MOMA: Visual Mobile Marker Odometry

1st Raul Acuna
IAT
TU Darmstadt
Darmstadt, Germany
racuna@rmr.tu-darmstadt.de

2nd Zaijuan Li
IAT
TU Darmstadt
Darmstadt, Germany
zaijuan.li@rmr.tu-darmstadt.de

3rd Volker Willert
IAT
TU Darmstadt
Darmstadt, Germany
vwillert@rmr.tu-darmstadt.de

***Abstract*—In this paper, we present a cooperative visual odometry system based on the detection of mobile markers. To this end, we introduce a simple scheme that realizes visual mobile marker odometry via accurate fixed marker-based camera positioning and we discuss the characteristics of errors inherent to the method compared to classical fixed marker-based navigation and visual odometry. The proposed cooperative scheme has the advantage of not needing any feature or fiducial marker in the environment, which can be used for indoor and underwater applications where it can be harder to extract reliable features. We provide specific configurations of UAV and UGV robots including one that allows continuous movements of the UAV, and a minimal caterpillar-like configuration that works with one UGV alone. Finally, we present a real-world implementation and an evaluation of the proposed configurations.**

***Index Terms*—visual odometry, cooperative, robots, fiducial marker, pose estimation, UAV, UGV**

I. INTRODUCTION

Visual pose estimation and localization is a problem of interest in 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.

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 [1], [2]; The second category works markerless (*MAL*) without any 3D scene knowledge [2], [3].

MA methods estimate the relative camera 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 as well as on what kind of algorithm is used to realize spatial resection [4], [5].

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.

This work was sponsored by the German Academic Exchange Service (DAAD) and the Becas Chile doctoral scholarship.

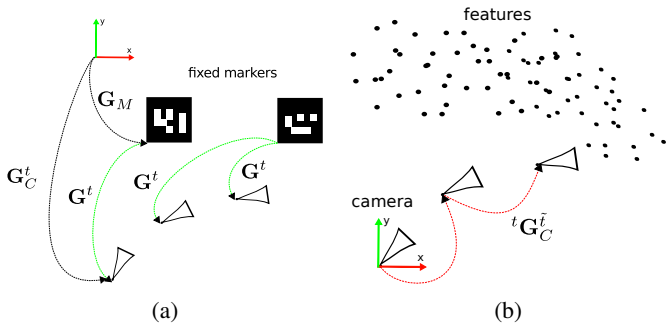

Fig. 1: Camera-based pose estimation methods. Marker-based (*MA*) pose estimation (a) uses known fixed markers with pose $\mathbf{G}_M$ to obtain the absolute pose of the camera $\mathbf{G}_C^t$ at each time instant t via the estimate $\mathbf{G}^t$ (in green). Visual odometry (b) detects fixed features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

These methods can be further divided into pure visual odometry (*VO*) [3] and more elaborate visual simultaneous localization and mapping (*V-SLAM*) approaches [6] including the new developments on Semi-Dense visual odometry [7]. Basic *VO* approaches estimate frame-to-frame pose changes of a camera based on some 2D feature coordinates, their optical flow estimates [8] and their 3D reconstruction using epipolar geometry in conjunction with an outlier rejection scheme to verify static features [9]. 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 [3].

V-SLAM approaches [6] 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 [10].

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 simple and unambiguously detectable landmarks. Thus, for *MA* methods, the error of spatial resection reduces to errors in the 2D coordinate estimation of known corresponding 3D coordinates projected onto the image plane [5]. 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 [9]. *MAL* methods usually require good illumination (enough brightness and contrast) of the environment, scenes rich in texture and a minimum amount of feature overlap between frames.

To summarize, in terms of accuracy and computational complexity, *MA* methods clearly outperform *MAL* methods. The biggest 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.

Fiducial markers have been used for relative pose estimation and tracking in the robotics community for quite some time, e.g. as beacons for UAV autonomous landing [11] or as landmarks for the relative pose estimation of an UAV to a group of UGV's [12]. Common coordinates systems for multi-robot configurations are also a topic of interest; Cooperative localization originally introduced by Kurazume et al. [13] introduces the use of some robots as moving landmarks and others to detect them using lasers. Fox et al. propose to use a sample-based version of Markov localization, which synchronizes each robot's belief to increase accuracy [14]. Wildermuth et al. uses 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 [15]. More recently, Dhiman et al. developed a system of mutual localization which uses reciprocal observation of fiducials for relative localization without ego-motion estimates or mutually observable world landmarks [16].

To the best of our knowledge, the idea of cooperative visual odometry based on mobile visual markers has not been published.

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 of environment features, or the modification of the environment to integrate fixed markers. The robots themselves will serve as the markers in the environment. This can be highly beneficial in cases where traditional odometry schemes fail due to environmental conditions, or in environments which are hard for traditional visual odometry algorithms, such as indoor spaces with flat textures, low light conditions, smoke or underwater environments.

For this purpose, we propose a cooperative visual odometry scheme based on mobile visual markers (*MOMA*). Our work is inspired by [13], 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 [1]. This avoids the need for using expensive laser-based sensors and opens the path to integrate this scheme into traditional visual odometry pipelines. 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.

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 another kind of known feature) that has one of 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, in our case a camera. The camera also needs to have one of these two states at a given time in order to perform pose estimation.

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 $\mathbf{R} \in \mathbb{R}^{3\times3}$.

a) Marker-based visual localization (MA): A marker is 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 the camera frame using a PnP method, and by extension the pose of the camera

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

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

[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.

The reasons for the robustness and preciseness of a *MA*-based pose estimate are 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 [1]. 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 [2]. The relations for *MA*-based pose estimations are sketched in Fig. 1a.

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 ${}^t\mathbf{G}_C^{\tilde{t}}$, 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

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

After including the collinearity equation, 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

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

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 changes 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 $\mathbf{x}^\mathbf{t}$. 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 algorithms [8]. 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 due to equation (2).

c) Mobile marker odometry (MOMA): In order to maintain the accuracy of fiducial marker pose estimation related to the camera $\mathbf{G}^t$ while 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 unambiguous and precise 3D-2D correspondences $\{\mathbf{X}_M, \mathbf{x}^t\}$ from a known fiducial marker that can be detected very robustly are used. 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 high frame rate of the camera like in *MAL-VO*. Finally, since the odometry scheme is only comprised of the camera and the mobile marker, no features in the environment are required.

As a conclusion, the whole *MOMA* odometry is only based on applying the least squares optimization along with a specific *caterpillar*-like (see also Sec. III) marker-camera motion pattern. The minimal motion pattern and concurrent optimizations are summarized in a plain vanilla pseudocode 1 and graphically in Figure 2.

The *advantages* of the visual *MOMA* odometry are: An **improved accuracy** with respect to other relative approaches like classical *MAL-VO*, only a **monocular camera** is needed

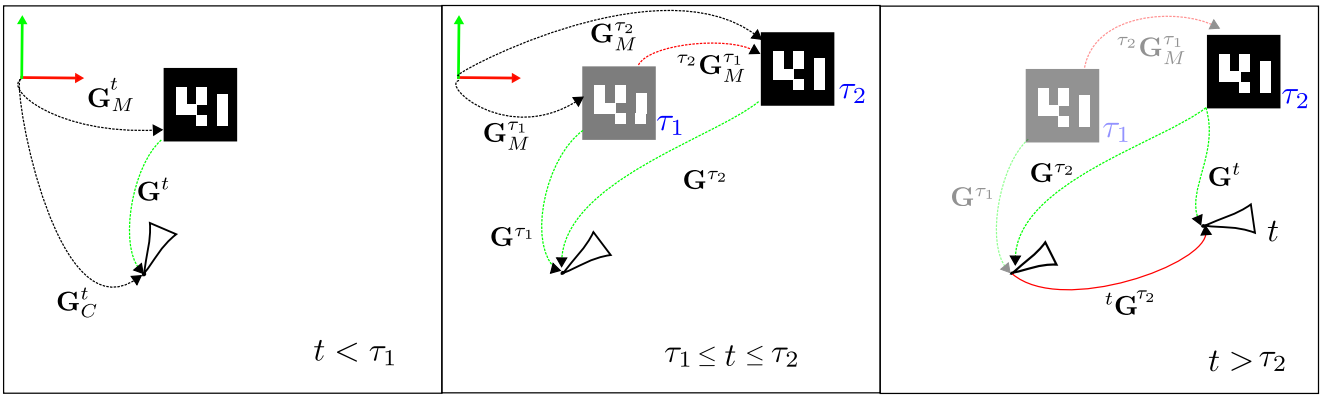

Fig. 2: The basic *MOMA* odometry cycle. At $t = 0$ the marker is static and the camera can obtain its initial pose $\mathbf{G}_C^0$ knowing the initial marker pose $\mathbf{G}_M^{\tau_1}$. In timesteps $0 \leq t < \tau_1$ the camera moves in relation to the static marker and estimates its pose $\mathbf{G}_C^t$ 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.

Pseudocode 1 Basic algorithm for visual *MOMA* odometry

```
Initialize G_M^{τ_1}
while i: marker localization cycles do
  if t = τ_i then
    static marker and static camera: Detect marker to get G^{τ_i}
  else if τ_i < t < τ_{i+1} then
    mobile marker and static camera: Continuously detect marker to get G^t and (10), (7) to get G_M^t
  else if t = τ_{i+1} then
    static marker and static camera: Detect marker to get G^{τ_{i+1}} and (10), (7) to get G_M^{τ_{i+1}}
  else if t > τ_{i+1} then
    static marker and mobile camera: Detect marker to get G^t and (6) to get G_C^t
  end if
end while
```

to localize several robots since the markers already provide the scale of the environment, and finally but more importantly, it **does not require features in the environment**. Additionally, this method provides localization to the camera and the marker simultaneously even during movement (both robots in the basic cooperative scheme with only one measurement device). The *disadvantages* are an increased control and navigation complexity, due to the restriction on the movement of the robots since the marker has to stay in the field of view of the camera.

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 at least one of the markers should remain static, then the camera and the rest of the markers are able to move (which is not possible for *CPS* [13]).
2) During the transitions, 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 action-perception cycle in robotics. 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. Thus, the marker 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 the observer, and the observer can also inform the *MOMA* which state is needed for the general behavior 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 traditional planar fiducial markers used in robotics. In the experimental section, the development and testing of a multi-robot system with two 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.

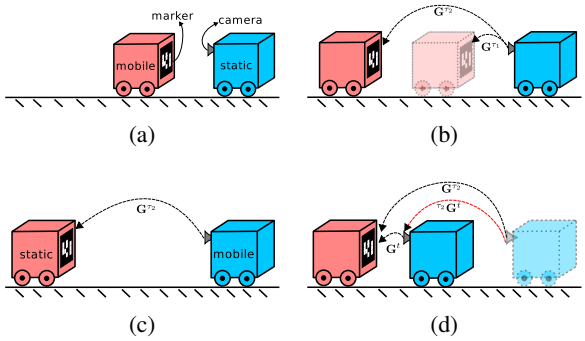

Fig. 3: Two-robot Caterpillar.

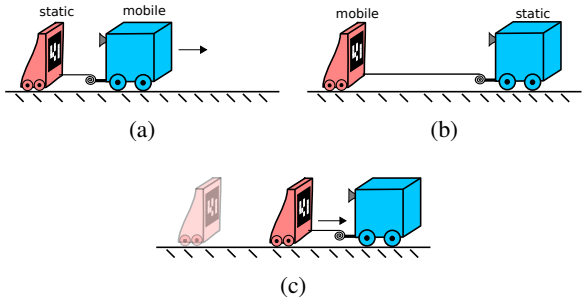

Fig. 4: Single-robot Caterpillar.

The *MOMA* and the *observer* move in turns, following the rules explained in Sec. II. The error accumulates 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 millimeters [17]. 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 lightweight and rigid sled with a simple pulley mechanism, see Figure 4. The robot can either actuate to pull the sled close to itself (while its wheels are locked to remain static) or let it drag behind. A monocular camera detects a fiducial marker in 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.

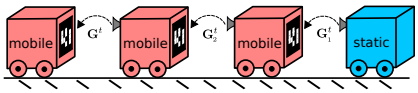

Fig. 5: Multi-robot Caterpillar.

3) Multi-robot Caterpillar: This is an extension of the basic caterpillar case for N robots, see Figure 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

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 Figure 6. The UGV's move in turns as in *MOMA* odometry but the *observer* is mobile.

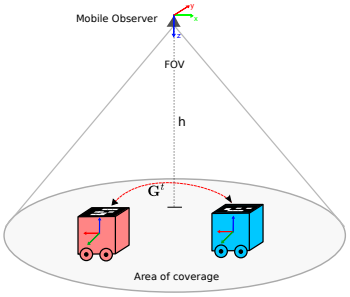

Fig. 6: Top Mobile Observer.

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. One advantage of this configuration is that the *MOMA* odometry system will also fully locate the *observer* and the *observer* is allowed to be in continuous movement.

C. Summary

The *MOMA* configurations may appear at first as a big restriction for the movement of a multi-robot system, but in practice, only one of the robots has to remain static to keep the *MOMA* odometry working, and this robot may be designed as a simple robot, since it only needs to move, not sense. The multi-robot system can use *MOMA* odometry to get to a particular working space, then the simple robot remains static meanwhile the complex ones move freely using some other odometry systems to perform their tasks, with the advantage that they can return to the static robot to correct drift as necessary. After finishing their tasks, they can move again as a group using *MOMA* odometry. This allows the robot team to maintain a drift-less odometry estimation meanwhile being able to perform more complex tasks in the environment.

IV. Experiments on a Multi-Robot System

A. Hardware configuration

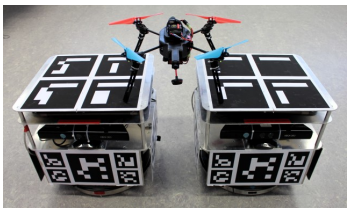

Fig. 7: Robots used in our experiments.

For a caterpillar-like configuration our experimental setup consists of two omnidirectional robots (Robotino® from Festo Didactic Inc.). Each Robotino has Aruco markers on the sides and top (Figure 7). One of the robots was defined as the observer (with a FLIR Blackfly monocular camera) and the

other as the mobile marker. For the top observer configuration, we use a quadcopter as an addition.

We calibrated the coordinate frames of the markers and the camera of the robots using the *easy-hand-eye* ROS calibration package and the intrinsic parameters of the cameras using the standard ROS camera calibration package.

B. Software architecture

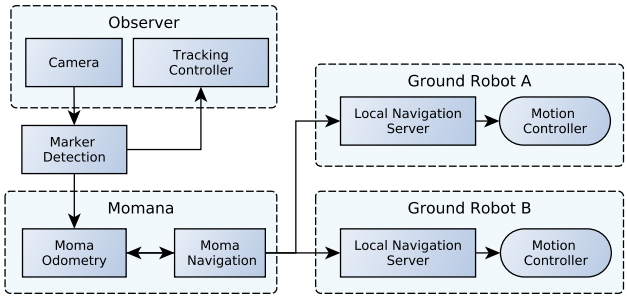

Fig. 8: *MOMA* odometry and multi-robot navigation system.

The modules which are part of our system are shown in Figure 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], the active mobile marker may be manually controlled by a human operator using a joystick overriding the automatic navigation.

C. Experimentation and discussion

1) Two-robot Caterpillar: This test was performed inside a room with a high precision OptiTrack localization system as ground truth. The two ground robots navigated 3 loops inside the room using the *MOMA* odometry scheme in caterpillar-like motion. The odometry of the robot's monocular camera was recorded both using *MOMA* odometry and the ground truth from the OptiTrack system. The measured trajectories for the final loop are displayed in Figure 9. The final error (Euclidean distance) was **0.045 meters or 0.138%** of the total trajectory. The behavior of the error during the whole navigation is shown in Figure 10. The error presents peaks followed by stabilizations periods and the peaks happen during the movement of the robots since the markers may appear blurred and in non optimal configurations for the PnP pose estimation algorithm, but as soon as the robots are static for doing the transition the error decreases. Additionally, it was observed that in closed-loop motions like this the errors are canceled due to symmetric motions. In order to properly study the performance of the *MOMA* odometry, a longer test without repetitive motions was performed.

In Figure 11 the results of a long trajectory are shown. The robots moved from room A, the one with the Optitrack

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

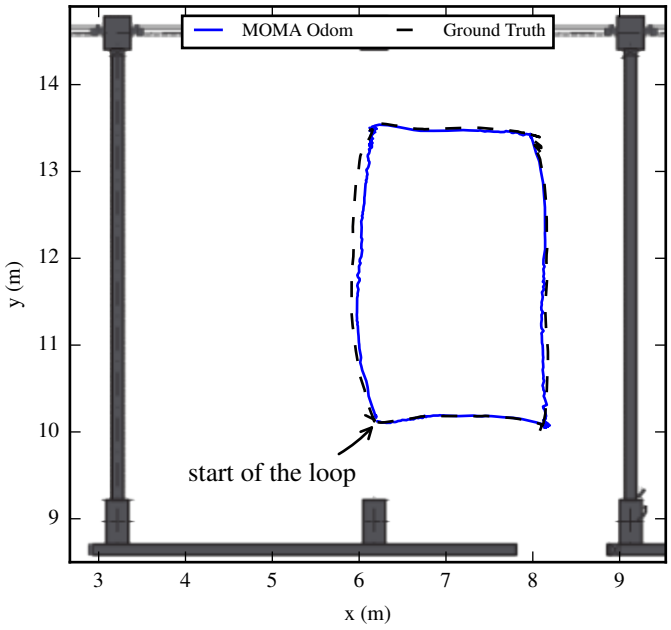

Fig. 9: Final loop of the navigation for the two-robot caterpillar configuration. The blue continuous line is the *MOMA* odometry and the black dashed line the ground truth from the OptiTrack system.

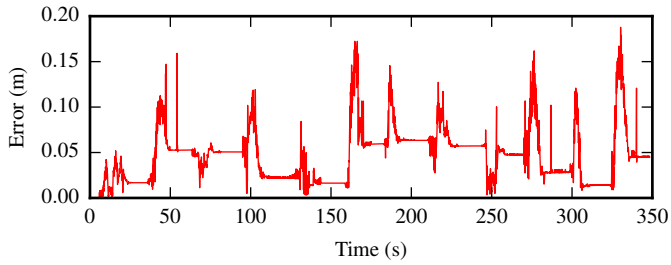

Fig. 10: Euclidean error of MOMA odometry during the trajectory of the final loop in the two-robot caterpillar test.

ground truth system, passed through a low illuminated hallway into room B and finally they returned again to Room A. With this test it was possible to measure with the Optitrack system the start and final position of the robots and compare against *MOMA* odometry. In Figure 12 a magnification of the starting and final trajectory of the robot and the ground truth are shown, the error at the start and end of the trajectory is shown in Figure 13. The final error was **0.38 meters or 0.68% of the total trajectory**.

2) Top Observer Configuration: Finally, the configuration presented in Figure 6 was implemented and tested. The UAV is an Ar.Drone 2.0 quadcopter with a camera attached to the bottom. A simple navigation task similar to the caterpillar case 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 main goal of this experiment was to compare *MOMA* odometry to a regular VO approach in an environment that does not provide enough features. The test was performed in a room in our laboratory which has white walls and a floor with a repetitive texture. This set-up is usually a problem for

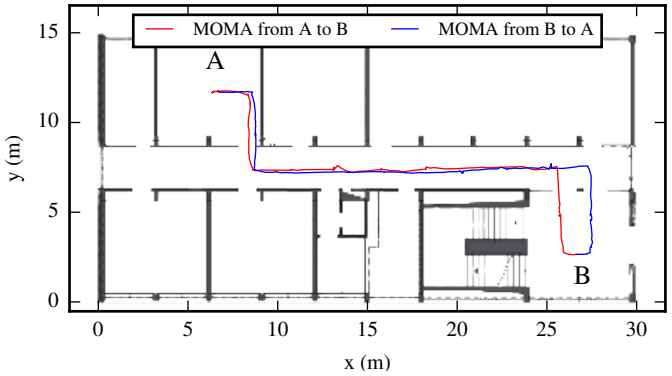

Fig. 11: Results of the *MOMA* odometry for a long trajectory from Room A to Room B (red line) and the return trajectory (blue line) over a map of our institute.

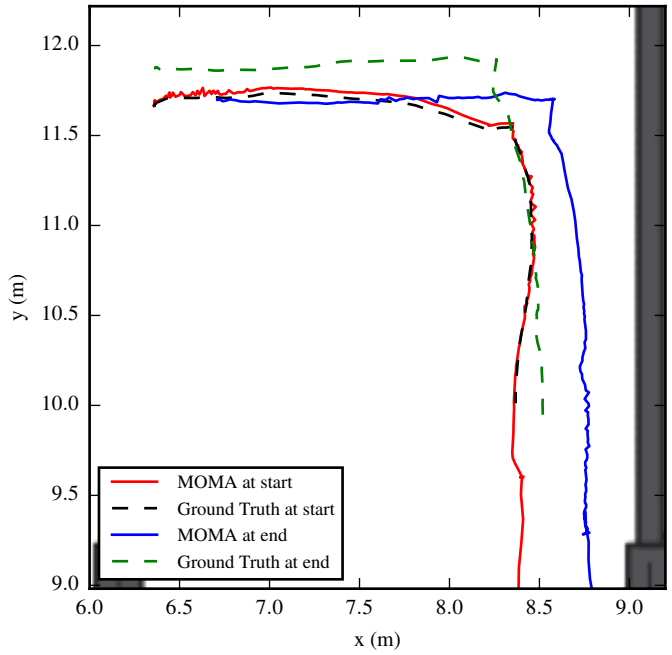

Fig. 12: Detail of the trajectories shown in Figure 11 for Room A with the ground truth obtained from OptiTrack.

VO systems.

For comparison, a frontal placed stereo camera was integrated into one of the UGVs and this video feed was used to perform visual odometry using the VISO2 algorithm [18] during the navigation task. The performance of both *MOMA* odometry (using the monocular camera on the UAV) and VISO2 for this robot were compared against the ground truth provided by fixed ceiling cameras. The final metric of comparison was defined as the final pose of the main robot after performing a loop measured by the ground truth system.

This test was executed 10 times. In Figure 14, the result of one of the experiments is shown. This test corresponds to the best VISO2 case for the navigation task, and VISO2 lost reference completely in 4 of the 10 cases. For clarity, only the odometry information related to the main UGV is displayed. VISO2 is only accurate as long as there are enough

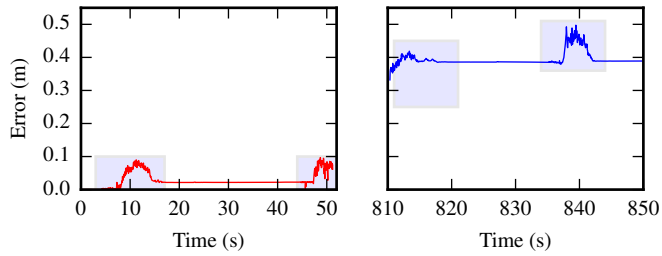

Fig. 13: Detail of the error for the long trajectory two-robot caterpillar test. Left side: error detected in Room A at the start of the trajectory. Right side: error detected in Room A at the end of the trajectory. The highlighted sections correspond to the movement of the robot followed by a static phase, during movement the accuracy decreases but improve when static.

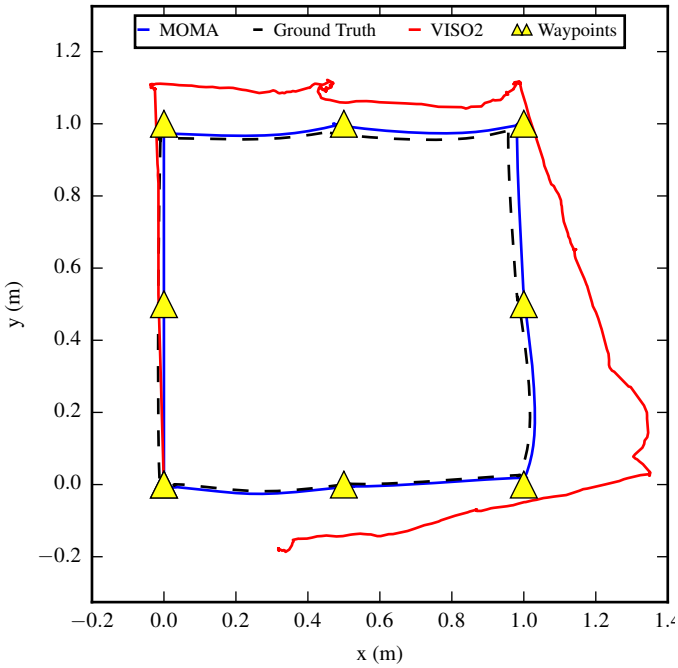

Fig. 14: Odometry results for the main robot after waypoint navigation. In red is shown the behaviour of Viso2, which how the errors increases during rotations and due to the lack of good features in and indoor environment. *MOMA* based odometry (blue) follows the waypoints with low error.

features in the environment (first quarter of the trajectory) and when 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 the pose estimation for the VO case increases significantly. A video of this test is also available[4].

For *MOMA* the error on the final position was 0.51cm or a 0.123% of the total trajectory. For VISO2 we obtained an error of 33.81cm or 7.99%, this was the best case for VISO2. It has to be pointed out that VISO2 and other VO algorithms may achieve close to 1% accuracy in environments with good features. However, this test was designed to be particularly challenging, highlighting the advantages of *MOMA* odometry in low-feature (or featureless) environments.

[4]https://youtu.be/0xASGfH8cDM

D. Summary

As a final evaluation of *MOMA*, we believe that the proposed method could be an interesting tool for existing multi-robot systems in featureless environments. Possible practical applications include: 1) Robot localization in indoor environments low in textures such as empty hangars or buildings, 2) Completely dark environments (the markers may be implemented as LEDs), this saves energy versus having to illuminate the whole environment, 3) Underwater environments, e.g. pool cleaning robots or seabed exploration, 4) Finally, together with any other odometry system to increase the localization accuracy.

We consider as well that *MOMA* odometry is a tool that can be used in conjunction with other odometry estimation methods and it can be integrated into other VO approaches to increase accuracy and robustness by using the information from environment features and *MOMA* markers into one optimization. We plan to continue the research on this direction.

Some remarks about the planar fiducial marker detection are relevant. In a caterpillar-like configuration the estimation of the pose for planar fiducial 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 to the UGVs, the UAV has to fly higher (which may decrease marker detection accuracy). 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, such as imposing additional constraints on the observer controller or by fusing the UAV's IMU measurements to counteract bad rotation estimates.

V. Conclusions and future work

We demonstrated a high accuracy cooperative visual odometry system without the need of environment features and with better accuracy than state-of-the-art MAL-based methods such as *VO* in featureless environments and comparable accuracy to *VO* methods in feature-rich environments. Our proposed method is easy to integrate into existing multi-robot systems since it only requires a cheap monocular camera and cheap fiducial markers. We believe that *MOMA* is particularly interesting for challenging environments (e.g. underwater environments). In the future, we would like to integrate *MOMA* in conjunction with other VO schemes for more motion flexibility and work towards improving the measurement accuracy during transitions, by fusing the information from several robots observing each other and include the inertial sensors of the robots. Additionally, we would like 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* movement constraints and select relative poses which maximize the PnP detection accuracy.

References

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

[2] 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, 2016.

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

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

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

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

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

[14] 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.

[15] 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.

[16] 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.

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

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