MOMA: Visual Mobile Marker Odometry

Raul Acuna[1], Zaijuan Li[1] and Volker Willert[1]

Abstract— In this paper, we present a cooperative odometry scheme based on the detection of mobile markers in line with the idea of cooperative positioning for multiple robots [1]. To this end, we introduce a simple optimization scheme that realizes visual mobile marker odometry via accurate fixed marker-based camera positioning and analyse the characteristics of errors inherent to the method compared to classical fixed marker-based navigation and visual odometry. In addition, we provide a specific UAV-UGV configuration that allows for continuous movements of the UAV without doing stops and a minimal *caterpillar*-like configuration that works with one UGV alone. Finally, we present a real-world implementation and evaluation for the proposed UAV-UGV configuration.

I. INTRODUCTION

Visual pose estimation and localization is a problem of interest to many fields from robotics to augmented reality and autonomous cars. Possible solutions are dependent on the camera(s) configuration available to the task (monocular, stereoscopic or multi-camera), as well as the amount of knowledge about the structure and geometry of the environment.

If a mobile multi-robot system is available, cooperative localization firstly introduced by Kurazume et al. [1] drastically speed-up and improve the accuracy of the localization of each of the robots [2], [3]. The original idea is to use some robots as moving landmarks and others to detect them. This allows a mobile robot-marker system to localize itself in an unstructured environment lacking enough features. A bunch of different realizations [4] and extensions to multiple-robot *SLAM* (simultaneous localization and mapping) have been investigated [5].

Visual pose estimation can be classified into two different categories: The first one, called marker-based (*MA*), relies on some detectable visual landmarks like fiducial markers or 3D scene models with known coordinates of its features/keypoints [6], [7]. The second category works markerless (*MAL*) without any 3D scene knowledge [7], [8].

MA methods estimate the relative pose to a marker with known absolute coordinates in the scene. Therefore, these methods are driftless, need only a monocular camera system, and the accuracy of the pose estimation is both dependent on the accuracy of the measurement of 2D image coordinates of known 3D marker coordinates and on what kind of algorithm is used to realize spatial resection [9], [10].

MAL methods estimate relative poses between camera frames based on static scene features with unknown absolute coordinates in the scene and apply dead reckoning to reach the absolute pose within the scene in relation to a known initial pose. Due to this incremental estimation, errors are introduced and are accumulated by each new frame-to-frame motion estimation, which causes unavoidable drift. These methods can further be divided into pure visual odometry (*VO*) [8] and more elaborate visual simultaneous localization and mapping (*V-SLAM*) approaches [11] including the new developments on Semi-Dense visual odometry [12]. Basic *VO* approaches estimate frame-to-frame pose changes of a camera based on some 2D feature coordinates, their optical flow estimates [13] and their 3D reconstruction using epipolar geometry in conjunction with an outlier rejection scheme to verify static features [14]. Even if some additional temporal filtering like extended Kalman filtering (*EKF*) or local bundle adjustment (*BA*) is applied, drift can be reduced but cannot be avoided [8].

V-SLAM approaches [11] not only accumulate camera poses but also 3D reconstructions of the back-projected extracted 2D features of *VO* in a global 3D map. Thus, drift can be reduced using additional temporal filtering on the 3D coordinates of the features in the map or global *BA* and *loop closure* techniques to relocate already seen features via map matching. Both approaches can be realized with a monocular or a stereo vision system, whereas the stereo approach is much less prone to drift because of the superior resolution of scale estimates. Alternatively, additional sensors like IMU can be integrated to improve the scale/drift problem in monocular systems and apply sensor fusion to increase robustness and reduce the drift as in Visual Inertial Odometry approaches [15].

The main advantage of *MA* versus *MAL* methods (besides the fact that it does not drift) is the knowledge of error free 3D coordinates of easy and unambiguously detectable landmarks. Thus, for *MA* methods the error of spatial resection reduces to errors in 2D coordinate estimation of known 3D coordinates projected onto the image plane [10]. In contrast, *MAL* methods have to deal with additional errors, like 1) outliers (e.g. non-static features), 2) 2D-2D correspondence errors from optical flow estimates and 3) 3D reconstruction errors stemming from inaccurate stereo vision, wrong disparities or scale estimations [14].

MAL methods usually require good illumination (enough brightness and contrast) of the environment, scenes rich in texture and a certain amount of feature overlap between frames.

To summarize, each method has its own advantages and problems. In terms of accuracy and computational complexity *MA* methods clearly outperform *MAL* methods. The

[1]All authors are within the Institute of Automatic Control and Mechatronics, TU Darmstadt, Germany.(racuna, zaijuan.li, vwillert)@rmr.tu-darmstadt.de

big advantage of *MAL* methods is that only features which are already present in the environment are needed for localization. Hence, it does not require the modification of the environment with artificial markers and/or a topological survey to define landmarks covering the whole navigation space of the sensor.

The main motivation of our work is to develop a real-time cooperative visual localization method that keeps the accuracy of marker-based pose estimation without having the need to modify the environment. For this purpose, we propose a cooperative visual odometry scheme based on mobile visual markers (*MOMA*).

Our work is an extension of the Cooperative Positioning System method (CPS) based on mobile landmarks developed by Kurazume et al. [1], but with a complete realization for the case when the landmarks are visual fiducial markers which can be detected with a monocular camera, e.g. Aruco markers [6]. This avoids the need of using expensive laser based sensors. Additionally, a study of the propagation of the error was performed based on the particulars of monocular camera fiducial marker detection and its pros and cons compared to other popular feature based *VO* and *V-SLAM* approaches.

Fiducial markers have been used for relative pose estimation and tracking in the robotic community for quite some time, e.g. as beacons for UAV autonomous landing [16] or as landmarks for the relative pose estimation of an UAV to a group of UGV's [4]. Common coordinate for multi-robot systems are also a topic of interest. Wildermuth et al. used a camera system mounted on top of a robot to calculate the relative position of each surrounding robot and their transformations in a common coordinate frame [17]. More recently, Dhiman et al. developed a system of mutual localization which uses reciprocal observation of fiducials for relative localization without egomotion estimates or mutually observable world landmarks [18]. To the best of our knowledge, the idea of cooperative visual odometry based on mobile visual markers has not been published.

The paper is structured as follows: In Sec. II, we introduce the basic principle of the *MOMA* odometry scheme including an analysis of possible error sources compared to other pose estimation systems. In Sec. III, we present different configurations of multi-robot-systems suitable to apply *MOMA* odometry. In Sec. IV a real robotic experiment is shown along with a comparison with state-of-the-art methods followed by an evaluation. We demonstrate that *MOMA* odometry is a reliable and accurate pose estimation method, especially when applied in multi-robot systems and summarize its pros and cons in Sec. V.

II. MOBILE MARKER BASED ODOMETRY

We define the concept of a Mobile Marker (*MOMA*) as a regular marker (fiducial or other kind of known feature) that has two possible configurable states at any given time: **Mobile**, if the marker is moving or permitted to move and **Static** otherwise. A *MOMA* can either be moved by some entity or by itself. We define the *observer* as the entity that performs the detection and pose estimation of the marker,
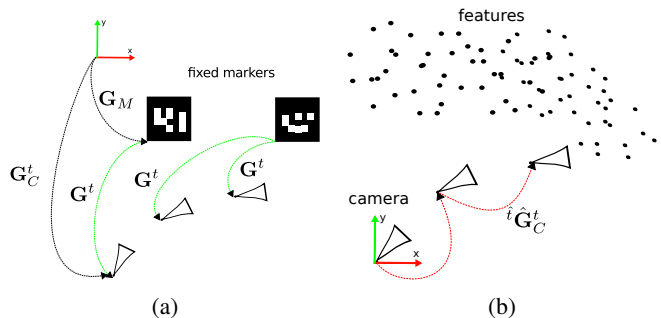

Fig. 1: Camera-based pose estimation methods. Marker-based (*MA*) pose estimation (a) uses known fixed markers with pose $\mathbf{G}_M$ to obtain the absolute pose of the camera $\mathbf{G}_C^t$ at each time instant t via the estimate $\mathbf{G}^t$ (in green). Visual odometry (b) detects fixed features along consecutive image frames in a markerless environment (*MAL-VO*) to estimate relative poses ($^t\hat{\mathbf{G}}_C^{\hat{t}}$) (in red) and infers the absolute pose $\mathbf{G}_C^t$ by concatenation.

in our case a camera. In order to do this pose estimation, the camera also needs to have one of these two states at a given time, **Mobile** or **Static** and also needs to use them in a certain way depending on the state of the marker.

The pose $\mathbf{G}$ in homogeneous representation[1] is given by the 3D translation vector $\mathbf{T} \in \mathbb{R}^3$ and the rotation matrix $\mathbf{R} \in \mathbb{R}^{3 \times 3}$.

a) Marker-based visual localization (MOMA): A marker is no more than a set of known features with known marker frame coordinates[2] $\mathbf{X}_M$. Visual marker based pose estimation uses known fixed markers *M* to obtain the absolute pose $\mathbf{G}_C^t$ of a camera *C* at some time *t* in world coordinates $\mathbf{X}_W$. We assume that the pose of the fixed marker in world coordinates $\mathbf{G}_M$ is known and also the structure of the marker is predefined and easy to detect. Once the marker is detected we can estimate the relative pose $\mathbf{G}^t$ of the marker in camera frame, and by extension the pose of the camera

$$\mathbf{G}_C^t = \mathbf{G}^t \mathbf{G}_M \tag{1}$$

in world coordinates using a PnP method. The error in global camera pose $\mathbf{G}_C^t$ will be only associated to the relative pose estimation between marker and camera $\mathbf{G}^t$. Hence, no drift will be accumulated as in dead reckoning approaches.

The reasons for the robustness and preciseness of a *MA*-based pose estimate is twofold. First, the 3D-2D correspondences $\{\mathbf{X}_M, \mathbf{x}^t\}$ can be extracted unambiguously using the knowledge about the configuration of the 3D points $\mathbf{X}_M$ on the marker [6]. Second, the coordinates $\mathbf{X}_M$ itself are known in advance from very precise measurements and do not have to be extracted online. Thus, the only source for errors is the extraction of the coordinates of the 2D projections $\mathbf{x}^t$ which depends on the resolution of the camera and the chosen method to get subpixel accuracy [7]. The relations for *MA*-based pose estimations are sketched in Fig. 1a.

[1] $\mathbf{G} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0_{1x3} & 1 \end{bmatrix}$

[2] All coordinates $\mathbf{X} = [X, Y, Z, 1]^T$ are assumed to be homogeneous coordinates, as long as not stated otherwise.

b) Markerless visual odometry (MAL-VO): Contrary to marker based pose estimation visual odometry is a dead reckoning (coupled navigation) approach given some initial known pose $\mathbf{G}_C^0$. To get the absolute position of the camera $\mathbf{G}_C^t$ the relative frame poses between time $\tilde{t} = t - 1$ and t, denoted $^{\tilde{t}}\mathbf{G}_C^t$, have to be estimated in order to get the absolute position via recursive accumulation:

$$\mathbf{G}_C^t = (^t\mathbf{G}_C^{\tilde{t}})\mathbf{G}_C^{\tilde{t}}. \tag{2}$$

The relative pose can also be extracted from the following 3D-3D correspondence

$$\mathbf{X}_C^{\tilde{t}} = (^{\tilde{t}}\mathbf{G}_C^t)\mathbf{X}_C^t. \tag{3}$$

Again including the collinearity equation, now the reprojection error between projected 3D coordinates $\mathbf{X}_C^t$ and 2D coordinates $\mathbf{x}^{\tilde{t}}$ can be formulated as follows:

$$\varepsilon_2^t = \|\mathbf{x}^{\tilde{t}} - \pi((^{\tilde{t}}\mathbf{G}_C^t)\mathbf{X}_C^t)\|_2. \tag{4}$$

Solving the least squares optimization

$$^{\tilde{t}}\hat{\mathbf{G}}_C^t = \text{argmin}_{\tilde{t}\mathbf{G}_C^t} \sum_{\mathbf{x}^{\tilde{t}},\mathbf{X}_C^t} \left(\varepsilon_2^t\right)^2, \tag{5}$$

leads to relative pose estimates $^{\tilde{t}}\hat{\mathbf{G}}_C^t$ (see also Fig. 1b). The 3D coordinates $\mathbf{X}_C^t$ of the features are not known and their estimation change over time. Thus, they have to be reconstructed as $\mathbf{X}_C^t = \lambda^t\mathbf{x}^t$, for example using a stereo vision system that extracts the depth λ^t of each 2D coordinate $\mathbf{x}^t$. Also a proper correspondence search to get the 2D-2D correspondences of $\{\mathbf{x}^{\tilde{t}}, \mathbf{x}^t\}$ coordinate pairs is needed for a proper reconstruction and a good optimization result from (5). Unfortunately, a correspondence search in a *MAL* environment is ambiguous and prone to errors because it is based on some optical flow algorithm [13]. Since this reconstruction is not error-free and accumulates along frames, the *MAL-VO* pose estimation is worse than *MA* pose estimation and prone to drift because of equation (2).

c) Mobile marker odometry (MOMA): In order to maintain the accuracy of fiducial marker pose estimation related to the camera $\mathbf{G}^t$ but using only one marker to cover the whole environment, the marker has to move. This means that the pose of the marker $\mathbf{G}_M^t$ may change at given time instances $t = \tau$ and the pose of the camera in world coordinates $\mathbf{G}_C^t$ is related to the marker pose via $\mathbf{G}^t$ as follows

$$\mathbf{G}_C^t = \mathbf{G}^t\mathbf{G}_M^{t=\tau}. \tag{6}$$

In order to get $\mathbf{G}_M^{t=\tau}$ at certain time instances τ, the pose change $^{\tau_2}\mathbf{G}_M^{\tau_1}$ of the marker between two specific consecutive time instances τ_1, τ_2 with $\tau_2 > \tau_1$ has to be estimated.

Once this pose change is known, the current pose of the marker $\mathbf{G}_M^{\tau_2}$ can be recursively calculated from the last marker pose in τ_1, which reads

$$\mathbf{G}_M^{\tau_2} = (^{\tau_2}\mathbf{G}_M^{\tau_1})\mathbf{G}_M^{\tau_1}. \tag{7}$$

Now we need to obtain this relative pose $^{\tau_2}\mathbf{G}_M^{\tau_1}$ by camera measurements. We start by fixing the camera into a static state with the following pose:

$$\mathbf{G}_C^{\tau_1} = \mathbf{G}^{\tau_1}\mathbf{G_M}^{\tau_1}. \tag{8}$$

For time interval $\tau_1 < t < \tau_2$ the marker is in the mobile state and it moves to a new fixed pose in τ_2 within the field of view (FOV) of the camera. Since the camera is static, the pose

$$\mathbf{G}_C^{\tau_2} = \mathbf{G}^{\tau_2}\mathbf{G_M}^{\tau_2} \tag{9}$$

is equal to $\mathbf{G}_C^{\tau_1}$. Hence, we can insert (8) into (9) and solve for the relative marker pose

$$^{\tau_2}\mathbf{G}_M^{\tau_1} = [\mathbf{G}^{\tau_2}]^{-1}\mathbf{G}^{\tau_1}. \tag{10}$$

The relative marker-camera poses $\mathbf{G}^{\tau_1}$ and $\mathbf{G}^{\tau_2}$ can be estimated and as long as the marker is static from time τ_2 on, the camera can acquire its pose as in the fixed marker case for all times $t > \tau_2$.

Although there is drift by the accumulation of the relative poses of the marker according to (7), as a matter of principle the accumulated error in (7) for mobile marker odometry is much lower than in (2) for visual odometry because no backprojection based on error-prone 3D reconstructions $\mathbf{X}_C^t$ has to be applied. Instead, only error-free marker coordinates $\mathbf{X}_M$ and very precise 3D-2D correspondences $\{\mathbf{X}_M, \mathbf{x}^t\}$ from a known fiducial marker that can be detected very robustly. Additionally, the error accumulation for *MOMA* odometry according to (7) only happens at discrete time instances $t = \tau_i$ which occur on a much lower frequency at certain waypoints rather than on the frame rate of the camera like in *MAL-VO*.

As a conclusion, the whole *MOMA* odometry is only based on applying the least squares optimization along a specific *caterpillar*-like (see also Sec. III) marker-camera motion pattern. The minimal motion pattern and concurrent optimizations is summarized in a plain vanilla pseudocode 1 for visual *MOMA* odometry.

Pseudocode 1 Basic algorithm for visual *MOMA* odometry

Initialize $\mathbf{G}_M^{\tau_1}$
while i: marker localization cycles **do**
  if $t = \tau_i$ **then**
    marker and camera static: Detect marker to get $\mathbf{G}^{\tau_i}$
  else if $\tau_i < t < \tau_{i+1}$ **then**
    marker mobile and camera static: Continuously detect
    marker to get $\mathbf{G}^t$ and (10), (7) to get $\mathbf{G}_M^t$
  else if $t = \tau_{i+1}$ **then**
    marker and camera static: Detect marker to get $\mathbf{G}^{\tau_{i+1}}$
    and (10), (7) to get $\mathbf{G}_M^{\tau_{i+1}}$
  else if $t > \tau_{i+1}$ **then**
    marker static and camera mobile: Detect marker to
    get $\mathbf{G}^t$ and (6) to get $\mathbf{G}_C^t$
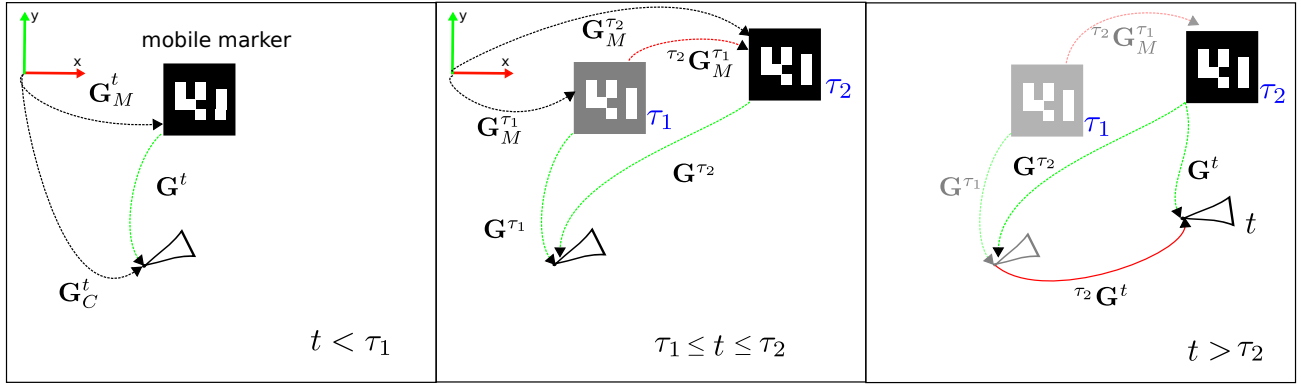 **end if**
end while

Fig. 2: The basic *MOMA* odometry cycle. At $t = 0$ the marker is static and the camera can obtain its initial pose $\mathbf{G}_C^0$ knowing the initial marker pose $\mathbf{G}_M^{\tau_1}$. In timesteps $0 \leq t < \tau_1$ the camera moves in relation to the static marker and estimates its pose $\mathbf{G}_C^t$ by estimating the relative pose $\mathbf{G}^t$ to the marker. During time $\tau_1 \leq t \leq \tau_2$ the camera is static and the marker starts to move to some new location in the FOV of the camera. Reaching time $t = \tau_2$ the marker stops moving and the marker pose change ${}^{\tau_2}\mathbf{G}_M^{\tau_1}$ can be estimated via $\mathbf{G}^{\tau_1}$ and $\mathbf{G}^{\tau_2}$. Finally, starting from $t > \tau_2$ the marker is static again and the camera moves using the marker pose $\mathbf{G}_M^{\tau_2}$ as a new reference to estimate its pose $\mathbf{G}_C^t$, closing the cycle.

The *advantages* of the visual *MOMA* odometry are: An **improved accuracy** with respect to other relative approaches like classical *MAL-VO*. **Less computation time**, because the detection and pose estimation of e.g. Aruco fiducial markers takes around 10ms [6] on a common 1 core PC, compared to realtime *VO* for common 1 core PCs e.g. 30-60 ms [19]. In its basic configuration only a **monocular camera** is needed. An important advantage is that it doesn't require features in the environment and no intervention of the environment is needed to setup the markers. Finally this method provides localization to the camera and the marker simultaneously even during movement. The *disadvantages* are an increased control and navigation complexity and the need of communication or coordination between the marker and the camera since the marker now has an associated state.

The motion patterns for *MOMA* odometry have the following movement restrictions:

1) The marker has to be static if the camera moves, and the camera has to be static as long as the marker moves. If more than one marker is used and one of the markers is static, then the camera is able to move all the time (which is not possible for *CPS* [1]).
2) The marker and the camera move in turns.
3) During the transitions, static to moving or vice versa, there must be a period of time *dt* where at least two devices are static (e.g. both camera and marker in a camera-marker configuration or two markers in a camera-multi-marker configuration).

A *MOMA* implies new considerations in the classical robotics action-perception cycle. The action-perception cycle is based on the premise of act then perceive or perceive and then act. Now, in the *MOMA* system we have what we call the perception-**interaction** cycle since the action of the marker affects the perception of the observer and in turn its action as well. The marker then can no longer be considered as a passive entity with no effect on the observer, a *MOMA* is able to provide information regarding its current state to
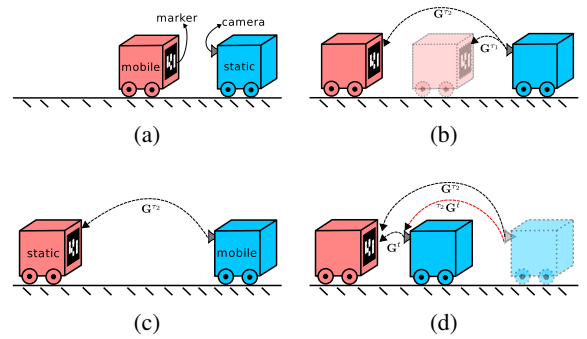

Fig. 3: Two-robot Caterpillar.

the observer, and the observer can also inform the *MOMA* which state is needed for the general behaviour of the system in a given situation.

III. POSSIBLE MOMA ROBOTIC ARCHITECTURES

In this section we will describe the possible robot configurations that we have considered based on monocular cameras and fiducial markers. In the experimental section the development and testing of a multi-robot system with one of these architectures will be shown.

A. Caterpillar-like Configurations

This is the most basic multi-robot configuration for the *MOMA* Odometry. It equals the structure we assumed in Sec.II to do the mathematical elaboration.

1) Two-robot Caterpillar: In this configuration one robot is the *MOMA* (the one with the marker) and the other one is the *observer* (the one with the camera), see Fig. 3. The *observer* follows the movement of the *MOMA* continuously thanks to the monocular camera. We named this particular kind of movement caterpillar-like motion, since each robot behaves like a segment of the body of a caterpillar.

The *MOMA* and the *observer* move in turns, following the rules explained in Sec. II. At the start, the *observer* is static
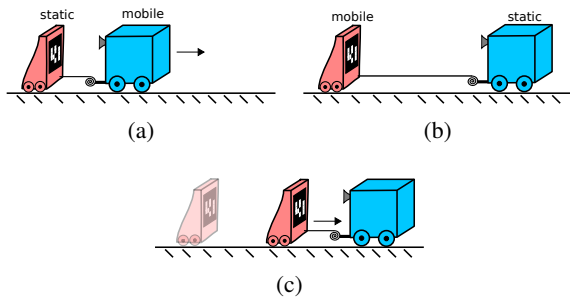
Fig. 4: Single-robot Caterpillar.

and the Moma is mobile and may move forward, Fig. 3b. Later on (Fig. 3c) the switching takes place, now *MOMA* is static and the *observer* is mobile. Finally, the *observer* moves as in Fig. 3d and the pose of the *observer* is obtained from marker detection closing the cycle.

The error will be accumulated only during the switching of the reference and is only dependent on the accuracy of the fiducial marker detection, which by using a good camera and proper calibration may be in the range of millimetres [20]. This system is also able to track the pose of the robots during the movement and not only in the transitions.

2) Single-robot Caterpillar: In this minimal configuration only one robot will be pulling a sled with a simple pulley mechanism, see Fig. 4. The robot can either actuate to pull the sled close to himself or let it drag behind. A monocular camera detects a fiducial marker in the front of the sled. The robot performs caterpillar-like motion leaving the sled behind as static reference when it has to move, then stops and pulls the sled performing the *MOMA* Odometry in the process.
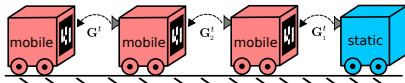

Fig. 5: Multi-robot Caterpillar.

3) Multi-robot Caterpillar: This is an extension of the basic caterpillar case for N robots, see Fig. 5. Each robot follows the one in front. In this configuration $N-1$ robots with cameras are needed for the relative transformations. If at least one member of the group is static, the rest may move.
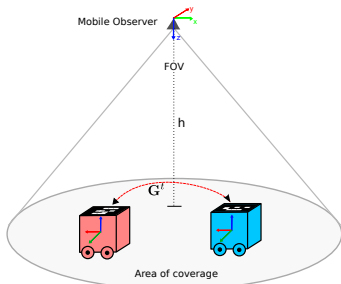
B. Top Mobile Observer

Fig. 6: Top Mobile Observer.

This configuration is based on two or more UGV's with fiducial markers on top and an external mobile *observer*

(UAV) which looks down to all the robots simultaneously using a monocular camera, see Fig 6. The UGV's move in turns as in *MOMA* Odometry but the *observer* is totally mobile.

The permitted action space on the ground $S = f(FOV, h)$ (area of coverage) for the movement of each robot will be a back projection of the field of view *FOV* of the camera on the ground plane dependent on the height from the camera to the ground h. Ideally, this coverage area will be centered in the middle of the UGV formation and the *observer* should adjust its pose in order to cover the major amount of the image with all the markers. If the robots are close together the h of the *observer* should decrease to improve the marker detection, and if they move further apart the *observer* has to move up in order to keep the markers inside the *FOV*.

The *observer* is a very general concept in this configuration, one logical choice is a quadcopter or any other type of UAV with a bottom camera. However, in our tests we also used a wireless camera in the hand of a person following the robots around the lab. An advantage of this configuration is that the *MOMA* Odometry system will also fully locate the *observer* and the *observer* is always allowed to be in continuous movement. A further advantage of measuring the relative pose between markers from a top observer is that the resolution of the camera is exploited equally for each of the marker-camera pose estimates, because of the same distance from camera to markers. This contributes to more precise estimates compared to the situation where the marker-camera poses are at different distances and the camera resolution can't be optimally exploited.

IV. EXPERIMENTS ON A MULTI-ROBOT SYSTEM

The Top Mobile Observer configuration (Fig 6) is more interesting because the area of coverage may be used as a local navigation space, with less robot movement restrictions than in the Caterpillar case. Hence, it was chosen to verify the accuracy of the Moma Odometry concept. This is also relevant in our group due to past research in the area of tracking and coverage using UAV's and UGV's [21].
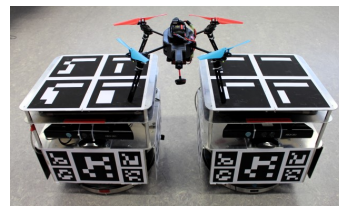
A. Hardware configuration

Fig. 7: Robots used in our experiments.

Our experimental setup consists of two omnidirectional robots (Robotino® from Festo Didactic Inc.). Each Robotino has an Aruco marker board on top, see Fig. 7. A wireless camera system was used for marker detection using a common configuration found in first person view racing drones. The video feed from the camera is transmitted to a ground station, digitized and processed by the PC (PAL format at

$25\,fps$). The UAV is an Ar.Drone 2.0 quadcopter with the wireless camera attached to the bottom and custom landing legs.
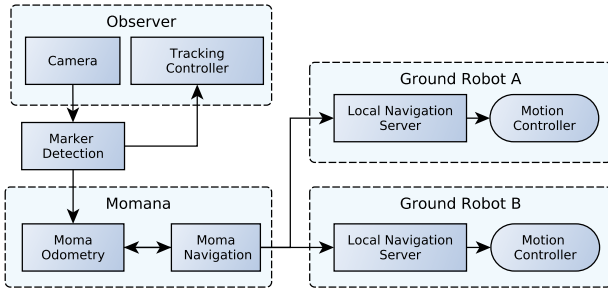
B. Software architecture

Fig. 8: Moma Odometry and multi-robot navigation system.

The modules which are part of our system are shown in Fig. 8. They comprise the tasks of marker detection, *MOMA* odometry estimation, global navigation planner (*MOMA* navigation), local navigation planner and motion control of the ground robots and the quadcopter. The code was implemented in the Robot Operating System (ROS) framework and is openly available at our research's group github account[3].

1) Marker Detection module: The flow of processing in the system starts at the *observer* level, where the images from the camera are captured and sent to the PC for marker detection. We use the ROS package *ar_sys*, which is a wrap of the Aruco detection library with ROS functionality for the detection. Additionally, we coded a pre-processing ROS package (*tud_img_prep*[4]) in charge of de-interlacing and adjusting the input image for optimal marker detection. The final outputs are the poses of all the markers detected in the image in camera coordinate frame.

2) Tracking Controller: The control of the quadcopter for proper tracking of the ground robots using the detected marker poses is implemented in this module. Note that only relative poses are needed for tracking, even though the *MOMA* Odometry system is able to provide it. The tracking control adjusts automatically the height, orientation and position of the quadcopter for optimal camera placement and marker detection.

3) Moma Odometry: Here, the algorithmic part of *MOMA* Odometry is implemented as explained in Section II. This module uses the detected markers in the camera frame to calculate the relative robot poses, it also tracks the current state of each *MOMA* (mobile or static) and calculates the pose of all the robots in the system, including the *observer* in the odometry coordinate frame.

There must be a time frame dt where both robots need to be static during the switching. Since the *observer* is mobile, the relative transform measurements will vary slightly with different *observer* positions (due to camera calibration errors

and image noise), so at any time the observer keeps an history of all the previous position estimates which then are used when both robots are static to calculate a better $\mathbf{G}^t$ transform during the switch, which serves as a good and simple strategy to minimize the accumulation of error.

4) Moma Navigation: The ROS navigation stack is used to calculate the navigation path from a current UGV pose to the next goal from a series of predefined waypoints. The calculated paths are then sent to ROS local move servers running on the laptops of each Robotino, which are in charge of performing the path-following. The ROS navigation stack currently is not properly adapted to multi-robot configurations, this means that the goals for the robots need to be configured manually by the user, taking in consideration the movement constrain of the *MOMA* Odometry scheme.

C. Experimentation and discussion

1) Waypoint navigation: A simple navigation task was defined for our robotic system as a set of goals that form a square shape (side=1m). Each goal is a position and orientation in the map coordinate frame $goal = (x, y, \theta)$. The navigation between the goals was performed using Moma Odometry and Moma Navigation.

In this experiment we wanted to compare the behaviour of our system to a VO approach in an environment that does not provide enough features for the VO. The square shaped navigation was performed in our laboratory, which has white walls, a radiator with a repetitive pattern and a floor without texture. This lack of features is usually a problem for VO systems. We added patterns rich in texture for the first half of the trajectory in the field of vision of the camera, while the second half was left without modification. As ground truth we used fixed ceiling HD cameras(*MA*) and we chose Viso2 [22] as the VO system. The final metric of comparison was defined as the final pose of the main robot after performing a loop measured by ceiling cameras. We calibrated the top *marker* coordinate frame and the *camera* of the robot offline using our marker-camera ROS calibration package [5] and we calibrated the intrinsic parameters of the cameras using the standard ROS Calibration package.

In Fig. 9, the result of one of the experiments is shown. For clarity, only the odometry information related to the main UGV is displayed. The blue solid line shows the odometry estimation using our proposed system (*MOMA*), the black dashed is the ground truth (*MA*) and the red solid line is the odometry estimation using visual odometry. The waypoints for the main robot are represented using yellow triangles.

Our odometry system follows the trajectory measured by the ground truth with great accuracy and is able to easily track the trajectory of the robots at all times even between transitions. The odometry estimation of VO is also accurate as long as there are enough features in the environment (first half of the trajectory) and the movement does not include pure rotations. When the main robot performs pure rotations at waypoint coordinates $(0,1), (1,1)$ and $(1,0)$, the error in

---

[3]http://github.com/tud-rmr/tud_momana
[4]http://github.com/tud-rmr/tud_img_prep

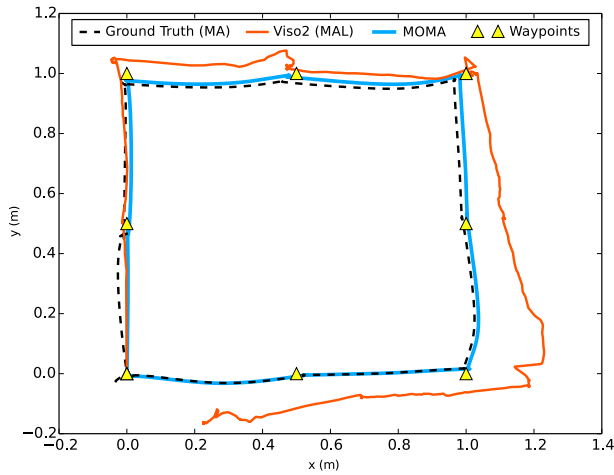[5]http://github.com/tud-rmr/tud_calibration

Fig. 9: Odometry results for the main robot after waypoint navigation. In red is shown the behaviour of Viso2, which how the errors increases during rotations and due to the lack of good features in and indoor environment. Moma based odometry (blue) follows the waypoints with low error.

the pose estimation for the VO case increases sharply. This is an expected behaviour for *MAL* based methods and confirms the advantages shown in Section II of the *MOMA* odometry system. In our tests we found an additional serious problem related to VO in cooperative robotic systems. The movement of other robots disturbs the measurements, e.g. if a robot moves too close to the camera it may occlude good static features.

The waypoint navigation task was executed 10 times in our robotic system with different configurations, using the UAV as *observer* and using a human with a hand-held camera as *observer*. The error of the estimation is defined as the euclidean distance between the position obtained by a given method and the position given by the ground truth (E), we then calculated the mean error for the trajectory ME. Our main metric of comparison was the error of the final position (E_f) after performing the navigation task and the mean of the E_f for all the tests was ME_f.

For our proposed method (*MOMA*) the mean error on the final position was $ME_f = 0.97cm$ ($std = 1.51$) which in percentage of the total trajectory (400cm) is 0.2425%, with a $ME = 1.97cm$ ($std = 0.69$). For Viso2 (*MAL*) we obtained a $ME_f = 33.08cm$ ($std = 16.42$), which in percentage of the total trajectory (400cm) is 8.27%, with a $ME = 16.29cm$ ($std = 6.98$), this only includes the cases were Viso2 didn't lose track, which happened in almost 40% of our tests. Our system's best case has 0.12% of error for the total distance of the navigation task (400cm) while Viso2's best case was 1.32%.

In Fig. 10 we perform a comparison of the behaviour of the error during the navigation task for two different cases: When the *observer* is the UAV and when it is a handheld camera. The error when using a handheld camera was less erratic than with the quadcopter. The error in both cases at the beginning was close to zero and at the end of the navigation it was less than 0.01m since the ground truth camera was
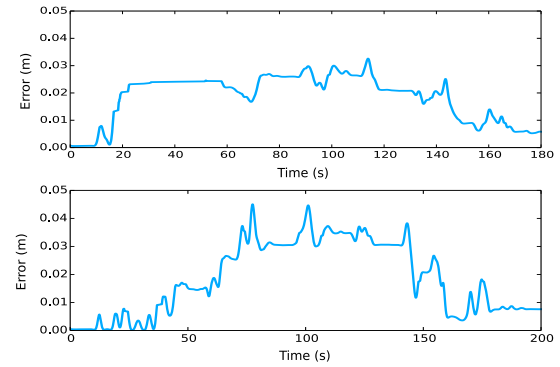

Fig. 10: Comparison of the *MOMA* odometry performance when using a handheld camera and a quadcopter.

calibrated for the starting position and the middle part of the trajectory is in the border of the ceiling camera *FOV*.

2) Line Following: This experiment was designed to show the behaviour of the error in Moma odometry. The main robot in a Top Observer configuration navigated a straight line of approximately 4.6 meters long three consecutive times (forward, backward and forward again) for an aproximate total distance of 13.785m. During the moments of the switching of reference (navigation keypoints) we measured with a laser the exact position of the robot in the X-axis. We used this as a ground truth to compare with the Moma estimation. During the course of a line segment navigation, 6 switching points (keypoints) were needed for a total of 18 for the whole trajectory. The results of the Moma estimation for one of the line segments and its corresponding keypoints are shown on Fig. 11. The pose error in X axis, in each keypoint for the whole trajectory is shown in Fig. 12. We also show the absolute value of the relative pose error in Fig. 13, that is the error in estimating the distance between keypoints. The final error after the 13.785m was 0.078m which correspond to 0.56% of the total path. It is possible to observe that even though some of the errors are high (10cm) they get cancelled between each other giving the final performance.
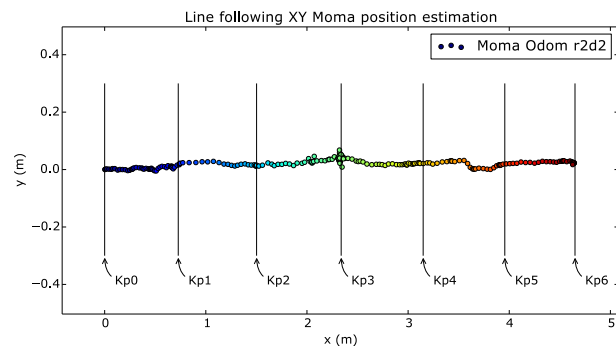

Fig. 11: Odometry results for the main robot after one third of the line following navigation. The first six keypoints (switching instants) are represented by vertical lines.

As a final evaluation of MOMA, we believe that the proposed method could be an interesting tool for existing multirobot systems, since it provides a convenient solution for cooperative robotics in featureless environments. How-
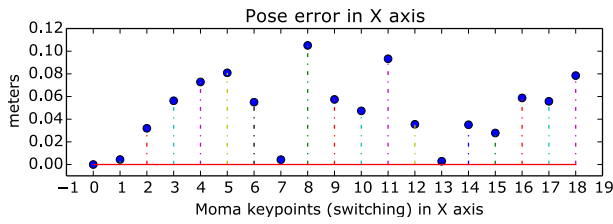
Fig. 12: Pose estimate error for the line following test, the error is bounded and no greater that 0.1 meters. These are absolute values, in practice some errors cancel each other.
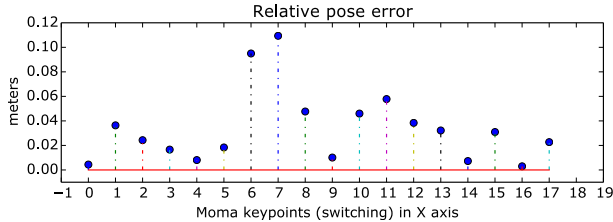

Fig. 13: The absolute value of the error in relative measurements for the line following test.

ever, there is still great room for improvement. According to our simulations, it is hard to obtain good relative measurements in the caterpillar-like motion (two UGVs), since the detection of Aruco markers does not provide good depth estimates (Z-axis of the camera). This may be solved by selecting other fiducial marker structures. The Top observer configuration is more precise since it is based on measurements on the XY plane of the camera, nonetheless, in order to give more freedom of movement for the UGVs, the UAV has to fly higher (decreasing marker detection accuracy) or the switching must happen when robots are in the border of the image (prone to distortion errors). Since the switching is the most critical part of the method (it is when the error accumulates), it is important to find new ways of improving the estimation accuracy by perhaps imposing additional constrains to the observer controller or by fusing the UAV's IMU measurements to counteract bad rotation estimates.

V. CONCLUSIONS AND FUTURE WORK

We demonstrated a *MOMA* Odometry system with greater accuracy than state-of-the-art MAL-based methods such as VO in featureless environments. Our proposed method is much easier to integrate into existing platforms since it only requires a cheap monocular camera and cheap fiducial markers, contrary to other methods. With our method no global positioning system like VICON is needed any more to conduct multi-robot navigation and control tasks. In future work we would like to improve measurement accuracy during transitions, e.g. by fusing the information from several robots observing each other and include the inertial sensors of the robots. Also, there is the necessity to implement a new layer (*MOMA* Navigation) on top of the ROS navigation stack, where the user can define a goal for the system, or for any individual robot and *MOMA* Navigation will calculate automatically the set of intermediate positions for each robot

and execute the path-planning and path-following with the *MOMA* constraint.

REFERENCES

[1] R. Kurazume, S. Nagata, and S. Hirose, "Cooperative positioning with multiple robots," in *IEEE Int. Conf. on Robotics and Automation*, 1994, pp. 1250–1257.

[2] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A probabilistic approach to collaborative multi-robot localization," *Autonomous robots*, vol. 8, no. 3, pp. 325–344, 2000.

[3] A. I. Mourikis and S. I. Roumeliotis, "Performance analysis of multirobot cooperative localization," *IEEE Trans. on Robotics*, vol. 22, no. 4, pp. 666–681, 2006.

[4] L. G. Clift and A. F. Clark, "Determining positions and distances using collaborative robots," in *Comput. Sci. Electron. Eng. Conf.*, 2015, pp. 189–194.

[5] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[6] S. Garrido-Jurado, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 4, no. 6, pp. 2280–2298, 2014.

[7] E. Marchand, H. Uchiyama, F. Spindler, E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality : a hands-on survey," *IEEE Trans on Visualization & Computer Graphics*, vol. 1, no. 1, 2016.

[8] "Visual odometry: Part II: Matching, robustness, optimization, and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 78–90, 2012.

[9] V. Willert, "Optical indoor positioning using a camera phone," in *Int. Conf. on Indoor Positioning and Indoor Navigation*, 2010.

[10] V. Händler and V. Willert, "Accuracy evaluation for automated optical indoor positioning using a camera phone," vol. 137, no. 2, pp. 114–122, 2012.

[11] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, "Vision-based slam: Stereo and monocular approaches," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–364, 2007.

[12] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1449–1456, 2013.

[13] V. Willert and J. Eggert, "A stochastic dynamical system for optical flow estimation," in *IEEE Int. Conf. on Computer Vision (ICCV Workshops)*, 2009, pp. 711–718.

[14] M. Buczko and V. Willert, "How to distinguish inliers from outliers in visual odometry for high-speed automotive applications," in *IEEE Intelligent Vehicles Symposium*, 2016, pp. 478–483.

[15] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visualinertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[16] W. Li, T. Zhang, and K. Kühnlenz, "A vision-guided autonomous quadrotor in an air-ground multi-robot system," in *IEEE Int. Conf. Robot. Autom.*, Shanghai, pp. 2980–2985.

[17] D. Wildermuth and F. E. Schneider, "Maintaining a common coordinate system for a group of robots based on vision," *Robotics and Autonomous Systems*, vol. 44, no. 3-4, pp. 209–217, 2003.

[18] V. Dhiman, J. Ryde, and J. J. Corso, "Mutual localization: Two camera relative 6-DOF pose estimation from reciprocal fiducial observation," *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 1347–1354, 2013.

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

[20] "Pi-Tag: A fast image-space marker design based on projective invariants," *Mach. Vis. Appl.*, vol. 24, no. 6, pp. 1295–1310, 2013.

[21] L. Klodt, S. Khodaverdian, and V. Willert, "Motion control for UAV-UGV cooperation with visibility constraint," in *IEEE Conf. on Control Applications*, pp. 1379–1385.

[22] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.