{ij}^e)$, and its attitude \mathbf{R}_{ij} in the quadrotor i 's flying frame. The filter tracks each platform independently at discrete time intervals of Δt seconds. Platform j states are predicted at step k , using the desired vertical thrust commands currently issued to the lower level control combined with the current attitude measurements provided by the on-board inertial sensors, ϕ_i , and θ_i , to predict the linear acceleration felt in the quadrotor flying frame, \mathbf{u}_i , at that step. The angular velocity around the z axis, $\dot{\psi}_i$, is also extracted and used as yaw rate. Roll and pitch rotations are not considered since the states are described in the flying frame. Note that the inertial sensor measurements are considered as noise-free. With this information, it is possible to define a motion model for the motion of platform j relative to quadrotor i 's flying frame:

$$\begin{bmatrix} \mathbf{x}_{ij}^e(k+1) \\ \mathbf{v}_{ij}^e(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \Delta t \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{ij}^e(k) \\ \mathbf{v}_{ij}^e(k) \end{bmatrix} - \begin{bmatrix} 0.5 \Delta t^2 \mathbf{I}_3 \\ \Delta t \mathbf{I}_3 \end{bmatrix} \mathbf{u}_i(k) + \begin{bmatrix} \mathbf{x}_{ij}^e(k) \times \boldsymbol{\Omega}_i(k) \\ \mathbf{v}_{ij}^e(k) \times \boldsymbol{\Omega}_i(k) \end{bmatrix} + \mathbf{w}_m(k), \quad (1)$$

where \mathbf{I}_3 , $\mathbf{0}_3$ are 3x3 identity and zero matrices, and $\boldsymbol{\Omega}_i$ is the vector $(0, 0, \dot{\psi}_i)$, which provides a yaw rotation of \mathbf{x}_{ij}^e and \mathbf{v}_{ij}^e for a non-zero yaw rate. This term uses directly the angular velocity from the z axis of the quadrotor, which makes the model only valid with small values of roll and pitch (otherwise a transformation would have to be applied in order to convert the angular velocity to the true yaw rate on the flying frame). But in this work it is assumed that those conditions are met, which makes sense since usually the quadrotors do not need to tilt more than a few degrees to achieve good reaction times. The last term, $\mathbf{w}_m(k)$, is the movement noise simply considered as a zero mean Gaussian distribution with a covariance profile that tries to encapsulate the noise of all the previous terms combined.

For each measurement of platform j generated by the image processing algorithm, the tracker updates the 3D relative position of the respective platform on the list. First, the measurement is converted to the flying frame, using the known camera pose in quadrotor i body frame, and the current ϕ_i and θ_i values. Platform j 's position in the flying frame is then updated with the following observation model:

$$\mathbf{z}_{ij}(k) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{ij}^e(k) \\ \mathbf{v}_{ij}^e(k) \end{bmatrix} + \mathbf{w}_o(k), \quad (2)$$

where $\mathbf{z}_{ij}(k)$ is the vector containing the 3D position measurement of platform j relative to quadrotor i , and $\mathbf{w}_o(k)$

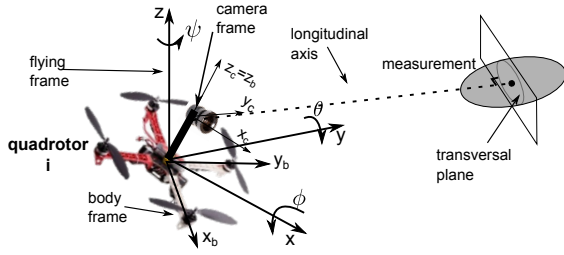


Fig. 3. Frames considered in the estimation process. The flying frame is centered at the quadrotor position, with the same yaw, but has its z -axis aligned with gravity (vertical). Observation noise is shown by the gray area, modeled with a longitudinal axis and a transversal plane, computed with respect to the camera frame, and afterwards transformed to the flying frame for estimate updates.

is the noise vector, with zero mean. The covariance of this noise is further explained in the next section.

It is important to consider the case when the platform leaves the camera FOV for a sufficient amount of time, causing the sensor to drop the platform. Pose and ID detection has to be reinitialized on FOV reentry. To avoid a potential interruption on the platform tracking in this period, the tracked platform is set into a lost mode, which allows platform measurements with unassociated IDs that are sufficiently close to the current estimated location, to be accepted as valid measures. The situation returns to normal as soon the ID initialization is finalized. The complete tracking system, as it is, does not deal well with marker occlusions (a study of the visibility is performed in the next section), but it could be easily extended with least square methods [6], or a particle filter [14] approach to tackle this issue. However, this factor is not considered to be the main focus of this work.

D. Sensor Noise

To characterize the sensor noise, the measurements extracted from the image processing algorithm are divided into range and bearing components. After a proper calibration of the system, the noise can be assumed to be produced solely by instant irregularities on the vision sensor (e.g., blur effects). Considering the pin-hole camera model in one dimension, the edges of a line segment of size l , parallel to the camera plane, at a range z from the camera focal point, and centered with its principal axis, have a size in the image p , of:

$$p = \frac{lf}{z}, \quad (3)$$

where f is the camera focal length (in this work f is defined in pixels). Differentiating Eq. (3) in respect to z and l , and rearranging the terms, it is possible to derive the range, δz , and bearing, δl , errors:

$$\delta z = -\frac{z^2}{lf} \delta p, \quad \delta l = \frac{z}{f} \delta p \quad (4)$$

where δp is an error on the observed segment length caused by pixel noise in the image. Note that this result is general for any position of the line segment around the camera. Also, it is possible to define f as:

$$f = \frac{N}{2 \tan(\alpha_M)}, \quad (5)$$

where N is the camera resolution (number of pixels), and α_M is half of the camera FOV. Therefore one can observe that the range error progresses faster with the range from the camera than the bearing error. Also, it depends on the object size, which is usually small. For this reason, the sensor noise is assumed to be mostly in the range component, and a transversal-longitudinal model is used for its covariance, represented by an ellipse with its major (longitudinal) axis pointing to the origin of the camera frame, and the other axis belong to the transversal plane, perpendicular to that axis, as shown in Fig. 3. The axes lengths are set to evolve linearly with the range from the sensor, as a simplified representation of the error dependency with range.

Additionally, from Eq. (4) it is possible to predict the range error δz , in function of z , given the knowledge of δp , l , and f . The bearing error is discarded since it is assumed to be small. Here, f is taken from the camera intrinsic parameters, and l from the diameter of the smallest sphere that can encapsulate the marker configuration, also called the object circumsphere. The marker configuration developed in this work has a circumsphere of 28 cm in diameter. The value of δp in this work is considered to be the maximum pixel error that can be observed in the image, and it can be computed from experiments that will be explained on the results section. These predictions can provide a metric for the maximum errors that can occur with a certain camera and marker configuration, providing a way to easily map desired performances across different design choices.

This result is considered to be an approximation for two reasons. Firstly, object rotations can produce errors not necessary related to the object size. Secondly, common distortion effects, such as radial distortion, alter this relationship, since the value of α_M changes (usually increases) as the object moves away from the camera principal axis, altering f .

IV. EXPERIMENTS

To evaluate the system performance, a set of experiments were conducted with the aid of a Motion Capture System (MCS), shown in Fig. 4, to provide ground truth millimetric and sub-degree accuracy of the full quadrotor poses. Sensor data were validated by comparing the sensor measurements with the MCS poses, both recorded on-board the vehicle for straightforward synchronization (MCS measurements were communicated to the quadrotor with a latency of about 5 ms, considered negligible for synchronization given the low speed of the vehicle maneuvers). A low resolution camera with 320x240 pixels and 90° FOV was used for the experiments. With this configuration, platform tracking could be done at an average of 20 Hz observing one quadrotor, and 17 Hz observing two quadrotors, with all functions running on-board. Note that the comparison between ground truth and sensor data revealed a sensing lag of around 150 ms. The considered maximum range of the sensor was 3 m.

The camera pose in the quadrotor body frame and its intrinsic parameters were manually calibrated, by overlapping 3D position readings from the sensor and the MCS. This was achieved by manually moving a quadrotor equipped

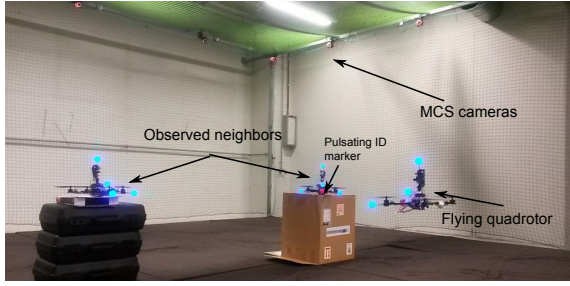


Fig. 4. Our MCS is manufactured by Motion Analysis Inc. and is composed of 20 Osprey cameras able to track a set of at least four reflective markers in an useful volume of $4 \times 7 \times 2.5$ meters.

with markers in front of the on-board sensor, at a certain distance. The quadrotor had a constant orientation during these trajectories. The camera's focal length, principal point and radial distortion parameters were considered. The parameters were divided into vertical and horizontal. The radial distortion is modeled, for each dimension, with a second degree polynomial model.

A. Relative Position Evaluation

The accuracy of the 3D relative positions extracted throughout the sensor FOV was assessed with experiments similar to the calibration procedure, where the moving quadrotor had the same orientation as the one used during calibration. Fig. 5 shows the norm of the 3D position error as a function of the relative distance between the observed platform and the camera sensor. The used dataset contained 1063 measurements, conducted with a relative distance of 1 to 3.5 m. Only 8 outliers were registered outside the 99.7% quantile of the computed data distribution, most of them with angles between 25° and 30° in the camera FOV.

To evaluate the estimator and the longitudinal-transversal noise model described above, the quadrotor was flown in a plane, 3 m from the camera in the y direction of the frame described in Fig. 8 (a). The quadrotor performs on that plane a set of vertical and horizontal movements, as described in Fig. 6 (a). The error covariance of the extracted sensor data was computed for the longitudinal and transversal components, which showed values of $(0.13 \text{ m})^2$ and $(0.03 \text{ m})^2$ respectively. This confirms the small bearing error that was assumed in this work. The respective axes of the covariance ellipse were set to respect the assumed linear relationships with the range to the sensor, and they were fitted so they would have, at a 3 m range, a covariance of

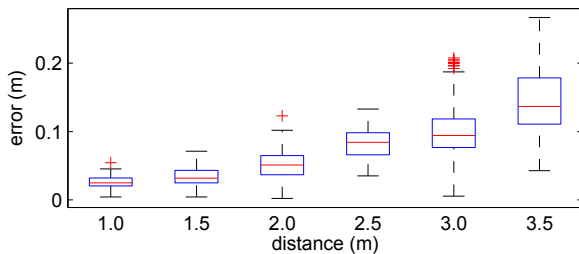


Fig. 5. 3D position sensor error of the tracked marker platform, in function of its absolute 3D distance to the static sensor.

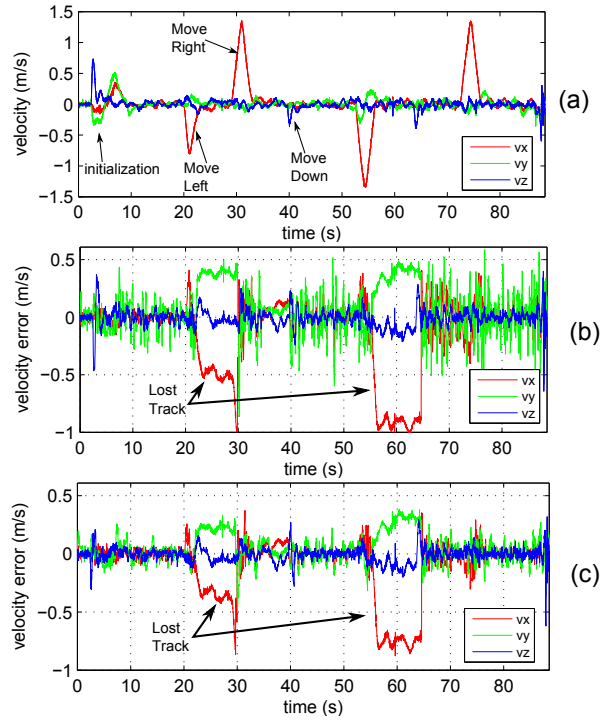


Fig. 6. Velocity estimation performance at a distance of 3 meters from the camera sensor. (a) The ground truth measurement given by the MCS. (b) Estimation error using the spherical sensor model. (c) Estimation error using the longitudinal-transversal sensor model. Note the regions where the tracking was lost.

$(0.15 \text{ m})^2$ for the longitudinal axis, and of $(0.03 \text{ m})^2$ for the transversal axes. The sensor data are then fed into the estimator described above, running at 40 Hz (setting Δt at 0.025 s). The attitude measured from the on-board inertial sensors and the desired thrust commands (in this case it has zero thrust because the sensor is static) were also provided. Fig. 6 shows the results of the velocity estimation error for a testing trajectory, comparing the longitudinal-transversal model with a simple spherical model with a covariance of $(0.15 \text{ m})^2$. The position estimations are not shown as the raw sensor data has already low noise. It is observed that the considered model drastically improves the velocity errors on the transversal dimension (in this case the y dimension).

The maximum errors are then predicted from Eq. (4) for the range, with f of 220 pixels, found during calibration, and l of 28 cm. To compute δp in function of the distance to the sensor, a set of experiments were performed, where a platform was placed in front of a camera sensor at different distances, and was shaken while its blob positions and their distances were being tracked. The value of δp was set to be half of the average of the distance fluctuation observed for each blob pair, since the error is assumed to fluctuate around the correct value. This values seemed to vary little, around 2.5 pixels for the considered distances of 1 to 3.5 m. This is because active markers produce image blobs that only change on the borders, which varies less with distance. Fig. 7 shows a comparison between the predictions obtained with the previous values, and the maximum errors of the distributions found for the accuracy performance, displayed

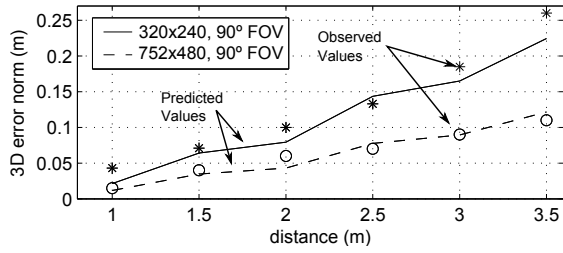


Fig. 7. Comparison of the maximum accuracy errors obtained using our setup, and the setup in [6]. The actual and predicted values are matched to assess the correctness of the accuracy metric.

in Fig. 5, considered to be inside the 99.7% quantile.

This work was also compared to the results in [6], which uses a higher resolution camera, with 752x480 pixels and 90° FOV, resulting in a f value $752/320 = 2.35$ times bigger. The values for δp were assumed to be the same as the ones computed in this work, as an active-marker localization system was also used, but the same experiments should be carried out on that system to be more accurate. They also provide a circumsphere with 22 cm diameter for their marker set, which was used for the l value. The maximum values of the error were taken from their boxplots in Fig. 7 (a), and matched against the predicted values. From Fig. 7, both works follow well the behavior of predicted values, although this work has measured values bigger than predicted. This could be explained by the distortion of the camera that affects the value of α_M in Eq. (5), with increased effect as tests are done closer from the FOV edges. As previously observed, the used datasets contained positions sufficiently far from the camera center for this to happen. From the previous results it is possible to conclude that the metric is a good indicator of the system limitation, and that accuracy of both systems have a comparable performance within their own design limitations.

B. Relative Attitude and Visibility Evaluation

The 3D attitude accuracy is assessed by applying rotations around the x , y , and z axis of a quadrotor with a marker platform, in several static positions with respect to the camera sensor, as shown in Fig. 8 (a). The attitude error is expressed in the axis-angle form, and the obtained angle is used for the error. Fig. 8 (b) shows these errors as a function of the distance between the observed platform and the camera sensor, for a dataset containing 1229 measurements, conducted between 1 and 2.5 m from the camera and throughout its FOV area. The angle errors are small but remain roughly the same also at larger ranges. Also, Fig. 8 (a) shows position errors up to 10% of the distance, evolving as the platform is rotated on the spot. This fact is associated to a calibration inaccuracy of either the platform geometries or the camera parameters, but is not explored further.

By leveraging the same rotational experiments mentioned above, we assessed the relative visibility of our solution. In particular, we observed the maximal rotation point until the camera sensor lost its pose due to occlusions between the markers and the quadrotor structure. In this case, we only considered yaw rotations, since quadrotors usually operate

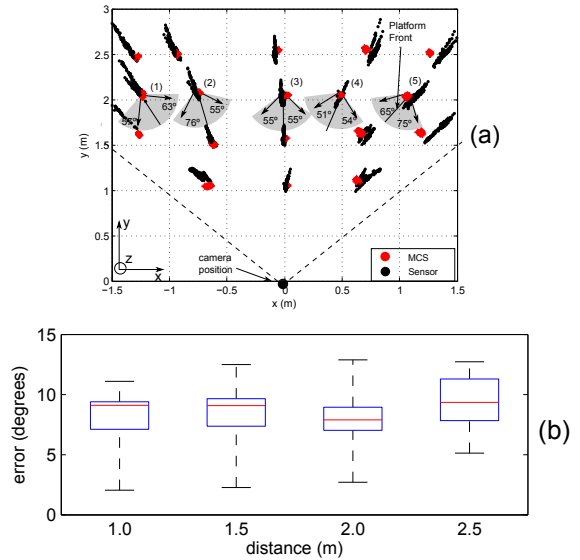


Fig. 8. 3D orientation validation. (a) Different positions used to perform rotations, and yaw rotation bounds from the platform front. (b) Relative attitude error of the tracked marker platform in function of relative 3D distance to the sensor, expressed in the axis-angle form.

with small roll and pitch angles. The experiments were done observing the platform 50 cm from above and below. Fig. 8 (b) shows the rotation bounds for the five highlighted locations, representative of the FOV coverage. In this case the platform was observed from below. Using all the data, it was revealed that the observed platform can rotate at least $\pm 90^\circ$ from the platform front when observed from above, and $\pm 45^\circ$ when observed from below. The bounds are smaller in the latter case as the middle marker (M4) is more easily occluded. Note that, in Fig. 6, the sensor loses track while still in these bounds. This happens at higher distances, due to a threshold of the blob detection algorithm that merges blobs that are close in the image. This could be solved if this threshold is made adaptive to the sizes of the blobs that are being merged.

These previously discussed bounds are above the FOV bounds of the camera ($\pm 45^\circ$ from the platform front). Therefore the platform meets the visibility requirements mentioned in Section II, allowing bi-directional sensing between itself and its neighbors while freely exploring its full camera FOV. The next experiment shows one such case.

C. Closed-Loop Validation

The system was tested in a closed-loop control experiment, illustrated in Fig. 4, where a flying quadrotor had to maintain a desired relative horizontal and vertical distance in respect to two static neighbors, using a proportional and derivative controller. The algorithm also adjusted the quadrotor yaw orientation on the fly in order to even both observed targets on each side of the sensor FOV. The quadrotor is first teleoperated into its initial position, using the MCS position feedback. When activated, the algorithm runs exclusively on-board, using just the camera sensor as sensory feedback, until a stop and land command is specified. The chosen ID marker frequencies were chosen to be 10 Hz for the flying quadrotor, 5 Hz, and 3.3 Hz for the static neighbors.

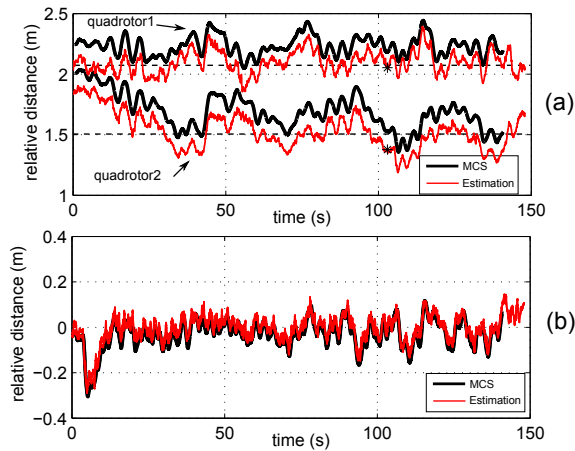


Fig. 9. Comparison between on-board estimation and the MCS values for the horizontal (a) and vertical (b) distances of the quadrotor to neighbors. Vertical distances are too close together to be distinguished. Target horizontal distances for each platform are shown with dashed lines. Star-shaped points shown in (a) represent zones where the sensor took more than 100ms to provide a measurement for a given platform.

Fig. 9 compares the MCS measures and the on-board sensor estimates of the evolution of the target distances, during a run of this task. The desired vertical distance was set to zero. The desired horizontal distances were 2.12 m and 1.5 m (see Fig. 9). The angle between the two neighbors and the flying quadrotor on the reported experiment was 45° . The experiment was tested up to 60° , where control fluctuations already made it difficult to hold both neighbors within the FOV. Besides from a small, almost constant (17 cm), mismatch between the estimated and the actual horizontal distances, the closed-loop control appeared to follow well the desired distances. The mismatch can be explained by the calibration errors observed in Fig. 8 (a). The higher oscillations (maximum of 25 cm) observed for the horizontal distances can be explained by of the quadrotor dynamics for horizontal movements, and the fact that camera transversal noise is higher than the tangential noise.

Besides the ID initialization period (not shown here but the maximum highest value was 1.5 s to track the 3.3 Hz frequency), the quadrotor received regular sensor measurements from both neighbors. There is just one point (see Fig. 9) where the measures stopped for more than 100 ms, but since it was for both neighbors simultaneously, the problem was attributed to sensing frequency fluctuations.

V. CONCLUSION

This work proposes a system capable of providing 3D relative pose measures and object ID for multiple platforms simultaneously within the sensor FOV. The experiments showed that the developed system has the capability to establish bi-directional sensing between robots with similar sensing capabilities, without compromising the possibility of exploring the sensor FOV in multi-robot cooperation scenarios. Despite the low sensor accuracy, the tracking system with the proposed noise model was able to track well the position and velocity of the observed quadrotors, allowing the stabilization of a second order system without the help of external systems.

Additionally, a metric was proposed to evaluate the sensor accuracy, and it was shown how to be used to map the performance between different designs. As future work, the tracker will be extended to mitigate the impact of marker occlusions. Also, a dedicated study will be performed on the camera borders in order to maximize the usability of the available FOV. Finally, the results will be exported to systems with higher FOV, making use of the developed metric.

REFERENCES

- [1] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor," *Robotics: Science and Systems*, 2013.
- [2] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M.W. Achtelik, M. Chl, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, H. Gim, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, and L. Meier, "Vision-Controlled Micro Flying Robots: from System Design to Autonomous Navigation and Mapping in GPS-denied Environments," *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26-40, 2014.
- [3] M. Turpin, N. Michael, and V. Kumar, "Trajectory Design and Control for Aggressive Formation Flight with Quadrotors," *Autonomous Robots*, vol. 33, no. 1-2, pp. 143-156, 2012.
- [4] F. Augugliaro, A. P. Schoellig, R. D'Andrea, "Generation of collision-free trajectories for a quadcopter fleet: A sequential convex programming approach," *International Conference on Intelligent Robots and Systems*, pp. 1917-1922, 2012.
- [5] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. Bülthoff, and P. Giordano, "Modeling and control of UAV bearing formations with bilateral high-level steering," *International Journal of Robotics Research*, vol. 31 no. 12, pp. 1504-1525, 2012.
- [6] M. Faessler, E. Mueggler, K. Schwabe, and D. Scaramuzza, "A Monocular Pose Estimation System based on Infrared LEDs," *IEEE International Conference on Robotics and Automation*, 2014.
- [7] M. Achtelik, T. Zhang, K. Kiihnlitz, and M. Buss, "Visual Tracking and Control of a Quadcopter Using a Stereo Camera System and Inertial Sensors," *International Conference on Mechatronics and Automation*, pp. 2863-2869, 2009.
- [8] M. Cutler, B. Michini, and J. How, "Lightweight Infrared Sensing for Relative Navigation of Quadrotors," *International Conference on Unmanned Aircraft Systems*, pp. 1156-1164, 2013.
- [9] L. Kneip, D. Scaramuzza, and R. Siegwart, "A Novel Parametrization of the Perspective-Three-Point Problem for a Direct Computation of Absolute Camera Position and Orientation," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2969-2976, 2011.
- [10] A. Breitenmoser, L. Kneip and R. Siegwart, "A Monocular Vision-based System for 6D Relative Robot Localization," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 79 - 85, 2011.
- [11] S. Roelofsen, D. Gillet, A. Martinoli, "Reciprocal Collision Avoidance For Quadrotors Using On-board Visual Detection," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4810-4817, 2015.
- [12] R. Tron, J. Thomas, G. Loianno, J. Polin, V. Kumar, and K. Daniilidis, "Vision-based formation control of aerial vehicles," *Robotics: Science and Systems*, 2014.
- [13] T. D'Orazio, C. Guaragnella, M. Leo, A. Distanti, "A new algorithm for ball recognition using circle Hough transform and neural classifier," *Journal of Pattern Recognition*, vol. 37, no. 3, pp. 393-408, 2004.
- [14] A. Censi, J. Strubel, C. Brandli, T. Delbruck, and D. Scaramuzza, "Low-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor," *International Conference on Intelligent Robots and Systems*, pp. 891 - 898, 2013.
- [15] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120 dB 15 us Latency Asynchronous Temporal Contrast Vision Sensor," *IEEE Journal of Solid-State Circuit*, vol. 43, no. 2, 2008.
- [16] M. Basiri, F. Schill, D. Floreano, and P. U. Lima, "Audio-based Localization for Swarms of Micro Air Vehicles," *IEEE International Conference on Robotics and Automation*, pp. 4729 - 4734, 2014.
- [17] J. Roberts, T. Stirling, J. Zufferey, and D. Floreano, "3-D relative positioning sensor for indoor flying robots," *Autonomous Robots*, vol. 33, no. 1, pp. 5-20, 2012.